

René Noack

**Content- und benutzungsgesteuertes
generisches Layout
Screenography**



Cuvillier Verlag Göttingen
Internationaler wissenschaftlicher Fachverlag

Content- und benutzungsgesteuertes generisches Layout

Content- und benutzungsgesteuertes generisches Layout

Screenography

René Noack

Dissertation
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
(Dr.-Ing.)

der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel
eingereicht im Jahr 2016

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

1. Aufl. - Göttingen: Cuvillier, 2018
Zugl.: Kiel, Univ., Diss., 2016

© CUVILLIER VERLAG, Göttingen 2018
Nonnenstieg 8, 37075 Göttingen
Telefon: 0551-54724-0
Telefax: 0551-54724-21
www.cuvillier.de

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen.

1. Auflage, 2018

Gedruckt auf umweltfreundlichem, säurefreiem Papier aus nachhaltiger Forstwirtschaft

ISBN 978-3-7369-9639-7
eISBN 978-3-7369-8639-8

Abstract

Information-intensive websites are constantly changing. It concerns the content as well as the layout. Therefore an adequate design process is highly relevant for such systems. Current approaches are mainly focused on content aspects. Only little attention has been paid to the layout development even though an inappropriate presentation is a main reason for acceptance and usability troubles.

The design of current web information systems considers presentation aspects very early or very late but not accompanying. The lack of integration into the development process causes flexibility and adaptivity restrictions. Moreover it limits the possible evolution of the system.

Currently, the presentation is usually developed at the implementation level. Adaptation as well as personalisation can only be realised on a small scale or with great effort at this low level of abstraction. On changing presentation requirements as a result of structural content changes can rarely be responded adequately. In particular, slow evolutionary changes can't be handled appropriately. A main reason is the strong connection that still exists between the content and the specification of its presentation. Furthermore, there are no options to check the presentation adequacy.

Screenography aims at a systematic development and presentation of websites. It combines web-engineering with current HCI approaches and allows an adequate presentation of content and functionality in consideration of presentation options and presentation preferences. For this purpose, it picks up aspects of scenography and dramaturgy, adopting and generalising it to the demands and issues of screen applications.

Existing restrictions of current approaches are removable by specifying the layout at a higher level of abstraction. It eases the reuse of decisions concerning the presentation and allows adaptive, content- and usage-driven generic layout.

It is possible to completely detach the layout development from implementation issues by separating presentation demands of the application and the content from presentation options of layout components. On the other hand, layout components can specify their presentation abilities to be able to derive presentable content types. Thus, layout becomes a collection of suitable presentation containers, whose use and interplay is determinable by scenography and dramaturgy rules. This late decision of platform-specific presentation issues helps to avoid advancement restrictions and benefits platform as well as technology independence.

Zusammenfassung

Webinformationssysteme (WIS) großen Umfangs erfahren fortwährend Änderungen. Sowohl die Inhalte als auch die Darstellung sind von dieser Dynamik betroffen. Dem Entwurf solcher Systeme kann daher eine enorme Bedeutung beigemessen werden. Der Fokus aktueller Ansätze liegt vor allem auf den Anforderungen, die den Content betreffen. Die Entwicklung des Layout erfährt hingegen nur wenig Beachtung, wenngleich eine ungeeignete Darstellung eine der Hauptursachen späterer Akzeptanz- und Nutzungsprobleme ist.

Beim Entwurf derzeitiger WIS wird die Darstellung entweder sehr früh oder aber sehr spät entschieden, nicht aber begleitend. Diese fehlende Integration in den Entwicklungsprozess führt nicht nur zu Beschränkungen der Flexibilität und Adaptivität, sondern begrenzt die mögliche Evolution des Systems.

Die Entwicklung der Darstellung erfolgt derzeit vor allem auf dem Implementationslevel. Adaption als auch Personalisierung sind auf diesem niedrigen Level der Abstraktion nur mit großem Aufwand oder in kleinem Umfang realisierbar. Auf veränderte Darstellungsanforderungen infolge struktureller Änderungen von Inhalten kann daher nur selten angemessen reagiert werden, insbesondere dann nicht, wenn sie im Zuge der Evolution des Systems entstehen. Hauptursache ist die starre Bindung zwischen den Inhalten und der Spezifikation ihrer Darstellung. Darüber hinaus fehlen aber auch Möglichkeiten, um die Eignung der Darstellung prüfen zu können.

Die Screenography hat die systematische Entwicklung und Gestaltung von Websites zum Ziel. Sie hat die Aufgabe das Web-Engineering mit bestehenden HCI-Ansätzen zu verbinden und ist bestrebt, Inhalte und Funktionalität unter Beachtung bestehender Präsentationsmöglichkeiten und -präferenzen geeignet darzustellen. Dazu greift sie Aspekte der Szenographie und Dramaturgie auf und überträgt diese auf die speziellen Erfordernisse und Problemstellungen von Bildschirmanwendungen.

Die bestehenden Beschränkungen der bisherigen Ansätze lassen sich auflösen, indem die Spezifikation der Darstellung auf einem höheren Abstraktionsniveau erfolgt. Dies erleichtert nicht nur die Wiederverwendbarkeit von Gestaltungsentscheidungen, sondern macht adaptives, content- und benutzungsgesteuertes generisches Layout möglich.

Das vollständige Loslösen der Entwicklung der Gestaltung von der konkreten Implementierung lässt sich erreichen, indem die Anforderungen der Anwendung und der Inhalte an die Darstellung unabhängig von den Möglichkeiten der Darstellungskomponenten festgelegt werden. Demgegenüber legen Darstellungskomponenten fest, welche Darstellungsmöglichkeiten sie besitzen und machen damit entscheidbar, welche Inhaltstypen präsentierbar sind. Das Layout wird dadurch zu einer Sammlung geeigneter Präsentationscontainer, deren Verwendung und Zusammenspiel mittels Szenographie und Dramaturgie bestimmt werden kann. Durch das späte Entscheiden der plattform-spezifischen Darstellung wird die Entwicklung als auch die Weiterentwicklung nicht beschränkt und begünstigt zudem Plattform- sowie Technologieunabhängigkeit.

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einführung | 1 |
| 1.1 | Derzeitige Probleme | 1 |
| 1.2 | Ziele | 5 |
| 2 | Der Entwicklungsprozess von WIS | 8 |
| 2.1 | 6 Dimensionen der Entwicklung von WIS | 8 |
| 2.1.1 | Intention | 9 |
| 2.1.2 | Story | 9 |
| 2.1.3 | Context | 14 |
| 2.1.4 | Content | 17 |
| 2.1.5 | Functionality | 17 |
| 2.1.6 | Presentation | 18 |
| 2.2 | Zusammenwirken der Dimensionen | 19 |
| 2.2.1 | Website Entwicklung - state of the art | 20 |
| 2.2.2 | Website Entwicklung - Architektur | 21 |
| 2.2.3 | Website Entwicklung - Kunst | 22 |
| 2.2.4 | Website Entwicklung - Zukunft (state of science) | 23 |
| 2.3 | Flexibilisierung des Entwicklungsprozesses | 24 |
| 2.4 | Website-Typen | 26 |
| 2.4.1 | Infotainment sites | 28 |
| 2.4.2 | eBusiness sites | 31 |
| 2.4.3 | Identity sites | 32 |
| 2.4.4 | Election sites (hybrid) | 34 |
| 3 | Screenography | 36 |
| 3.1 | Ursprung und Ziele | 37 |
| 3.2 | Gestaltungsgrundlagen | 39 |
| 3.3 | Lessons Learned | 41 |
| 3.4 | Raum und Handlung | 43 |
| 3.4.1 | Beziehungen von Gestaltungsräumen | 43 |
| 3.4.2 | Erschließung des Gestaltungsraums | 44 |
| 3.4.3 | Raumdefinition | 47 |
| 3.5 | Szenographie des Gestaltungsraums | 50 |
| 3.5.1 | Entwicklungsdimensionen der Szenographie | 51 |
| 3.6 | Dramaturgie des Gestaltungsraums | 52 |
| 3.6.1 | Unterschiede bei der Dramaturgie von WIS | 55 |

| | | |
|----------|--|------------|
| 3.6.2 | Story-Typen | 59 |
| 3.6.3 | Szenen-Eigenschaften | 66 |
| 3.7 | Screenography im Detail | 68 |
| 3.7.1 | Screenography am Beispiel einer Museumsanwendung | 69 |
| 4 | Patterns und Grids | 75 |
| 4.1 | Grids | 75 |
| 4.1.1 | Grid Geometrie | 76 |
| 4.1.2 | Grid Typen | 78 |
| 4.1.3 | Grid Spezifikation | 79 |
| 4.1.4 | Grid Erweiterung - (grid cluster) | 88 |
| 4.1.5 | Ausblick | 89 |
| 4.2 | Templates | 89 |
| 4.3 | Patterns | 91 |
| 4.3.1 | Verwandte Arbeiten | 92 |
| 4.3.2 | Pattern Klassifikation | 93 |
| 4.3.3 | Composition patterns | 94 |
| 4.3.4 | Progress patterns | 100 |
| 4.3.5 | Evolution pattern | 106 |
| 5 | Adaption | 109 |
| 5.1 | Adaptionsfähigkeiten aktueller CMS | 110 |
| 5.2 | Adaption der Gestaltung | 111 |
| 6 | Resümee | 116 |
| | Glossar | 117 |

Abbildungsverzeichnis

| | | |
|------|---|----|
| 2.1 | 6 Dimensionen der WIS Entwicklung | 9 |
| 2.2 | Szenenübergang | 13 |
| 2.3 | Dimensionen der Website Entwicklung - state of the art | 20 |
| 2.4 | Dimensionen der Website Entwicklung - Architektur | 21 |
| 2.5 | Dimensionen der Website Entwicklung - Kunst | 22 |
| 2.6 | Dimensionen Hierarchie | 23 |
| 2.7 | Dimensionen der Website Entwicklung - Zukunftsszenario | 24 |
| 2.8 | Grundgerüst der Entwicklung | 25 |
| 2.9 | Perspektiven der Entwicklung - User (links), System (rechts) | 26 |
| 2.10 | Entwicklungsdimensionen - Notation | 28 |
| 2.11 | Dimensionen bei Infotainment sites | 30 |
| 2.12 | Dimensionen bei eBusiness sites | 32 |
| 2.13 | Dimensionen bei Identity sites | 32 |
| 2.14 | Schema Entwicklung bei Identity sites | 34 |
| 2.15 | Dimensionen bei Election sites | 35 |
| | | |
| 3.1 | Gestaltungsraum Einflüsse | 43 |
| 3.2 | Räume in der Architektur | 44 |
| 3.3 | Szenographie Entwicklung | 52 |
| 3.4 | Szenographie Entwicklung - Einflüsse | 52 |
| 3.5 | Dramaturgie Entwicklung | 53 |
| 3.6 | Unterbrechbarkeit von Handlungen | 55 |
| 3.7 | Story-Elemente | 60 |
| 3.8 | Vollständige Story | 61 |
| 3.9 | Unvollständige Story | 62 |
| 3.10 | Echte Story | 63 |
| 3.11 | Personalisierbare Story | 65 |
| 3.12 | Nicht und partiell terminierende Story | 65 |
| 3.13 | Vollständig terminierende Story | 66 |
| 3.14 | Unerreichbare Szene / unerreichbares Story-Ende | 67 |
| 3.15 | Partiell erreichbare Szene / partiell erreichbares Story-Ende | 67 |
| 3.16 | Szenenbehandlung bei mehreren Start- und Endszenen | 68 |
| 3.17 | Museums Story - Proaktive Suche | 71 |
| 3.18 | Museums Story - Reaktive Suche | 72 |
| 3.19 | Museums Story - Einfacher Rundgang | 73 |
| 3.20 | Museums Story - Individueller Rundgang | 74 |

| | | |
|------|--|-----|
| 4.1 | 3-Spalten Grid: Aufbau | 76 |
| 4.2 | Rhythmische Struktur von Gridelementen | 77 |
| 4.3 | Fibonacci Grid Modell | 78 |
| 4.4 | Evolutions Grids | 88 |
| 4.5 | Pattern Klassifikation | 95 |
| 4.6 | 3-Spalten Grid - geeignetes Zuordnungsmuster | 97 |
| 4.7 | 3-Spalten Grid - ungeeignetes Zuordnungsmuster | 98 |
| 4.8 | Visuelle Abhängigkeiten - Grid Größen | 100 |
| 4.9 | Visuelle Abhängigkeiten - Grid Stapel | 100 |
| 4.10 | Visuelle Abhängigkeiten - Grid Größen & Stapel | 101 |
| 4.11 | Dimensionen der Evolution | 101 |
| 4.12 | Typen des Handlungsverlaufs | 105 |
| 4.13 | Handlungstypen: Beziehungen | 107 |

1 Einführung

1.1 Derzeitige Probleme

Ein Webinformationssystem (WIS) ist ein Informationssystem, auf welches über das World-Wide-Web zugegriffen werden kann. Ein solches System kann allgemein von beliebiger Größe sein und von einer unbeschränkten Anzahl von Nutzern benutzt werden. Gerade umfangreiche WIS unterliegen permanenten Änderungen, die das System adäquat verarbeiten muss. Dem Entwurf dieser Systeme kann daher eine enorme Bedeutung beigemessen werden. Dies führte zu einer Reihe von Entwicklungsmethoden wie OOHDM [59], WebML [7], WIS co-design [51, 52], HERA [20], WSDM [13] und sogar UML wurde angepasst, um die Entwicklung von WIS zu unterstützen [11, 29]. All diese Methoden sind Content orientiert und generieren letztlich Webseiten, wobei das Layout bisher nur wenig Beachtung erfährt, wenngleich ein ungeeignetes Layout eine der Hauptursachen von Akzeptanz- und Nutzungsproblemen ist.

Die Entwicklung des Layout ist neben der Interaktion Gegenstand der HCI Techniken [4, 6, 23, 71], die jedoch keine Bindung zu den WIS Techniken aufweisen, obwohl technologisch keine fundamentalen Probleme zur Verbindung dieser Ansätze auf dem Implementationslevel bestehen. Trotz der fehlenden Bindung zu WIS Techniken bieten die entwickelten Präsentations- und Interaktionskonzepte der HCI Community [3, 30, 62, 48] Hilfestellungen, um geeignete User-Interfaces auch im Falle großer Informationssysteme zu realisieren. Unterstützend wirken hierbei Konzepte aus dem Bereich der Ergonomie [19], um die Zufriedenheit der Nutzer messen, anpassen und verbessern zu können. Die Erkenntnisse dieses Forschungsbereichs flossen unter anderem in die Entwicklung des Standards ISO9241 (Ergonomics of human-system interaction) ein. Viele HCI-Techniken sind für die Anwendung im Kleinen entwickelt worden, um die Güte vollständiger Interaktionsprozesse anwendungsspezifisch zu maximieren. Häufig lassen sich die Ziele kleiner Anwendungen klar spezifizieren und abgrenzen und auch potentielle Nutzergruppen sind gut bekannt und abbildbar. Insbesondere bei großen Informationsumfängen und einer großen Anzahl von Informationstypen sind jedoch Methoden wie *Paper Prototyping*, *Wireframes* und *Card Sorting* nicht oder nur sehr eingeschränkt nutzbar. Auch *Focus groups* sind aufgrund ihrer Subjektivität nur bedingt geeignet. Die HCI-Techniken können somit bei großen WIS höchstens exemplarisch auf repräsentative Teilbereiche angewandt werden, nicht jedoch erschöpfend.

Um HCI-Techniken auf große Informationssysteme derart übertragen zu können, dass eine content- und benutzungsgesteuerte Präsentation von Inhalten möglich wird, ist die Spezifikation des Layout auf höheren Abstraktionsleveln [65] als dem Implementationslevel erforderlich. So lässt sich beispielsweise ein zu erzielendes Ambiente auf dem

strategischen Level allgemein definieren und schrittweise verfeinern bis dieses auf dem konzeptionellen Level in ein Farbschema überführt werden kann, welches sowohl den Kontext als auch die Wirkung auf konkrete Nutzergruppen berücksichtigt [36]. Dieses Schema kann in der Folge in unterschiedlichen, konkreten Situationen zur Anwendung kommen.

Im Allgemeinen ist es Nutzern möglich, die wahrgenommene Qualität von Webseiten zu bewerten. Berücksichtigung finden können hierbei Faktoren wie die Wirkung der Komposition von Inhalten sowie die Benutzbarkeit und Nützlichkeit der Anwendung. Das Gewicht dieser Faktoren wird dabei in Abhängigkeit von der individuell verschiedenen Wahrnehmung der Anwendung als auch Erwartungen an diese in einer konkreten Situation bestimmt. Von Vorteil ist die Erhebung solch subjektiver Präferenzen und Bedürfnisse nur im Falle kleiner Anwendungen und sehr homogener Nutzergruppen. Geringere Schwankungen in der Bewertung lassen sich erzielen, wenn Faktoren höherer Abstraktion Berücksichtigung finden, da mit steigender Abstraktion der Anteil subjektiver Faktoren sinkt. Dies erlaubt das Ermitteln von Kernfaktoren, um eine geeignete Komposition und Präsentation von Inhalten auch für sehr heterogene Nutzertypen finden zu können.

Aktuell wird der Layout-Entwicklungsprozess großer WIS vor allem von den Erfahrungen der Entwickler und Designer dominiert, da globale Layout-Entwicklungskonzepte nicht existieren. Die Gründe für Gestaltungsentscheidungen sind häufig unbekannt und scheinen zwingend Teil eines künstlerischen Schaffungsprozesses zu sein. Um diese Aspekte der Entwicklung von WIS zugänglich zu machen, müssen die grundlegenden Prinzipien qualitativer Webseiten ermittelt werden. Dies ist unter anderem durch die Analyse existierender Websites möglich, indem nach Ähnlichkeiten und Mustern gesucht und anschließend ihre Anwendbarkeit und Übertragbarkeit hinsichtlich vergleichbarer Problemstellungen überprüft wird. Darüber hinaus lassen sich Erkenntnisse der kognitiven Psychologie und der Gestaltpsychologie [26, 27, 76] aufgreifen und in Gestaltungsrichtlinien überführen.

Die Entwicklung geeigneter und visuell ansprechender Webinformationssysteme (WIS) stellt ein komplexes Problem dar. Derzeit fehlen systematische Konzepte, die sich, beginnend mit intentionalen Definitionen, schrittweise der tatsächlichen Gestaltung auf dem Präsentationslevel nähern. In Abhängigkeit von der Größe der Anwendung werden Aspekte der Gestaltung oft sehr früh oder sehr spät betrachtet, nicht jedoch begleitend. Dies führt zu unflexiblen und eingeschränkt adaptiven Lösungen, wodurch Erweiterungen und Änderungen erschwert werden. Ein frühzeitig entwickeltes Layout wird nur selten der Vielfalt der später darzustellenden, komplexen Contentobjekte in vollem Umfang gerecht, da zum Zeitpunkt der Layoutentwicklung die Darstellungsanforderungen nur bedingt abgeschätzt werden können. Begegnet wird diesem Problem häufig mit der Beschränkung der unterstützten Contentobjekte und verlustbehafteter Transkription der Inhalte. Vor allem die Transkription beeinträchtigt die evolutionäre Adaptionsfähigkeit der Gestaltung, indem beispielsweise Code-Level-Details auf dem konzeptionellen Level der Inhalte verwendet werden. Im Kontrast dazu führt die späte Entwicklung von Layout gerade bei umfangreichen Anwendungen oft zu sehr schlichten Repräsentationsformen, um allen Anforderungen aller komplexen Contentobjekte zu genügen und zugleich die

Erweiterbarkeit sicherzustellen. Weil in dieser Situation keine Probleme bei der prinzipiellen Bereitstellung der Inhalte auftreten, wird die Notwendigkeit der Adaption partiell unterschätzt. Nicht selten wird in der Folge die Anzeige einzelner Inhaltsobjekte separat optimiert, anstatt deren Wirkung und Bedeutung im Zusammenspiel mit anderen Inhalten zu betrachten. Eine typisches Resultat dieser Herangehensweise ist eine schlechte Benutzbarkeit und eine wenig intuitive Orientierung innerhalb der Anwendung.

Die Software- und insbesondere die Datenbankentwicklung nutzen seit längerem die Vorteile der Unterscheidung verschiedener Abstraktionslevel, um die Komplexität großer Anwendungen zu reduzieren und die Flexibilität zu erhöhen. Wenngleich dies bisher nicht erfolgt, so lässt sich auch die Entwicklung der Gestaltung auf verschiedenen Abstraktionsleveln betrachten. Eine solche Abstufung erlaubt erstmalig eine echte Einbindung der Entwicklung der Gestaltung in den WIS-Gesamtentwicklungsprozess. Wenn sich die Entwicklung der Inhalte und Gestaltung gegenseitig beeinflusst ohne dabei den jeweils anderen Entwicklungsprozess zu dominieren, kann eine enge Verzahnung frühzeitig Problemstellen aufdecken und sogar Wechselwirkungen ausnutzen. *Grids* können in diesem Zusammenhang mehrdimensionale Präsentationsbereiche in Teilbereiche teilen, mit dem Ziel die Gestaltung durch Abstraktion zu flexibilisieren und somit die Wiederverwendbarkeit zu erleichtern. Grids sollen sich dabei sowohl an Gestaltungsregeln als auch individuellen Wünschen orientieren und zudem die Zuordnung der späteren Inhalte erleichtern. Die Beziehungen und definierten Kontraste der Teilbereiche eines Grids können ferner die Anwendung von Gestaltungsregeln beeinflussen. Wegen der oft gleichartigen Problemstellungen, treten bei der Entwicklung der Gestaltung Probleme wiederholt in ähnlicher Weise auf. Eine universelle Herangehensweise für eine Problemstellung lässt sich beschreiben, indem das Muster (*Pattern*) einer Lösung formuliert wird. Dies erleichtert die Wiederverwendung bereits entwickelter Konzepte einer Anwendung.

Web-Informationssysteme sind durch den Nutzer gerade dann gut bedienbar, wenn sie sich ihm intuitiv erschließen und ihn bei der Bewältigung seiner Aufgaben unterstützen. Neben den rein funktionalen Anforderungen sind daher die Fähigkeiten, Fertigkeiten und Präferenzen von Nutzern oder Gruppen zu berücksichtigen, um diesen die Informationsaufnahme zu erleichtern. Je detaillierter das Profil und Aufgabenportfolio eines Nutzers umschrieben wird, desto besser ist die erreichbare Adaption. Welche Repräsentation für einen Nutzer am besten geeignet ist, hängt nicht ausschließlich von dessen individuellen Bedürfnissen ab. Oft lassen sich allgemeingültige Regeln finden, wann sich eine Darstellung eignet und wann nicht. Dementsprechend sind, zur Reduzierung des Adaptionaufwands, vor allem die verantwortlichen Parameter zu analysieren, um die Eignung einer Darstellung zu überprüfen. Nur selten kann auf ein vollständiges und permanent aktuelles Profil zurückgegriffen werden, sodass die Präferenzen und Kenntnisse eines Nutzers nur bedingt, gar nicht oder fehlerhaft bekannt sind. In Abhängigkeit von der Datenbasis ist jedoch eine intelligente Erkennung der Nutzer möglich [75]. Der Abgleich bekannter Konzepte und häufig genutzter Pfade mit der tatsächlichen Benutzung durch den Akteur erlaubt sowohl eine Verfeinerung und Korrektur der Datenbasis als auch benutzungsgesteuerte Adaption. Allgemein ist für eine gute Adaptionfähigkeit von Daten deren geeignete Ablage Voraussetzung. WIS-Inhalte werden typischerweise durch den Einsatz von Content Management Systemen (CMS) verwaltet, die über die Struk-

tur, zulässige Möglichkeiten der Komposition und Integration als auch den Ort und die Art der Sicherung entscheiden. Soll auch die Darstellung und Gestaltung der Inhalte zum Zeitpunkt der Ausgabe flexibel entschieden werden, muss die Sicherung der Daten möglichst abstrakt und unabhängig von der späteren Repräsentation erfolgen. Im Idealfall sind somit weder Gestaltungsanweisungen noch Gestaltungshinweise Bestandteil der Contentobjekte. Aufgrund des hohen Aufwands einer vollständigen Trennung auf dem Implementationslevel wird in der Praxis höchstens eine partielle Trennung umgesetzt. Auch content- und benutzungsgesteuertes Layout erfordert eine möglichst umfängliche Trennung von Inhalten und ihrer Darstellung. Der Umsetzungsaufwand lässt sich jedoch beschränken, indem die Entwicklung weitestgehend auf höheren Abstraktionsleveln erfolgt und damit spätestens auf dem konzeptionellen Level die Vielfalt der Darstellungsoptionen stark begrenzt werden kann. Ausgehend vom konzeptionellen Level lassen sich die verbleibenden Optionen nutzen, um beispielsweise durch geringfügige Variationen in der Darstellung die Aufmerksamkeit des Nutzers fortgesetzt aufrecht zu erhalten.

Neben der Ausgestaltung jeweils konkreter Handlungsabschnitte, lässt sich eine Handlung auch insgesamt flexibilisieren. Diese Perspektive führte unter anderem zu *Website Description Languages* wie *SiteLang* [68], aber auch *Task Models* streben eine Generalisierung des Entwicklungsprozesses an. Mit *SiteLang* lassen sich mögliche Handlungen innerhalb der Anwendung beschreiben. Der Kommunikationsprozess entsteht jedoch erst durch das spezifisch verschiedene Zusammenwirken von Stories, Interaktionsoptionen und Kontextbedingungen [65]. *Task models* helfen bei der Planung von Aufgaben, die ein Akteur erledigen soll und sind daher nicht für alle Website-Typen geeignet, z.B. *Identity sites* und *Forum sites*. Sie berücksichtigen, dass der Nutzer in verschiedenen Rollen agieren kann und in verschiedenen Situationen unterschiedliche Aufgaben zu erfüllen hat. *Concur Task Trees (CTT)* [47] bilden als Vertreter der hierarchischen *task models* die Aufgaben eines Nutzers innerhalb eines Baumes ab. Ein solcher Baum erlaubt eine abstrakte Aufgabenplanung für klar voneinander abgrenzbare Teilprozesse. Darüber hinaus ist durch die schrittweise Verfeinerung der Spezifikation über mehrere Ebenen hinweg eine Wiederverwendung von Aufgaben oder Aufgabenteilen innerhalb neuer Systeme möglich. Wenngleich sich prinzipiell Variationen in der Ausführungsreihenfolge spezifizieren lassen, sollten die Aufgaben eines CTT möglichst sequentiell ablaufen, da andernfalls die Übersichtlichkeit der grafischen Darstellung schnell verloren geht. Ferner ist eine flexible Kombination von Prozessschritten nicht vorgesehen, die es erlaubt, innerhalb eines CTT den Ablauf in Abhängigkeit von einer konkreten Situation oder einem Nutzerprofil deutlich zu variieren. Durch eine Erweiterung der CTT um *decision nodes* [30], kann diese Funktionalität jedoch teilweise realisiert werden. Die erreichbare Komplexität von Aufgabensequenzen eines CTT wird durch unäre und binäre Operatoren begrenzt. Für die Erstellung einfacher Formulare und bedingt flexibler Prozesse ist dies ausreichend, im Falle komplexer oder sogar kollaborativer Prozesse kann dies aber die Ausführungszeit des Gesamtprozesses stark verlangsamen. Typischerweise besteht eine Bindung zwischen den spezifizierten Blättern eines *Task Trees* und den Vorstellungen des Interface Designers hinsichtlich der Visualisierung auf einem konkreten Endgerät. Dies ermöglicht zwar eine gerätespezifische Optimierung, begrenzt aber die Eignung für verschiedene Gerätetypen. Die Wiederverwendbarkeit solch spezialisierter Lösungen ist

möglich, indem für diese geeignete Gerätefamilien angegeben werden [30]. Eine derart allgemeine Trennung sichert jedoch nur dann eine korrekte Darstellbarkeit und Nutzbarkeit zu, wenn ausschließlich Fähigkeiten der Schnittmenge aller Geräte einer Familie genutzt werden. Insbesondere im Bereich der mobilen Endgeräte beschränkt sich dies auf ein Minimum, wenn auch ältere Modelle dieser Familie berücksichtigt werden. Zudem bedeutet die Annotation jeder einzelnen Aufgabe einen mitunter nicht unerheblichen Aufwand und zugleich einen hohen Wartungsaufwand, falls Änderungen der Klassifikation nötig werden und große Task Trees betroffen sind. Das Binden der Tasks an Darstellungsanforderungen auf dem Implementationslevel wirkt dem Ziel der Abstraktion und guten Wiederverwendbarkeit entgegen. Eine echt konzeptionelle Spezifikation ist durch vollständige Trennung möglich und ist zudem unabhängig von den Fähigkeiten der Endgeräte bzw. der möglichen Visualisierung auf diesen. Dies setzt jedoch voraus, dass die Taskspezifikation lediglich Aufgaben, Teilaufgaben sowie zulässige Prozesse und Partitionierungen definiert, nicht aber gewünschte Partitionierungen, Positionierungen oder sogar Visualisierungen. Anschließend ist innerhalb einer allgemeinen Adaptionsschicht festzulegen, welche Aufgabe in welche Darstellungsform überführt werden kann. Die tatsächliche Visualisierung erfordert ferner eine Adaption im Speziellen und orientiert sich an den Fähigkeiten des Geräts, der individuellen Benutzung sowie den Präferenzen des Nutzers.

Ein weiteres, besonders häufiges und folgenreiches Problem großer WIS ist die Evolution der Gestaltung. Aufgrund der Nähe der Spezifikation der Gestaltung zum Code-Level und einer typischerweise großen, evolutionären Anreicherung der Darstellungsregeln im Vorfeld eines tatsächlichen Layoutwechsels, ist die umfassende Änderung eines bestehenden Gestaltungskonzepts oft sehr aufwendig. Die Folge sind vollständige Neuentwicklungen in großen aber regelmäßigen Abständen. Die Verlagerung der Layout-Spezifikation auf höher liegende Abstraktionsschichten durch den in dieser Arbeit beschriebenen Ansatz kann diese Problematik vermeiden und den Wartungsaufwand drastisch reduzieren.

1.2 Ziele

Sowohl die Wahrnehmung als auch die Erwartungen eines Nutzers an die Darstellung unterliegen permanenten Änderungen und werden zusätzlich durch kulturelle Prägungen sowie persönliche Präferenzen, Besonderheiten und Kenntnisse beeinflusst. Somit ist ein hohes Maß an Adaptivität erforderlich, wenn eine Anwendung nicht nur ausgewählten Zielgruppen in geeigneter Weise zugänglich sein soll.

Die Screenography führt einen Ansatz für die Entwicklung und Gestaltung des Layout von Websites ein, der die Bedürfnisse der Nutzer in den Vordergrund stellt, unter Beachtung bestehender Präsentationsmöglichkeiten sowie kultureller und individueller Präferenzen. Den Dynamikanforderungen und Erwartungen der Nutzer an die Gestaltung begegnet die Screenography durch Aufgreifen von Erkenntnissen aus Dramaturgie und Szenographie, unter Berücksichtigung der besonderen Anforderungen aber auch Möglichkeiten von WIS. Auf diese Weise verbindet die Screenography Techniken

der Gestaltung und Inszenierung von Handlungsräumen mit Techniken des HCI-Design (Human-Computer-Interface-Design).

Neben der Adaption bedarf es einer wirksamen Abstraktion, um den Entwicklungs- und insbesondere den Wartungsaufwand flexibler, adaptiver Anwendungen beherrschbar zu halten, ohne die Nutzbarkeit und Nützlichkeit der Anwendung auf ausgewähltes Equipment zu beschränken. Die Screenography erlaubt eine weitere Entkoppelung der Bindung zwischen dem Layout und den Inhalten, indem Inhalte ihre Darstellungsanforderungen und Layout-Elemente ihre Darstellungsmöglichkeiten festlegen. Damit lässt sich ermitteln welche Inhaltstypen präsentierbar sind, aber auch eine automatisierte Entscheidung der Darstellung wird möglich.

Den meisten entwickelten Präsentationen liegt ein Raster zugrunde, welches den Gestaltungsbereich mehr oder weniger deutlich erkennbar, aber für den Nutzer deutlich wahrnehmbar in Teilbereiche, Teilverantwortlichkeiten und Teilaufgaben teilt. Dies ist nötig, um die Orientierung innerhalb der Anwendung zu wahren und dem Nutzer die Bedienung der Anwendung zu erleichtern. Wenngleich die Teilbereiche wahrnehmbar sind, so ist typischerweise nicht das zugrunde liegende Raster direkt sichtbar. Die Entwicklung eines Gridsystems trägt im Sinne der Screenography dazu bei, den Präsentationsbereich geeignet zu partitionieren, um die Platzierung und Zuordnung von Inhalten zu erleichtern. Um wiederholt auftretende Problemstellungen gleicher oder ähnlicher Art schneller lösen zu können, werden zudem Patterns entwickelt. Aufgrund der vielfältigen Anwendungsmöglichkeit ist initial eine Patternklassifikation nötig, um die verschiedenen Bereichen der visuellen Gestaltung unterscheiden zu können.

Das Ziel dieser Arbeit ist das Entwickeln der Grundlagen und der Technologie für ein adaptives, content- und benutzungsgesteuertes generisches Layout. Hierfür sind unter anderem die Platzierung, der Umfang als auch die Darstellung der Inhalte an die Bedürfnisse verschiedener Nutzer und Situationen anzupassen, ohne dabei mehrere parallele Layout-Entwicklungsprozesse initiieren und verwalten zu müssen. Vielmehr sollen die Fähigkeiten des Equipments, der Nutzer und der Inhalte, unter Beachtung der zu erreichenden Ziele, den möglichen Gestaltungs- bzw. Präsentationsraum definieren und eingrenzen. Eine weitere Eingrenzung kann durch die Berücksichtigung von Präferenzen und Bedingungen seitens der Nutzer und Provider erfolgen.

Durch das Sichern von Inhalten in einer möglichst generischen Form, wird die Auspielflexibilität erhöht und dadurch eine Content-Steuerung realisierbar. Die Komposition zu und Separation in adäquate Mengeneinheiten kann durch generische Sicherung flexibel realisiert werden. Contentgesteuerte Layout-Entwicklung vermeidet insbesondere Repräsentationsfehler, da weder der Gestalter über die möglichen Darstellungen konkreter Inhalte entscheidet noch der Redakteur, der diese einstellt. Allein die Kriterien der Rahmenspezifikation sollen den Inhaltstyp von Inhalten bestimmen. Je nach gewähltem Layout können im späteren Realisierungsschritt mögliche konkrete Darstellungsmöglichkeiten abgeleitet werden.

Benutzungsgesteuertes Layout passt sich sehr schnell neuen Situationen an, indem die Darstellung von Inhalten von dem tatsächlichen Verlauf der Story abhängig gemacht wird. Ein derart erzeugtes Layout weist ein Höchstmaß an Dynamik auf und lässt sich an individuelle Bedürfnisse anpassen. Zugleich werden Variationen bezüglich der Präsentation möglich, um die Aufmerksamkeit des Nutzers auch längere Zeit aufrecht erhalten zu können, ohne ihn durch Beliebigkeit oder ungeeignete Repräsentationen zu irritieren.

Die entwickelten Grundlagen sollen darüber hinaus innerhalb einer flexiblen Basis-Architektur zur Anwendung kommen. Die Flexibilität dieser Architektur wird durch 6 Dimensionen erreicht, deren Einfluss in Abhängigkeit von den Anforderungen bestimmt werden kann und ermöglicht Inhalte entsprechend der individuellen Bedürfnisse des jeweiligen Nutzers und unter Beachtung der Darstellungsmöglichkeiten zu präsentieren.

2 Der Entwicklungsprozess von WIS

Im Vorfeld der Entwicklung einer Modellierungssprache für die Präsentation, gilt es deren Aufgaben und Anforderungen zu klären. Allgemein lassen sich die 3 Bereiche Darstellung, Integration und Adaption unterscheiden. Der Bereich der Darstellung betrifft insbesondere das 'Wie' des Präsentierens und erfordert Regelungen für die Raster, Rahmen und Container-Entwicklung, auf die folgend eingegangen werden. Die Integration umfasst hingegen das 'Was' der Präsentation, wozu nicht nur die Inhalte zählen, sondern auch Funktionalität. Der Bereich der Adaption soll schließlich sicherstellen, dass die Darstellung der Inhalte auch die Bedürfnisse der Nutzer berücksichtigt. Ferner soll das Aufgabenportfolio, die Entwicklung der Story sowie existierende Kontextbedingungen in die Entscheidungsfindung einfließen.

Dieses Kapitel führt zunächst 6 Dimensionen ein, die das Fundament der Entwicklung von WIS bilden. Hierzu werden der Nutzen, die Aufgaben und Teilbereiche der einzelnen Dimensionen diskutiert, aber auch Problemstellungen, die die Entwicklung der Präsentation beeinflussen. Anschließend wird das Zusammenwirken der Dimensionen erörtert, da gegenseitige Abhängigkeiten zwischen den Dimensionen existieren und deren Gewichtung den Generierungsprozess entscheidend beeinflussen können. Schließlich soll anhand ausgewählter Website-Typen verdeutlicht werden, dass unterschiedliche Zielvorgaben und Anforderungen der Anwendung eine geänderte Entwicklungsreihenfolge der Dimensionen erfordern können.

2.1 6 Dimensionen der Entwicklung von WIS

Die Entwicklung von Informationssystemen und damit auch die Entwicklung des Layout wird durch eine Vielzahl von Parametern und Bedingungen beeinflusst. Eine bessere Überschaubarkeit der existierenden Stellgrößen lässt sich erreichen, indem allgemeine Dimensionen gefunden werden, denen die ermittelten Einflussfaktoren zugeordnet werden können. Diese Arbeit basiert auf der Unterscheidung von 6 allgemeinen Entwicklungsdimensionen nach [55] und nutzt diese für die Entwicklung des Layout, dargestellt durch Abbildung 2.1. Die Verwendung aller Dimensionen soll dabei verdeutlichen, dass die Entwicklung des Layout nicht allein Gegenstand der Präsentationsdimension ist oder sein muss.

Die folgenden Kapitel führen in diese 6 Dimensionen kurz ein und erläutern ihre Bedeutung für die Layout Entwicklung.

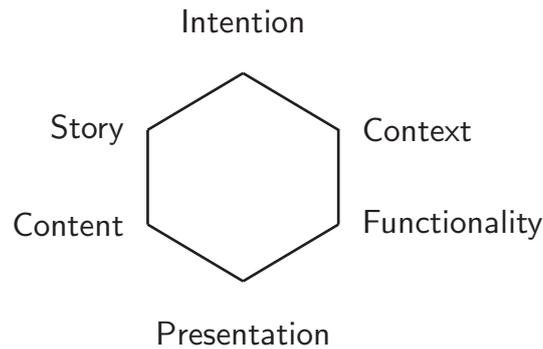


Abbildung 2.1: 6 Dimensionen der WIS Entwicklung

2.1.1 Intention

Die Aufgabe der *Intention*-Dimension ist es, den Typ der zu entwickelnden Anwendung abstrakt zu charakterisieren. Dazu werden Ziele, Absichten und Visionen analysiert und in einen geeigneten Anwendungstyp überführt.

Durch die Spezifikation der Ziele kann die Bedeutung von Einflussfaktoren ermittelt werden, die die globale Ausprägung der Anwendung bestimmen. Unterstützend wirkt hierbei die Definition des Ambiente, von dem entwicklungsbegleitend schrittweise der konkrete Style abgeleitet wird. Das Ambiente wird durch die Intention-Dimension abstrakt definiert, um allgemeingültig die beabsichtigte Wirkung von Farben, Formen und Kompositionen beschreiben zu können, sodass auf diesem Abstraktionsniveau subjektive, regionale oder religiöse Wahrnehmungsunterschiede nicht existieren. Erst mit der Überführung des Ambiente in eine konkrete Ausprägung werden subjektive und lokale Gegebenheiten berücksichtigt. Den größten Einfluss auf die Ermittlung eines geeigneten Anwendungstyps besitzen die Profile und Portfolio [54] der Nutzer und des Providers. Aus den Profilingaben eines Nutzers geht einerseits hervor, welche Interaktions- und Präsentationsformen bevorzugt werden und welche nicht verwendbar sind, z.B. infolge bestehender Wahrnehmungsdefizite. Andererseits definieren Profile den Grad der möglichen Adaptivität. Je nach Anwendungstyp kann eine weitreichende Individualisierung oder strikte Einheitlichkeit erstrebenswert sein. In Abhängigkeit von den jeweiligen Arbeitsaufgaben muss der Nutzer durch eine geeignete Arbeitsumgebung bei der Erledigung dieser unterstützt werden. Daher ist auch das Portfolio eines Nutzers bereits frühzeitig in den Entwicklungsprozess einzubinden, um den generellen Typ der Arbeitsumgebung zu definieren.

2.1.2 Story

Die Wirkung von Inhalten, welche der Informationsvermittlung oder -aufnahme dienen, ist stark an die Art deren Einbettung in die Handlung des zugrunde liegenden Werks gebunden. Aufgrund dieser Abhängigkeit ist die Leistungsfähigkeit und Flexibilität der Story Dimension von größter Bedeutung. Dennoch spezifizieren derzeitige, dateninten-

sive Webanwendungen Handlungen typischerweise nicht explizit und präsentieren die Inhalte entsprechend ihrer Ablagestruktur. Die starke Bindung an die Ablagestruktur der Inhalte erlaubt dabei die Spezifikation von Handlungsalternativen nur sehr eingeschränkt. Durch Entkopplung und explizite Spezifikation von Handlungsoptionen können auch individuelle Herangehensweisen von Nutzern abgebildet werden. Nur selten wird jedoch eine einzige Repräsentationsform solch individuell verschiedenen Handlungen in vollem Umfang genügen. Sollen auch die Bedürfnisse der Nutzer sowie die Anforderungen eines konkreten Handlungspfads Berücksichtigung finden, so ist die Entwicklung eines benutzungsgesteuerten Layout notwendig.

Der Begriff *Story* wird unter anderem im Bereich der Filmproduktion (Story (Film)) verwendet und beschreibt dort die festgelegte Abfolge einer Handlung. Die *Story*, so wie diese hier verwendet wird (Story (Screen)), orientiert sich stark an dieser Semantik und legt fest, in welcher Weise und auch Reihenfolge dem Nutzer die Inhalte einer Web-Anwendung präsentiert werden können und dürfen. Erweiternd schließt der hier verwendete Story-Begriff die Spezifikation von Handlungsalternativen ein, sodass eine konkrete Handlung an die aktive Mitwirkung und Entscheidung eines Akteurs gebunden werden kann. Stellvertretend kann ein Akteur durch ein System emuliert werden, welches Entscheidungen auf der Grundlage von Entscheidungskriterien und -logiken trifft.

Die Entwicklung jeder Story erfolgt innerhalb eines Story-Raums, der als Rahmen die Befugnisse der Erstellung bestimmt und begrenzt. Um verschiedene Handlungsverläufe spezifizieren zu können, lassen sich innerhalb eines Story-Raums mehrere Stories entwickeln. Wir nutzen die in [65] eingeführte Definition eines Story-Raums, erlauben jedoch erweiternd eine zentrale Beschränkung der Sicht (V) auf die Datenbasis, auf welche eine innerhalb des Story-Raumes definierbare Story zugreifen kann. Ferner werden Spezifikationsregeln $Spec_R$ zur Steuerung der Entwicklung von Stories erlaubt, um die Spezifikation auf ausgewählte Handlungsverläufe im Kleinen und Großen beschränken zu können. Dadurch kann die Storyentwicklung nicht nur von den Spezifikationsmöglichkeiten an Entscheidungspunkten abhängig gemacht werden (im Kleinen), sondern auch vom bisher entwickelten Handlungsverlauf, komplexen Verlaufsmustern sowie von anderen Stories des Story-Raums (im Großen).

$$\begin{aligned}
\text{Story-Raum} &= (\{\text{Szene}\}, E, \lambda, \kappa, V, Spec_R) \\
E &\subseteq \{\text{Szene}\} \times \{\text{Szene}\} \\
\lambda &: \{\text{Szene}\} \rightarrow \text{SzeneBeschreibung} \\
\kappa &: E \rightarrow \text{TransitionBeschreibung} \\
&\quad \text{TransitionBeschreibung} \subseteq \\
&\quad (\text{Ereignisse} \cup \text{Aktivitäten}) \times \text{Akteure} \times \text{Vorbedingungen} \times \\
&\quad \text{Nachbedingungen} \times \text{Priorität} \times \text{Häufigkeit} \times \text{Wiederholrate} \\
V &\subseteq \text{Content} \\
Spec_R &\subseteq \{\kappa \times \dots \times \kappa\}
\end{aligned}$$

Story-Spezifikationen sollten sich an den Aufgaben und Bedürfnissen der Nutzer orientieren, mit dem Ziel den Akteuren gezielt nur jene Inhalte zur Verfügung zu stellen, die diese zur Erledigung ihrer Aufgaben benötigen. Wird ferner der Umfang der zu

präsentierenden Inhalte entsprechend dem Nutzer-Profil bzw. den Fähigkeiten des jeweiligen Akteurs in einer konkreten Situation gewählt, lassen sich Überlastungssituationen vermeiden, die durch Überfrachtung der Anwendung mit Inhalten entstehen.

Szenen

Szenen sind logisch zusammengehörige Abschnitte einer Handlung, die ihrerseits aus weiteren Szenen bestehen können. Eine Szene beschreibt einen Zeitpunkt oder eine Zeitspanne innerhalb des Story-Raums. Jeder Szene lassen sich Contentobjekte als Teil einer Content-Suite zuordnen, die für die Content-Kollektion (Input), Content-Verarbeitung (Computation) und Content-Propagation (Output) von Inhalten benötigt werden. Nach Thalheim [65] besteht eine Szene aus folgenden Komponenten.

Szene = (SzeneID,
 DialogSchrittAusdruck,
 Content-Suite,
 Akteure (
 AkteurID,
 Rechte,
 Aufgaben (
 Zuordnung,
 Rolle)),
 Repräsentation (Stil, Defaultwerte, Betonung, ...),
 Kontext (Hardware, Software, Kanal, Intention))

In Anlehnung an diese Gliederung lässt sich eine Szene als folgendes 6-Tupel beschreiben (Gl. 2.1).

$$Sc = (sc_{id}, sc_{constraints}, C_{suite}, A, R, K) \quad (2.1)$$

Da Szenen aus Teilszenen bestehen können, ist die Zugänglichkeit von Contentobjekten der Elternszenen innerhalb ihrer Kindszenen zu entscheiden. Bei restriktiver Vorgehensweise wird neuen Teilszenen der Zugriff auf Contentobjekte der Elternszenen stets verwehrt. Dies ermöglicht eine kontrollierbare Szenenentwicklung, verursacht aber einen hohen, administrativen Aufwand beim Erzeugen von Substrukturen, wenn zu Beginn der Entwicklung deren inhaltlicher Bedarf nicht jedoch deren Struktur und Tiefe feststeht. Alternativ kann neuen Teilszenen der Zugriff auf die Contentobjekte von Elternszenen vollständig gewährt werden. Die ausschließliche Erweiterbarkeit führt allerdings mit zunehmender Anzahl von Verfeinerungsebenen auch zu einer immer größeren und schwerer überschaubaren Anzahl von Contentobjekten. Daher ist die Kombination beider Ansätze sinnvoll.

Folgend wird die Zugänglichkeit erlaubt (Def. 1), um den Aufwand der expliziten Contentobjektzuordnung für jede Kindszene gerade dann zu vermeiden, wenn alle Kindszenen auf die Contentobjekte der Elternszene zurückgreifen können sollen und keine erweiternde Sicht benötigen. Eine flexibilisierende Ausnahme von dieser Regel bildet

der explizite und lokale Entzug von Zugriffsrechten auf konkrete Contentobjekte. Jede Elternszene legt somit das Standard-Spezifikationspotential ihrer Kindszenen fest und kann dadurch die Entwicklung und Weiterentwicklung auf einfache Weise steuern.

Auch ohne den expliziten Entzug von Rechten ist eine nachträgliche Beschränkung möglich. Dies lässt sich durch Einfügen einer neuen und beschränkten Geschwister-Szene auf dem Level der Elternszene erreichen. Teilszenen, denen nicht alle bisherigen Contentobjekte zugänglich gemacht werden sollen, werden dabei in die neue Szene verschoben.

Definition 1. Sei \mathcal{C} eine Menge von Content-Objekten und \mathcal{S} eine Menge von Szenen. Ferner gelte $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{S}$. Seien $s_1, s_2 \in \mathcal{S}$ und \mathcal{R} gegeben. s_2 ist Teilszene von s_1 (i.Z. $s_2 \downarrow s_1, s_1 \uparrow s_2$) gdw. $(\forall c \in \mathcal{C})(s_1, c) \in \mathcal{R} \Rightarrow (s_2, c) \in \mathcal{R}$

Szenen stellen die Knoten des Handlungspfades dar und lassen sich im Kontext einer Storyspezifikation durch gerichtete, bewertete Kanten verbinden. Eine Story lässt sich damit als eine Menge verbundener Szenen (Gl. 2.2) beschreiben.

$$S = (\mathcal{S}, \prec) \quad (2.2)$$

Jede Szene kann Eingaben erfordern und Ausgaben produzieren, wobei die Ausgaben von den Eingaben abhängen können und umgekehrt (Abb. 2.2). Eingabeseitig sind daher Objekte oder Attribute, einschließlich ihrer Datentypen und deren Wertebereiche, zu definieren, die die Szene erwartet ($I_{expected}$) und die ihr optional ($I_{optional}$) übermittelt werden können (Vorbedingungen der Szene). Optionale Angaben können sich dabei sowohl neutral gegenüber einer Szene verhalten, indem höchstens deren Weitergabe an Folgeszenen gestattet wird, als auch das Szenengeschehen beeinflussen, indem sie Ausgaben oder den Verarbeitungsprozess sowie dessen Effektivität oder Effizienz beeinflussen. Ergänzend lässt sich mit Hilfe von Constraints definieren, unter welchen Bedingungen Eingaben von einer Szene angenommen werden dürfen. Dies ist erforderlich, da für zwei Attribute a_1 und a_2 in Bezug auf die Eingabemenge I die Bedingung $a_1 \in I \Leftrightarrow a_2 \notin I$ gelten kann. Kann ein Attribut oder Objekt potentiell von mehreren beteiligten Vorgängerszenen bereitgestellt werden, ist eine bewusste Auswahl zu treffen, solange nicht die Gleichheit aller existierenden Alternativen garantiert werden kann. Grundlage dieser Auswahl ist die Möglichkeit der eindeutigen Identifikation von Szenen. Existieren Schnittstellen zu externen Anwendungen und Szenen oder dem Nutzer, ist dies gegebenenfalls nur durch Authentifikation sicherzustellen. Analog der Eingabeseite, gilt es festzulegen, welche Ausgaben eine Szene produzieren kann und darf (Nachbedingungen der Szene). Da die Nützlichkeit der Ausgabe einer Szene oft von den Anforderungen und Möglichkeiten einer Folgeszene abhängt, ist die Adaptivität dieser Schnittstelle von großer Bedeutung. Allgemein kann die Rolle einer Szene variieren. Dadurch kann nicht nur die Übergabe von Werten ($O_{transduction}$), sondern auch das Zusammenwirken der Szene mit Anderen situationsabhängig verschieden sein.

Unter Berücksichtigung der Ein- und Ausgaben durch und für Nutzer oder externen Anwendungen sowie Szenen, umfasst die die Eingabeerwartung einer Szene immer mindestens die Eingabeerwartungen aller ihrer Teilszenen, abzüglich der Erwartungen jener

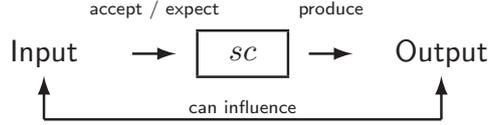


Abbildung 2.2: Szenenübergang

Teilszenen, die durch Ausgaben von Vorgänger-Teilszenen oder der Szene selbst befriedigt werden können (Gl. 2.3). Analog entspricht die Ausgabe einer Szene mindestens den Ausgaben aller ihrer Teilszenen, die ihre Daten an die Szene, den Nutzer oder externe Anwendungen zur Ausgabe weitergeben (Gl. 2.4).

$$I_{Sc_a} \supseteq \bigcup_{Sc_i \downarrow Sc_a} I_{Sc_i} \setminus I_{Sc_n} \quad (2.3)$$

$$\begin{aligned} I_{Sc_n} &\subseteq O_{Sc_m} \\ Sc_m &\prec Sc_n \\ ((Sc_m \wedge Sc_n) \downarrow Sc_a) &\vee ((Sc_m = Sc_a) \wedge (Sc_n \downarrow Sc_a)) \end{aligned}$$

$$O_{Sc_b} \supseteq \bigcup_{Sc_i \downarrow Sc_b} O_{Sc_i} \setminus O_{Sc_p} \quad (2.4)$$

$$\begin{aligned} O_{Sc_p} &= I_{Sc_q} \\ Sc_p &\prec Sc_q \\ ((Sc_p \wedge Sc_q) \downarrow Sc_b) &\vee ((Sc_q = Sc_b) \wedge (Sc_p \downarrow Sc_b)) \end{aligned}$$

Der Übergang (Transition) von einer Szene zu einer Folgeszene wird innerhalb einer Story durch gerichtete Verbindungen definiert (Gl. 2.5). Eine solche Verbindung kann an Regeln geknüpft werden, um beispielsweise einer eingeschränkten Nutzergruppe den Übergang zu erlauben. Dies beinhaltet auch die Möglichkeit, die tatsächlich zu übertragenden Ausgaben einer Szene einzuschränken.

$$T_{Sc_1, Sc_2, constraints}(S) : Sc_1 \xrightarrow{constr.} Sc_2 \quad (2.5)$$

Eine gültige Verbindung zwischen einer Szene Sc_1 und einer Folgeszene Sc_2 entsteht nur dann, wenn die Ausgabe-Contentobjekte von Sc_1 zusammen mit möglichen Nutzereingaben mindestens die Eingabeerwartungen von Sc_2 erfüllen kann (Gl. 2.6).

$$\begin{aligned} &Contentobject \mathcal{C} \\ Input \ I &= I_{expected} \cup I_{optional} \\ Output \ O &= O_{presentation} \cup O_{transduction} \end{aligned}$$

$$Sc_1 \prec Sc_2 :\Leftrightarrow C_{O_{transduction}}(Sc_1) \wedge C_{I_{user}} \supseteq C_{I_{expected}}(Sc_2) \quad (2.6)$$

Erweiternd kann als Transitionstyp die Verwendung von *Joins* erlaubt werden, um das Erreichen einer Szene von der vollständigen Ausführung mehrerer, zusammenwirkender Vorgängerszenen Sc_i abhängig zu machen. Mitunter muss dann festgelegt werden, welche Vorgängerszene welche Ausgaben liefern muss und darf. Kann ein Parameter von mehreren Vorgängerszenen geliefert werden, ist daher eine exakte Zuordnung oder aber die Definition einer Hierarchie erforderlich, sobald dieser in Abhängigkeit von der Vorgängerszene variieren kann. Eine gültige Verbindung entsteht genau dann, wenn die Join-Vorgängerszenen mindestens alle Eingabeerwartungen der Szene erfüllen können (Gl. 2.7).

$$Sc_i \prec Sc_2 :\Leftrightarrow \bigcup C_{O_{transduction}}(Sc_i) \wedge C_{I_{user}} \supseteq C_{I_{expected}}(Sc_2) \quad (2.7)$$

Soll das parallele Auslösen mehrerer Folgeszenen unterstützt werden, sind als Transitionstyp *Splits* zu erlauben. Einschränkend lässt sich dabei spezifizieren, welcher Folgeszene welche Ausgaben der Szene zur Verfügung gestellt werden dürfen. Eine gültige Verbindung entsteht dann, wenn für jede Folgeszene oder Folgeoperation (im Falle sich direkt anschließender Joins) mindestens deren Eingabeerwartungen erfüllt werden (Gl. 2.8).

$$Sc_1 \prec Sc_i :\Leftrightarrow C_{O_{transduction}}(Sc_1) \wedge C_{I_{user}} \supseteq \bigcup C_{I_{expected}}(Sc_i) \quad (2.8)$$

2.1.3 Context

Die *Context*-Dimension betrifft eine Vielzahl von nicht-statischen Randbedingungen innerhalb und außerhalb der Anwendung, die sich temporär ändern können und dadurch das Aufnehmen, Verarbeiten und Verwenden der präsentierten Inhalte erschweren oder unmöglich machen. In Abhängigkeit von der Signifikanz der Änderungen und den verfügbaren Interaktionsschnittstellen können derartige Parameterdifferenzen zwischen dem Zeitpunkt der Entwicklung und dem Zeitpunkt der Nutzung die gesamte Systembenutzung verändern oder unterbinden.

Der wichtigste Bereich der *Context* Dimension ist die Verwaltung der Rechte der Nutzer des WIS, da sehr oft nicht alle Informationen jedem beliebigen Nutzer zugänglich sein sollen. Zur Verwaltung der Rechte haben sich 3 Konzepte etabliert. Beim Konzept der Discretionary Access Control (DAC) entscheidet der Besitzer, welche anderen Nutzer auf seine Objekte in welchem Umfang zugreifen dürfen. Da diese Nutzer ihrerseits die erhaltenen Rechte oder einen Teil dieser Rechte ohne Kontrolle weiterreichen können, kann mit diesem Konzept die Vertraulichkeit nicht garantiert werden. Eine Abschwächung dieses Konzepts und Erleichterung der Rechtevergabe entstand durch das Einführen von Gruppen. Eine weitere Abschwächung mündete im Konzept der Role Based Access Control (RBAC) [49]. Dieses Konzept ordnet Nutzer nicht direkt Gruppen bzw. Nutzerrechten zu, sondern Rollen, in denen sie aktiv sein können. Stellvertretend werden den Rollen die Gruppen zugeordnet, um Rechte zu erteilen. Der Unterschied zu einer Gruppenshierarchie besteht im nicht-kumulativen Verhalten der Gruppenrechte

aller Rollen, falls ein Nutzer mehreren Rollen zugeordnet wurde. Die Gruppenrechte können nur dann vom Nutzer wahrgenommen werden, wenn sich dieser innerhalb der Rolle befindet, der die Gruppe zugeordnet wurde. Der Nutzer kann somit in Abhängigkeit von der Rolle mitunter stark unterschiedliche Gruppenrechte wahrnehmen. Ein Vorteil des rollenbasierten Zugriffs besteht darin, dass Rechte die dem Nutzer in einer Rolle A zugestanden werden, nicht automatisch in allen anderen Rollen verfügbar sind. Ohne ein Rollenkonzept müssen einem Nutzer mit verschiedenen Aufgaben oft deutlich mehr globale Rechte eingeräumt werden als gewünscht. Andernfalls müssten mehrere Accounts pro Nutzer vorgehalten werden, um verschiedenste Aufgaben wahrzunehmen. Die Mandatory Access Control (MAC) wird als drittes Konzept hauptsächlich in Bereichen starrer, hierarchischer Strukturen verwendet, z.B. als Multi-Level-Security. Ziel dieses Konzeptes ist der Schutz vor unauthorisiertem Zugriff, der mit DAC nicht ausgeschlossen werden kann, und partiell die Sicherung der Integrität. Bei MAC entscheiden im Gegensatz zum DAC globale Regeln und Eigenschaften darüber, auf welche Objekte von welchen Nutzern zugegriffen werden kann. Der Nutzer wird einer Schutzstufe und gegebenenfalls Verbänden innerhalb seiner Schutzstufe zugeordnet, kann über den Zugriff auf Objekte aber typischerweise nicht direkt selbst entscheiden. Im Bereich bisheriger WIS werden überwiegend um Gruppen erweiterte DAC-Konzepte eingesetzt. Noch zu selten wird im Web bisher der RBAC Ansatz verfolgt, obwohl gerade bei diesen Systemen sehr oft unterschiedliche Funktionen von einer Person wahrgenommen werden. Allgemein muss das Konzept des Zugriffs vor allem zur Anwendung passen, wodurch auch das MAC Konzept nicht ausgeschlossen werden sollte, wenngleich das potentielle Anwendungsspektrum beim RBAC Ansatz wesentlich breiter ist.

In Abhängigkeit von der Anwendung und aktuellen Kommunikationssituation können verschiedenste Kommunikationsengpässe auftreten, weshalb die QoS (Quality of Service) ebenfalls Bestandteil der Context Dimension ist. Welche Güte der Kommunikation noch akzeptabel ist, hängt von den Anforderungen des Nutzers ab. Werden die Content-Objekte hinsichtlich ihrer Kommunikationsanforderungen annotiert und analysiert das System fortwährend den Kommunikationskanal hinsichtlich gebotener Dienstgüte, so kann proaktiv die Aufbereitung der Inhalte erfolgen, entsprechend der Anforderungen des Nutzers und der aktuellen Situation. Einhergehend kann der Nutzer informiert werden, wenn bestimmte Dienste nicht oder nur mit Einschränkungen erbracht werden können. Gegebenenfalls kann die Darstellung auf Zusatzinformationen verzichten, um die Lauffähigkeit der Hauptanwendung entsprechend der Nutzerforderungen zu gewährleisten.

Ein weiterer Bestandteil der Context Dimension sind dynamisch veränderliche Parameter, welche die Wahrnehmung der Anwendung beeinflussen können, obwohl sie kein direkter Bestandteil der Anwendung sind. Die Umgebungsparameter können die Benutzung der Anwendung einschränken und unterbinden, wenn sich während der Benutzung starke Abweichungen zu den angenommenen oder verwendeten Werten zum Zeitpunkt der Entwicklung ergeben. Bei sich stark verändernden Einsatzbedingungen kann eine explizite Berücksichtigung solcher Parameter zwingend notwendig werden, um die Informationsausgabe an die jeweilige Situation anpassen und dem Nutzer adäquat wahrnehmbare Informationen präsentieren zu können. Ein Beispiel hierfür sind starke Schwankungen der

Helligkeit oder Lautstärke der Umgebung, die unter anderem zu Überdeckungseffekten führen können. Die Sensorik zum Erkennen derartiger Veränderungen ist verfügbar und auch integrierbar. Da die Lautstärke der Anwendung oft über das Level der Umgebung angehoben werden kann, ist in diesem Falle das direkte Verhindern von Überdeckungen möglich. Der Adaption der Helligkeit sind mit diesem Ansatz jedoch enge Grenzen gesetzt, da die Helligkeit des Displays zumeist nur geringfügig verändert werden kann und zudem ein Überstrahlen verhindert werden muss. Auch der Wechsel des Kommunikationskanals ist nicht in jeder Situation möglich, da z.B. private oder sicherheitskritische Informationen nicht auditiv zugänglich gemacht werden sollen bzw. dürfen. Helligkeitsprobleme lassen sich aber auch indirekt reduzieren indem die Lesbarkeit verbessert wird, z.B. durch eine vergrößerte Darstellung. Handelt es sich um sicherheitskritische Informationen ist hingegen der Adaption des Farbschemas (Kontrasterhöhung, Farbreduktion) Vorrang zu gewähren.

Neben den Umgebungsparametern gilt es die Fähigkeiten des Equipments bzw. die der Kommunikationsgrundlage zu berücksichtigen. Zum Equipment zählen sowohl die verwendete Hardware als auch die verfügbare Software. Bei der Hardware soll zwischen den Endgeräten und dem Kommunikationskanal unterschieden werden. Der Kommunikationskanal ist insbesondere bei WIS von Bedeutung, da sich bei diesen die Informationen nicht auf dem lokalen System befinden. Für das Benutzen der Anwendung sind daher die Eigenschaften des Kommunikationskanals von Bedeutung, z.B. die zur Verfügung stehende Bandbreite, die Ausfallsicherheit und existierende Latenzen. Aufgrund der Vielzahl von Kommunikationstypen ist bei den Endgeräten insbesondere der Endgerätetyp zu berücksichtigen. Im Falle eines Displays ist in der Regel die Farbfähigkeit, die Auflösung und Größe von Bedeutung. Die Software eines Systems ist immer dann von Bedeutung, wenn Inhalte existieren, die spezielle software-spezifische Anforderungen besitzen, um die Präsentation der Inhalte zu ermöglichen. Je genauer die Fähigkeiten des Equipments bekannt sind, desto besser können die Inhalte an dieses adaptiert werden, ohne es zu über- oder unterfordern. Dies bedeutet beispielsweise für mobile Endgeräten, dass so wenig wie möglich rechenintensive Operationen auf diesen durchzuführen sind, um die Antwortzeiten und Akkubelastung gering zu halten.

Der Anwendungskontext ist ebenfalls Bestandteil der Context Dimension und beeinflusst in welcher Weise, welcher Geschwindigkeit und welchem Umfang die Daten der Anwendung durch den Nutzer intuitiv oder explizit aufgenommen (Wahrnehmung, Sicherung), verarbeitet (Evaluierung, Selektion) und verwendet (Nutzung) werden. Typischerweise sind nicht alle Parameter statisch, sondern können sich dynamisch ändern. Zu den nicht direkt ermittelbaren aber hoch dynamischen Parametern zählt die Aufmerksamkeit, die der Nutzer dem System bzw. der Anwendung entgegenbringen kann. Bei Navigationsgeräten ist zum Beispiel von Bedeutung, ob man während der Benutzung fährt oder steht, sich innerhalb eines Fahrzeugs befindet oder mit dem Rad oder zu Fuß unterwegs ist, wie schnell man sich fortbewegt, wie hoch das Verkehrsaufkommen, wie breit und sicher die Straße aber auch wie gut ihr Zustand ist. Der Detaillierungsgrad der Darstellung sollte während der Fahrt von der Zeit abhängig gemacht werden, die dem Nutzer zur Verfügung steht, um sich zu entscheiden. Ferner ist es auf ein individuelles Minimum zu beschränken, um die Aufmerksamkeit des Fahrers nicht zu stören. Somit

darf der Detaillierungsgrad, einschließlich möglicher Zusatzinformationen, bei freier und kontinuierlicher Fahrt auf einer Autobahn oder während einer längeren Stillstandsphase deutlich höher sein als bei Fahrten innerhalb einer Stadt. Allgemein ist für die Wahl des Detaillierungsgrades einer Anwendung auch der Zweck der Nutzung zu berücksichtigen, da eine Anwendung unterstützend, empfehlend aber auch instruierend aktiv sein kann. Bei sicherheitskritischen Anwendungen ist es sinnvoll, den Anwendungskontext explizit vorzugeben, damit ein Nutzer jederzeit in der Lage ist, mit dem System bzw. der Anwendung zu interagieren. Zugleich kann festgelegt werden, welche Interaktionsformen in welchen Situationen zulässig sind. Im Zusammenhang mit der Generierung der Informationsausgabe stellen die so erzeugten Regeln die Grenzen der zulässigen Adaption dar.

2.1.4 Content

Der Content ist der wichtigste Teil eines Informationssystems und die Absicht diesen zu kommunizieren, ihn anderen Nutzern zugänglich zu machen oder ihn zu erheben, ist allgemein der Grund für den Aufbau eines solchen Systems. Die Wirkung und Wahrnehmung kommunizierter Inhalte ist subjektiv, da Nutzer selten über identische Profile verfügen und sich somit Fähigkeiten, Fertigkeiten, Erfahrungen und Präferenzen unterscheiden. Die Bedeutung der Content Dimension für die Entwicklung der Präsentation ergibt sich daher insbesondere aus der Verschiedenheit der Nutzer. Die Eignung einer Darstellung hängt zudem nicht unwesentlich von dem Typ der Anwendung ab, da jeder Anwendungstyp eine Erwartungshaltung beim Nutzer hervorruft. Ferner lässt sich nicht jeder Inhalt in beliebiger Form präsentieren, da die Darstellungsfähigkeiten des Equipments beschränkt sein können und sich z.B. multimediale Inhalte nicht generisch in eine statische Text- oder Bildform konvertieren lassen. Ein System wird diesen unterschiedlichen Anforderungen nur dann gerecht, wenn es genügend Flexibilität aufweist, um die Informationsausgabe an die Bedürfnisse der Nutzer in der jeweiligen Situation anzupassen. Die geforderte Flexibilität lässt sich partiell erreichen, wenn den Content beeinflussende Stilmittel unabhängig von diesem gesichert werden. Das Generieren dynamisch adaptiver Ausgaben verlangt aber nicht nur Stilmittel-Suiten, die je nach Situation ad-hoc auf die Inhalte angewendet werden können, sondern Stil-Annotation auf konzeptioneller Ebene, da eine adaptive Ausgabe andernfalls nur auf ähnliche Endgerätetypen beschränkt ist. Auch der Content sollte in einer Equipment-unabhängigen Form gesichert werden, um Adaptionprobleme zu vermeiden. Neben der direkten Präsentation ist auch die Struktur bzw. Gliederung des Content von Bedeutung, da beispielsweise ein Formular aus zwei Teilabschnitten bestehen kann. Sobald nicht genügend Platz für die Anzeige des Gesamtformulars zur Verfügung steht, kann eine solche Information verwendet werden, um die Abschnitte des Formulars separat zu präsentieren.

2.1.5 Functionality

Die *Functionality*-Dimension von WIS wird häufig vor allem durch die Navigation und Interaktion geprägt. Das zeitgleiche Präsentieren aller Inhalte eines WIS ist in der Re-

gel unmöglich, da der durch das Equipment zur Verfügung gestellte Darstellungsplatz beschränkt und der Umfang der Inhalte oft immens ist. Zudem soll vermieden werden, dass der Fokus und die Übersichtlichkeit der Darstellung verloren geht, infolge zu großen Umfangs der Inhalte und Themen. Trotz dieser Beschränkungen kann die Erreichbarkeit der Inhalte sichergestellt werden, wenn diese nacheinander zugänglich gemacht werden. Navigation erlaubt es Nutzern, aktiv zu entscheiden und zu steuern, wann welche Inhalte präsentiert werden. In Abhängigkeit von der Anwendung können verschiedene Navigationstypen geeignet sein, um zu weiteren Inhalten zu gelangen.

Durch Interaktion kann der Nutzer dem System seinen Navigationsziel mitteilen. Interaktionselemente stellen somit Schnittstellen zwischen dem Nutzer und dem System dar, um Navigation auslösen und durchführen zu können. Die Interaktion ist jedoch nicht an die Navigation gebunden, da durch Interagieren nicht zwingend ein Fortgang in der Story verbunden sein muss, dies aber eine Forderung der Navigation ist, um Handlungen durchführen zu können ohne die Orientierung zu verlieren.

Ein weiterer Bereich dieser Dimension sind Funktionen. Allgemein lässt sich zwischen statischen und dynamischen Funktionen unterscheiden, wobei Ergebnisse statischer Funktionen infolge feststehender Eingabewerte stets identisch sind. Der Aufruf einer Funktion kann explizit oder implizit erfolgen. Implizite Aufrufe von Funktionen werden zumeist in Verbindung mit einer Interaktion durch den Nutzer ausgelöst. Der Aufruf der Funktion kann sowohl nebenläufig erfolgen und unabhängig von der eigentlichen Interaktion sein als auch bewusst die Interaktion beeinflussen und sogar das Ausführen der angestrebten Interaktion, z.B. Navigieren zu weiteren Inhalten, vom Ergebnis der Berechnung abhängig machen. Häufig wird dies eingesetzt, um Formular-Eingaben der Nutzer vor dem tatsächlichen Versenden zu prüfen. Auch ohne Interaktion können implizit Funktionen aufgerufen werden, z.B. um selbsttätig eine Aktualisierung der Anzeige zu bewirken. Dies wird unter anderem für Chat-rooms benutzt. Das explizite Aufrufen einer Funktion ist mit einer Interaktion und häufig mit Nutzereingaben verbunden. Die Suchfunktion ist ein Beispiel für diesen Funktionstyp.

2.1.6 Presentation

Die *Presentation*-Dimension weist die größte Bindung zur konkreten Entwicklung von Layout und Playout auf und kann am stärksten durch die anderen Dimensionen beeinflusst werden. Ihre Aufgabe besteht darin, die Definitionen und Restriktionen der anderen Dimensionen zu konkretisieren bzw. eine geeignete Darstellungsform zu beschreiben, die den Anforderungen der anderen Dimensionen gerecht wird. Die Presentation-Dimension definiert das Look&Feel der Ausgabe und wird neben pragmatischen durch Design Entscheidungen geprägt. Design Entscheidungen hängen nicht nur vom Typ der Anwendung ab, sondern auch vom Profil des Nutzers und kulturellen Prägungen. Infolgedessen unterscheidet sich das zu entwickelnde Ambiente einer Business-Site deutlich von einer Website für Kinder. Das Ambiente einer Website umfasst neben der Farbgebung, auch die verwendeten Formen, die Platzierung der Inhalte, die Komplexität der Navigation und den Umfang der Inhalte pro Seite.

Allgemein wird die tatsächliche Repräsentation der Inhalte durch das Zusammenwirken aller 6 Dimensionen der WIS Entwicklung bestimmt. Nicht in jedem Falle ist es sinnvoll, die Presentation-Dimension stets als letzte Dimensionen zu entwickeln, da obligatorische Anforderungen an die Präsentation bestehen können. Mitunter kann das Definieren von Beschränkungen innerhalb anderer Dimensionen dazu führen, dass die gestellten Anforderungen an die Präsentation nicht erfüllt werden können. Sollen bewusst ausgewählte Eigenschaften der Präsentation dominieren, so kann der Presentation-Dimension das höchste Gewicht zugeordnet bzw. der größte Einfluss eingeräumt werden.

Der Typ einer Anwendung beeinflusst häufig die Anforderungen hinsichtlich der Partitionierung des Gestaltungsraums, da sich nicht nur der Umfang und die Art der Inhalte stark unterscheiden kann, sondern auch deren Zugänglichkeit, z.B. durch die Wahl des Navigationstyps. Während die Themen großer Information-Sites durch Verschlagwortung oft auf sehr verschiedene Weise zugänglich sind, erlauben Edutainment-Sites zu meist nur kleinere Abweichungen im Navigationspfad innerhalb einer Lerneinheit. Dient die Anwendung der Erledigung von Aufgaben gilt es zu berücksichtigen, ob die Bearbeitungsreihenfolge von Teilaufgaben relevant ist. Kann eine Aufgabe typischerweise nicht innerhalb einer Session vollständig erledigt werden, sind visuelle Erinnerungen hilfreich, die darüber Auskunft erteilen welche Teilaufgaben noch nicht, nur partiell oder bereits vollständig erledigt wurden.

Beim Entwurf der Gestaltung ist auch das Wissen zu beachten, über welches ein Nutzer vor und während der Nutzung der Anwendung verfügt. In Abhängigkeit von der Komplexität möglicher Probleme kann der Entwurf expliziten Raum für Hilfestellungen erfordern, um den Nutzer bei der Erledigung seiner Aufgaben geeignet zu unterstützen. Hilfestellungen können nicht nur Inhalte betreffen, sondern auch die übrigen Dimensionen.

Beim Entwurf großer WIS werden Layoutaspekte oft sehr spät in den Entwicklungsprozess involviert. Partiiell entsteht das Layout sogar nahezu unabhängig von der konkreten Anwendung, indem lediglich Typ der zu entwickelnden Anwendung beschränkendes Mittel ist. Dies führt nicht selten dazu, dass das Layout einer Website und die Inhalte nicht miteinander harmonieren. Zur Flexibilisierung der Ausgabe werden insbesondere Templates eingesetzt. Templates reduzieren nur bedingt den Aufwand für Erweiterungen, Änderungen und Re-Designs, da sie auf dem Code-Level entwickelt werden und dadurch in ihrer Ausgabeform und Flexibilität beschränkt sind. Eine echte Adaption der Inhalte ist mit dem Template-Ansatz nicht möglich, da den Inhaltstypen eine konkrete Repräsentation zugeordnet wird, anstelle einer Menge von Repräsentationstypen aus denen das System den Geeigneten auswählen kann. Dieser Umstand erschwert die Equipment Adaption, da infolgedessen pro Endgerätetyp eine eigenständige Repräsentation entwickelt, konfiguriert und gewartet werden muss oder andernfalls Einschränkungen hinsichtlich der Adaptivität bzw. Qualität der Darstellung hingenommen werden müssen.

2.2 Zusammenwirken der Dimensionen

Prinzipiell können die oben eingeführten Dimensionen separat entwickelt und spezifiziert werden, allerdings können sie sich auch gegenseitig beeinflussen. Da der Einfluss der ein-

zelen Dimensionen vom jeweiligen Typ der Anwendung abhängt, ist es notwendig, die Spezifikationsreihenfolge der Dimensionen für einen konkreten Entwicklungsprozess festzulegen. Durch die Gewichtung der Dimensionen können die Präferenzen der Anwendung besser berücksichtigt werden. Folgend soll anhand von Beispielen und Analogien gezeigt werden, dass unterschiedliche Gewichtungen nötig sind, um die Ausgaben flexibel an die Bedürfnisse der Nutzer anzupassen.

2.2.1 Website Entwicklung - state of the art

An der klassischen Entwicklung von Websites sind primär die Dimensionen Content, Functionality und Presentation beteiligt, wie in Abbildung 2.3 hervorgehoben dargestellt. Sollen vordefinierte Templates eingesetzt werden, so kann die Spezifikation der Presentation Dimension als abgeschlossen betrachtet werden. In diesem Fall sind somit lediglich die Inhalte und die Funktionalität gemäß den Vorgaben des Templates zu integrieren. Für die Integration ist es essentiell, dass alle Inhaltstypen und die Funktionalität in allen Situationen mit den Möglichkeiten des Templates abgeglichen werden und der Entwickler über mögliche Probleme bei der Integration informiert wird. Andernfalls können Darstellungsprobleme nicht ausgeschlossen werden. Da der Aufwand der Überprüfung mit zunehmendem Inhalts- und Funktionsumfang exponentiell ansteigt, werden oft nur typische Szenarios getestet und etwaige andere Darstellungsprobleme erst im Falle des Entdeckens im Live-Betrieb behoben. Ferner gilt es zu beachten, dass nicht alle Darstellungsprobleme automatisiert erkannt werden können, da die Eignung auch durch Design-Entscheidungen beeinflusst wird. Sollten Inhaltsobjekte nicht zur Darstellung geeignet sein oder eine geeignete Repräsentation nicht bekannt sein, muss ein anderes Template gefunden werden, um die Darstellung nicht zu gefährden, da Alternativdarstellung beim klassischen Template-Konzept nicht existieren. Darüber hinaus besteht die Möglichkeit, das bestehende Template zu ändern oder zu erweitern. Der Template-Ansatz ist vor allem für Websites kleineren Umfangs geeignet, da sich in diesem Fall der Test bzgl. bestehender Integrationsprobleme notfalls auch manuell realisieren lässt. Alternativ kann mit der Spezifikation der Inhalte und der Funktionalität begonnen und erst danach die Präsentation abgeleitet werden.

| | | | |
|-----------|---------|----------------------|---------------------|
| Intention | Story | Content | Presentation |
| | Context | Functionality | |

Abbildung 2.3: Dimensionen der Website Entwicklung - state of the art

Die verbleibenden Entwicklungsdimensionen spielen bei der derzeitigen Website Entwicklung eine untergeordnete Rolle und werden oft sogar vollständig vernachlässigt. Unter anderem ist die Intention Dimension kein explizit modellierter Teil derzeitiger Entwicklungen. Oft nehmen die an der Entwicklung beteiligten Personen bis zur Realisierung der Anwendung lediglich an, dass sie die selbe Vorstellung von den Zielen haben. Aufgrund fehlender, überprüfbarer allgemeiner Konzepte wird jedoch nicht geprüft, ob auch real eine übereinstimmende Hauptentwicklungsrichtung besteht. Auch die Story

und Context Dimension werden nur selten berücksichtigt. Das Ignorieren des Kontextes kann zu Ausgaben führen, die ungeeignet für das Equipment und die Zielgruppe sind. Ohne die Spezifikation einer Story ist es nur möglich, die Inhalte zu definieren, jedoch nicht in welcher Weise die Inhalte semantisch in Verbindung stehen.

2.2.2 Website Entwicklung - Architektur

Die Notwendigkeit und Möglichkeit einer alternativen Herangehensweise zeigt sich am deutlichsten unter Zuhilfenahme der Architektur als Analogie. Folgend soll die abweichende Bedeutung der Dimensionen am Beispiel der Gebäudeplanung gezeigt werden. Entwicklungen im Bereich der Architektur sind stark durch den Kontext beeinflusst, da z.B. Gebäude nicht ohne Berücksichtigung ihrer Umgebung geplant werden. Oft ist es ein wichtiges Ziel des Bauherren, dass sich das Bauwerk gut in die Umgebung bzw. in die bestehende Architektur und Gestaltung einfügt. Auch innerhalb eines Gebäudes ist der Kontext von Bedeutung. Beispielsweise ist die Orientierung innerhalb des Gebäudes nur dann befriedigend, wenn sie den Erwartungen und Anforderungen der Hauptnutzer entspricht. Auch bei der Entwicklung einer Webanwendung kann die Forderung einer dominierenden Context Dimension auftreten, um z.B. Kompromisse bei den Zugriffsrechten oder einzelner QoS Parameter nicht hinnehmen zu müssen.

Bei einer Entwicklung nach dem Architekturansatz besitzt die Intention Dimension häufig oberste Priorität. Sie repräsentiert den Ausgangspunkt der Entwicklung und bestimmt ferner die Entwicklungsrichtung. Nachfolgend entwickelte Dimensionen, wie die Context Dimension, können von den Definitionen abweichen, sollten jedoch nicht die gesamten intentionalen Aspekte ignorieren. Lediglich wenigen, ausgewählten Parametern sollte es erlaubt sein, von den Erwartungen der Nutzer abzuweichen, da andernfalls eine Beeinträchtigung der Wahrnehmung und des Verständnisses zu erwarten ist. In Einzelfällen kann es sinnvoll sein, die Wahrnehmung oder Erwartungen gezielt zu stören, um die Aufmerksamkeit des Nutzers zu steigern. Die dritte Hauptdimension dieses Entwicklungsansatzes ist die Funktionalität. Sie wird in der Regel nach dem Definieren der Intention Dimension spezifiziert, mitunter aber vor der Context Dimension. Abbildung 2.4 hebt die Hauptdimensionen des Architekturansatzes hervor.

| | | | |
|------------------|----------------|----------------------|--------------|
| Intention | Story | Content | Presentation |
| | Context | Functionality | |

Abbildung 2.4: Dimensionen der Website Entwicklung - Architektur

Trotz der Bedeutung von Intention, Context und Functionality Dimension können auch die untergeordneten Dimensionen das Gesamtergebnis beeinflussen. Der Spezifikation der drei Hauptdimensionen folgt bei diesem Entwicklungstyp die Spezifikation von Content und Story. Die Story definiert mögliche Pfade, die den Nutzern durch die Funktionalität zur Verfügung gestellt werden, unter Beachtung der definierten Zugriffsbeschränkungen. In diesem Fall kann die Story direkten Einfluss auf die Usability ausüben, da z.B. die Interaktionsform bereits bekannt ist. Die Content Dimension kann

Platzierungsanforderungen aufweisen und muss die Vereinbarkeit ihrer Definitionen mit Entscheidungen anderer Dimensionen abgleichen. Beispielsweise hängen die Definitionen der Content Dimension von der verfügbaren Funktionalität und dem Context ab, da z.B. nicht jeder Inhalt in jeder Form über einen Kommunikationskanal geringer Bandbreite transportiert werden kann, ohne QoS Forderungen zu verletzen oder Erwartungen der Nutzer zu enttäuschen. Erst in einem dritten Schritt wird beim Architekturansatz die Presentation Dimension spezifiziert und wird unter Berücksichtigung der bis dahin vorhandenen Vorgaben der anderen fünf Dimensionen entwickelt. Nichtsdestotrotz ist diese Dimension bedeutsam, da sie die tatsächliche visuelle Repräsentation bestimmt und zugleich Fehler oder zu enge Vorgaben oberer Dimensionen aufzeigt.

2.2.3 Website Entwicklung - Kunst

Eine dritte Variante der Entwicklung von WIS lässt sich von der Kunst ableiten, insbesondere mit Blick auf die klassischen Dramen und das Arrangieren von Museums-Ausstellungen oder Theateraufführungen. Haupttreiber sind dabei die Dimensionen Content, Story und Presentation, wodurch dieser Entwicklungsansatz zum Gegenstück des Architekturansatzes wird. In Abbildung 2.5 werden die Hauptdimensionen des Kunstansatzes hervorgehoben.

| | | | |
|-----------|--------------|----------------|---------------------|
| Intention | Story | Content | Presentation |
| | Context | Functionality | |

Abbildung 2.5: Dimensionen der Website Entwicklung - Kunst

Der Content einer Theaterveranstaltung ist in der Regel implizit durch ein Manuskript oder Buch gegeben. Beim Arrangement klassischer Dramen existiert sogar der vollständigen Handlungsverlauf, da dieser aus dem Buch hervorgeht und innerhalb der Story Dimension als Storyboard formuliert wird. Die Präsentation der Inhalte wird folglich stark durch die Content und Story Dimension beeinflusst, jedoch nicht ausschließlich. Gerade bei klassischen Dramen besteht oft die Anforderung, die Aufführung und Präsentation nicht strikt gemäß des Ursprungsdokuments zu realisieren, um die Attraktivität für die Zuhörerschaft zu steigern. Die Intendanten versuchen daher nicht selten klassische Themen mit aktuellen Ereignissen zu verknüpfen, infolgedessen die Präsentation der Inhalte vom Original abweicht und die Sicht des Intendanten reflektiert wird. In diesem Fall wird die Präsentation durch die sekundären Dimensionen des Kunstansatzes beeinflusst, die Intention, Context und Functionality Dimension. Möchte der Intendant beispielsweise klassische und moderne Aspekte verbinden, wird das notwendigerweise die Art der Darstellung der Inhalte als auch deren Komposition beeinflussen. In gleicher Weise kann sich die Präsentation ändern, wenn Kontextbedingungen vom implizit angenommenen Standardkontext abweichen, indem z.B. im Freien statt innerhalb des Theaters aufgeführt wird. In solch einem Fall wird sich auch die verfügbare Funktionalität unterscheiden, wodurch gegebenenfalls weitere Einschränkungen die Präsentation betreffend erfolgen müssen.

Die beschriebene Gewichtung der Dimensionen besteht auch für das Arrangement von Ausstellungsgegenständen in Museen, da die Presentation, Story und Content Dimension auch dort die Hauptentwicklungsrichtung vorgeben, während die anderen 3 Dimensionen eine sekundäre Rolle einnehmen und Variationen bzw. Individualisierungen erlauben. Ferner eignet sich die Gewichtung nicht nur für die klassische Präsentation, sondern kann auch für Online Repräsentationen von Museen verwendet werden. Die Übertragbarkeit ergibt sich infolge ähnlicher Anforderungen, da Ausstellungsgegenstände beispielsweise innerhalb eines virtuellen Rundgangs erlebbar gemacht werden können. Das Online Pendant bietet gegenüber der klassischen Ausstellung den Vorteil, dass sich die Darbietung der Inhalte, z.B. visuell oder auditiv, gezielt an die Anforderungen und Möglichkeiten des Nutzers anpassen lässt.

2.2.4 Website Entwicklung - Zukunft (state of science)

Die Entwicklung von Webinformationssystemen lässt sich gemäß [67] in mehrere Stufen bzw. Detaillierungsgrade unterteilen. Das Abstraktionsschichten-Modell (Abstraction Layer Model - ALM) [65] unterscheidet dazu Abstraktionssschichten explizit durch eine schrittweise Entwicklung der Ebenen und Verfeinerung der Spezifikation.

Auch eine schrittweise Entwicklung des Layout ist möglich, indem die Dimensionen den Schichten des ALM zugeordnet werden. Die Spezifikationsreihenfolge der Dimensionen ist dabei nicht zwingend global gültig, sondern kann für jede Abstraktionsebene separat festgelegt werden. Ferner werden nur selten alle Dimensionen auf allen Abstraktionsschichten spezifiziert. Deutlich wird dies bei näherer Betrachtung des folgenden Beispiels, dargestellt in Abbildung 2.6.

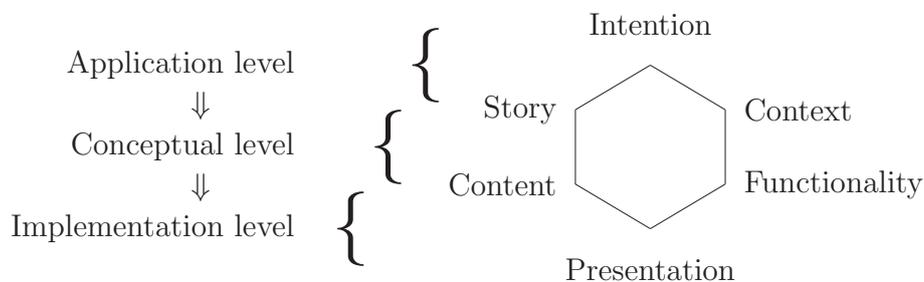


Abbildung 2.6: Dimensionen Hierarchie

Die in Abbildung 2.6 gezeigte Zuordnung der Dimensionen beschreibt einen Layout-Entwicklungsprozess, der alle Dimensionen umfasst, jedoch nur wenige Dimension auf mehreren Abstraktionsschichten spezifiziert. Der Spezifikationsprozess beginnt in diesem Fall mit der Intention auf dem Anwendungslevel, um die Motivation für die Entwicklung der Anwendung zu ermitteln als auch Ziele und Absichten. Erst im Anschluss werden Aspekte der Story und Context Dimension spezifiziert unter Berücksichtigung der

bisherigen Festlegungen. Aufgrund ihrer großen Bedeutung für die Anwendung, werden Story und Context auch auf der konzeptionellen Ebene fortentwickelt. Durch die Vielfalt des Equipments kann insbesondere die Context Dimension die spätere Repräsentation der Inhalte stark beeinflussen. Die Story Spezifikation beeinflusst durch den entstehenden Handlungsverlauf die Suche nach einer geeigneten Partitionierung des Darstellungsraums. Je nach Gewichtung der Dimensionen kann die Story die Context Dimension beeinflussen oder berücksichtigt deren Vorgaben. Auf dem konzeptionellen Level werden ferner die nötige Funktionalität und die Eigenschaften des sowie die Anforderungen an den Content definiert. Die Konzepte von Content und Funktionalität werden anschließend auf die Implementationsebene übertragen und präzisiert. Im letzten Schritt dieses Entwicklungsansatzes wird die Presentation Dimension spezifiziert, um das Layout im Detail zu bestimmen, unter Berücksichtigung der spezifizierten Anwendungsmerkmale. Abbildung 2.7 veranschaulicht die gewählte Gewichtung der Dimensionen, die sowohl Flexibilität als auch Adaptivität ermöglicht. Dabei werden Intention, Story und Context explizit entwickelt und deren Festlegungen bei der Spezifikation der übrigen Dimensionen berücksichtigt.

| | | | |
|------------------|----------------|---------------|--------------|
| Intention | Story | Content | Presentation |
| | Context | Functionality | |

Abbildung 2.7: Dimensionen der Website Entwicklung - Zukunftsszenario

2.3 Flexibilisierung des Entwicklungsprozesses

Wie im vorangegangenen Abschnitt gezeigt, lässt sich die Entwicklung des Layout von Webinformationssystemen durch das Einführen von Dimensionen flexibilisieren, indem deren Bedeutung an die Forderungen der Anwendung gebunden werden. Die oben eingeführten Dimensionen eignen sich jedoch nicht nur für die Entwicklung des Layout, sondern für den gesamten Entwicklungsprozess von Webinformationssystemen. In [53] wurde dazu ein allgemeiner Entwicklungsansatz für Webinformationssysteme eingeführt, der mehrere Komponenten der Entwicklung unterscheidet. Notwendig ist diese Ergänzung, da frühe Spezifikationsentscheidungen innerhalb einzelner Komponenten der Anwendung den Gesamtentwicklungsprozess maßgeblich beeinflussen können. Darüber hinaus wird nicht nur ein höheres Maß an Flexibilität innerhalb einer Komponente ermöglicht, sondern durch die entwicklungsphasenabhängige Gewichtung der Komponenten auch die Möglichkeit geschaffen, den Gesamtentwicklungsprozess auf allen Abstraktionsleveln aktiv zu steuern. Die aktive Steuerung wird durch Variation der Spezifikationsreihenfolge der Komponenten erreicht, in Abhängigkeit von den Zielen der Anwendung und damit verbundenen Restriktionen. Die globale Reihenfolge ist Schichtübergreifend möglich, sollte jedoch innerhalb einer Komponente immer unter Erhalt der Entwicklungsrichtung erfolgen, generalisierend oder spezialisierend.

Exemplarisch soll der Spezifikation des Frontend die Spezifikation des Backend gegenübergestellt werden. Einhergehend ist die Reihenfolge und Präferenz der Entwicklung

der Teilbereiche beider Komponenten zu bestimmen. Da weiterhin eine schrittweise Systementwicklung erfolgen soll, ist die Anzahl der zu entwickelnden Abstraktionsschichten für jede Komponente festzulegen. Je nach Komplexität der Anwendung kann der volle Umfang des Abstraktionsschichtenmodells erforderlich sein oder aber nur ein Teil. Liegt beispielhaft eine 3-Schichten-Architektur zugrunde, so ergibt sich ein Grundgerüst der Entwicklung, wie in Abbildung 2.8 dargestellt.

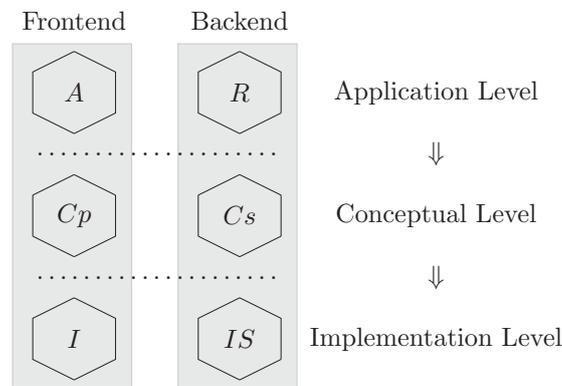


Abbildung 2.8: Grundgerüst der Entwicklung

Für die Spezifikation des Frontend ist zunächst die Anwendungsdomäne A zu definieren, die allgemeine Layout-Anforderungen wie das Ambiente und den Typ der Anwendung umfasst. Ausgehend von dieser ist das Präsentationskonzept Cp konzeptionell zu entwickeln, um eine hinreichende Flexibilität der Ausgabe zu ermöglichen. Anschließend wird das Präsentationskonzept in eine konkrete Implementierungsform I überführt und erreicht innerhalb dieser Schicht so spät wie möglich das eigentliche Code-Level.

- A - Application domain
- Cp - Concept of presentation
- I - Implementation of presentation system

Aus der Perspektive des Backend gilt es zunächst die Anforderungen R unter Beachtung der Anwendungsdomänenspezifikation zu beschreiben, um davon ausgehend ebenfalls das konzeptionelle Modell Cs entwickeln zu können. Die Implementationsschicht aus der Sicht des Backend resultiert durch weitere schrittweise Verfeinerung im eigentlichen Informationssystem IS .

- R - Requirements prescriptions
- Cs - Conceptual specification
- IS - Information System

Ausgehend von der 3-Schichten-Architektur werden in Abbildung 2.9 zwei Entwicklungsansätze dargestellt, die sich hinsichtlich der Komponenten-übergreifenden Entwicklungsreihenfolge der Abstraktionsschichten unterscheiden. Entwickler inhaltlich umfangreicher Anwendungen bevorzugen oft die sogenannte *System Perspektive*, dargestellt durch die rechte Figur in Abbildung 2.9. Auf diese Weise entwickelte Anwendungen werden stark durch die Festlegungen des Backends bestimmt, während die Frontend-Entwicklung diesen Bestimmungen folgt. Eine Ausnahme stellt die Entwicklung der Anwendungsdomäne dar, die üblicherweise vor den Backend Anforderungen spezifiziert wird. Zu den Vorteilen dieser Herangehensweise zählt die Stabilität und Portabilität des Systems aus dem Blickwinkel der Datenhaltung.

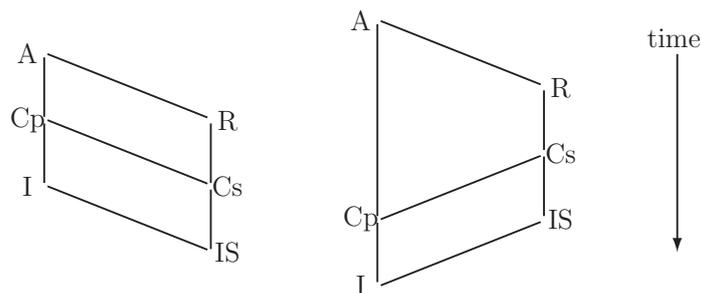


Abbildung 2.9: Perspektiven der Entwicklung - User (links), System (rechts)

Einen konträren Ansatz stellt die Entwicklung in *User Perspektive* dar, deren Fokus primär auf der Frontend-Entwicklung liegt. Diese Entwicklungsvariante, dargestellt durch die linke Figur in Abbildung 2.9, wird häufig für Anwendung geringen, inhaltlichen Umfangs verwendet. Vorteil dieses Entwicklungsprozesses ist die geringe Abhängigkeit der Entwicklung der Repräsentation von dem Entwicklungsstand der Datenhaltung. Dieser Umstand kann allerdings dessen (Weiter-)Entwicklung nicht nur beeinflussen, sondern auch behindern. Aus dem Blickwinkel des Frontend entwickelte Anwendungen können adaptiver hinsichtlich der Präsentation der Daten sein, da nicht die optimale Datenhaltung im Vordergrund steht, sondern die flexible Ausgabe. Dies bedeutet jedoch mitunter Redundanzen, eine geringere Migrationsfähigkeit und Integrierbarkeit.

2.4 Website-Typen

Die Entwicklung jeder Web-Anwendung wird von einer Reihe impliziter und expliziter Entscheidungen begleitet, welche die Stellung und Beziehung von Einflussfaktoren festlegen. Das frühzeitige Treffen dieser Entscheidungen ist wünschenswert und kann den Typ einer Website charakterisieren. Eine übermäßig detaillierte Spezifikation kann allerdings die spätere Wiederverwendung eines solchen Website-Typs behindern, ein zu geringer Spezifikationsumfang hingegen fälschlich eine universelle Eignung suggerieren. Zur Vermeidung dieses Konflikts kann die Zugehörigkeit zu einem Website-Typen auf einem abstrakteren Niveau entschieden werden. Zugleich erlaubt dies eine Top-Down

Entwicklung anstelle der Verwendung des Bottom-Up Ansatzes. Ein mögliches Klassifikationsmerkmal ist der *Einsatzbereich der Anwendung*, dessen Eignung aus der Analyse verschiedener Web-Projekte hervorging und in [65] zur Unterscheidung folgender Website-Typen führte:

- e-Business sites
- Edutainment sites
- Information sites
- Collaboration sites
- Identity sites
- Infotainment sites

Unter Berücksichtigung dieser Klassifikation werden folgend ausgewählten Website-Typen allgemeine Einflussfaktoren zugeordnet, die eine erste Schärfung und Abgrenzung erlauben. Jede nutzbare Ausprägung eines Website-Typs bedarf einer weiteren Verfeinerung, die spezifische Entwicklungsbedingungen und -vorgehensweisen beschreibt und damit die tatsächliche Verwendbarkeit weiter einschränkt. Die hier betrachteten Website-Typen werden damit maßgeblich durch den Einfluss beteiligter Faktoren, deren Beziehungen und gewähltes Abstraktionsniveau bestimmt.

Für die Entwicklung von Web-Anwendungen bedeutet diese Vorgehensweise zunächst, allgemeine Ziele zu analysieren, mit Hilfe derer relevante Einflussfaktoren abgeleitet werden. Durch eine nähere Betrachtung der Ziele wird danach das jeweilige Gewicht der beteiligten Faktoren bestimmt. Diese Gewichtung wirkt sich auch auf die erforderliche Spezifikationsreihenfolge der Entwicklungsdimensionen aus und beeinflusst damit maßgeblich das Ergebnis der Anwendungsentwicklung. Anschließend ist ein Website-Typ auszuwählen, der am besten zu den ermittelten Anforderungen der Entwicklung passt. Sind mehrere Typen gleichermaßen gut geeignet, sind die Ziele der Entwicklung erneut und verfeinernd zu betrachten. Ist keine Eignung gegeben, ist die bestehende Spezifikation meist zu detailliert. Eine Erweiterung der Website-Typen-Klassifikation sollte nur erfolgen, wenn die Existenz des neuen Typen zwingend ist bzw. die Entwicklung nicht durch eine Variante bestehender Typen realisierbar ist. Andernfalls ist rasch eine Anhäufung sehr ähnlicher, spezialisierter Website-Typen zu erwarten, deren Abgrenzbarkeit gering ist. In der Folge würde auch für zukünftige Entwicklungen die Auswahl des geeignetsten Website-Typs erschwert.

Die folgend betrachteten Website-Typen werden hinsichtlich der Bedeutung der Entwicklungsdimensionen untersucht und nötige Entwicklungsschritte diskutiert. Dabei wird auch auf Unterschiede eingegangen, die zwischen der derzeitigen Entwicklung eines Website-Typs und der gewünschten Vorgehensweise bestehen. Darüber hinaus zeigt sich, dass Erweiterungen der derzeitigen Klassifikation gleichermaßen möglich sind wie die Kombination bereits bestehender Typen. Die hierzu verwendete Notation für die Darstellung des Einflusses der Entwicklungsdimensionen (Abb. 2.10) unterscheidet zwischen

Dimensionen, die verpflichtend zu spezifizieren sind und solchen, bei denen dies optional möglich ist. Darüber hinaus wird unterschieden, ob dies explizit oder implizit geschieht.

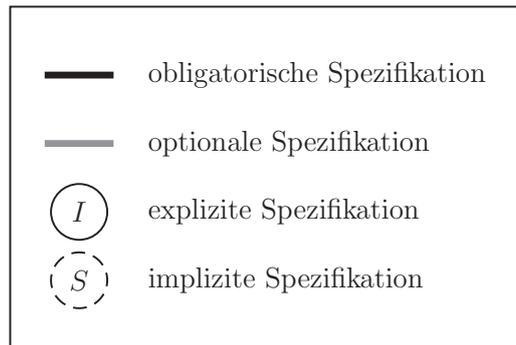


Abbildung 2.10: Entwicklungsdimensionen - Notation

Der optional-implizite Fall ist zu vermeiden, stellt aber eine gültige Option dar, wenn zum Zeitpunkt der Spezifikation die implizite Beteiligung nur partiell vorausgesetzt werden kann. Eintreten kann dieser Fall, wenn die implizite Spezifikation einer Story zwar für komplexe, nicht aber für einfache Content-Typen als zwingend angenommen werden kann, der Bedarf komplexer Content-Typen jedoch noch nicht entschieden ist.

Der Einfluss einer Dimension auf eine Andere wird durch Pfeile zwischen den Dimensionen dargestellt. Hierbei unterscheidet die Notation in gleicher Weise wie bei den Entwicklungsdimensionen selbst zwischen obligatorischem oder optionalem sowie explizitem oder implizitem Einfluss.

2.4.1 Infotainment sites

Zu den Infotainment sites zählen Web-Anwendungen, die von den Nutzern nicht nur als informativ (*Information*) wahrgenommen werden, sondern sich diesen auch unterhaltsam (*Entertainment*) präsentieren. Abbildung 2.11 zeigt in drei Figuren jene Dimensionen, die die Entwicklung von Infotainment sites implizit oder sekundär (gestrichelte Linie), optional (graue Linie) und direkt (schwarze Linie) beeinflussen.

An der Entwicklung derzeitiger Infotainment sites werden die Dimensionen Content, Funktionalität und Präsentation explizit beteiligt, dargestellt durch die linke Figur in Abbildung 2.11. Aspekte der Dimension Intention sind lediglich implizit von Bedeutung, indem Ziele und Absichten höchstens informal und ohne direkten Einfluss auf die Gesamtentwicklung formuliert werden. Auch die Wahl des Ambiente erfolgt überwiegend auf Basis des subjektiven Geschmacks der Auftraggeber, Designer oder Entwickler und stimmt nicht zwingend mit den Anforderungen der Anwendung oder den Wünschen der Zielgruppe überein. Wird die Intention nicht explizit berücksichtigt, ist oft eine langsame Zunahme von Differenzen die Folge, die nur unzureichend und in der Regel zu spät erkannt wird. Im Falle nicht-kommerzieller Infotainment sites geringen Umfangs ist der Aufwand für eine detaillierte Zielgruppenanalyse in der Regel zu hoch, weshalb der Erfolg der Anwendung stark von den Fähigkeiten und der Erfahrung der Betreiber

abhängt. Wiederverwendbare Spezifikationsvorlagen (Pattern) können jedoch auch dann die Entwicklung erleichtern und vor typischen Fehlern bewahren.

Die Story Dimension wird oft nur implizit an der Entwicklung der Anwendung beteiligt, indem sich die Struktur der Inhalte an den Festlegungen der Intention orientiert. Eine typische Folge dieser Herangehensweise sind schwach verbundene Graphen oder sogar Bäume, die direkt als Navigationsschema dienen. Dies erleichtert die Pflege und Wartung, vermeidet aber auch zusätzlichen Konfigurationsaufwand zum Zeitpunkt der Entwicklung. Ferner ist die direkte Verfügbarkeit und Erreichbarkeit neu hinzugefügter Inhalte in jedem Falle gegeben. Überdies lassen sich prinzipiell beliebig große Informationsmengen auf diese Weise verwalten. Von Nachteil ist, dass ein so erzeugtes Navigationsschema Individualisierungsbestrebungen nur sehr eingeschränkt zulässt und folglich nur bedingt an die Bedürfnisse eines konkreten Nutzers angepasst werden kann. Durch Kategorisierung werden oft umfangreiche Themenbereiche separiert, um dem Nutzer den Einstieg oder die gezielte Suche zu erleichtern. Mit der Vielfalt der Inhalte steigt allerdings auch die Komplexität der Benennung und thematischen Zuordnung, infolgedessen die zielgerichtete Navigation für den Nutzer erschwert oder gar unmöglich wird. Die Notwendigkeit der Story-Spezifikation hängt somit maßgeblich vom der Größe der Anwendung ab und sollte daher optional sein, wie durch die mittlere Figur der Abbildung 2.11 dargestellt.

Die Entwicklung von Infotainment sites orientiert sich in der Regel an einer konkreten Zielgruppe und Situation. Durch die starke Fokussierung auf eine Zielgruppe ist eine Unterscheidung von Rollen nicht zwingend erforderlich. Auch die Unterscheidung von Benutzerrechten kann unnötig sein und ohne realen Mehrwert die Unterhaltungswirkung der Anwendung sogar beeinträchtigen. Die Gesamtheit intentionaler Beschränkungen führt häufig dazu, dass die Context Dimension nur implizit definiert wird. Die fehlende explizite Spezifikation während der Entwicklungsphase kann die spätere Adaptionsfähigkeit beeinträchtigen, hinsichtlich der Darstellung der Inhalte durch verschiedenstes Equipment.

Die Spezifikation der verbleibenden 3 Dimensionen erfolgt explizit. Durch die Struktur und Art der Inhalte wird typischerweise deren Präsentation maßgeblich bestimmt. Daneben hat die Intention durch die implizite Definition des Ambiente direkten Einfluss auf die Präsentation von Infotainment sites. Die mögliche Funktionalität, die dem Nutzer bereitgestellt werden soll, hängt direkt von der Darstellung der Inhalte ab. Dies schließt die Intention einer konkreten Darstellung ein, die eine ausgewählte Funktionalität erfordern kann. Ferner wird die Funktionalität durch die Context Dimension beeinflusst, da unter anderem die Fähigkeiten des Equipments beschränkend wirken.

Abbildung 2.11 macht deutlich, dass Infotainment sites nicht starr definiert werden, sondern Variationen zulassen, um bei der Entwicklung individuelle Anforderungen berücksichtigen zu können. Bei einer Vorgehensweise entsprechend der rechten Figur in Abbildung 2.11, ist aus der Perspektive des Schema-Entwurfs zunächst das Datenbank-Schema zu entwickeln. Die Struktur des Schemas berücksichtigt dabei in Abhängigkeit von den Bedürfnissen der konkreten Anwendung implizit oder explizit die Möglichkeiten der Story Spezifikation.

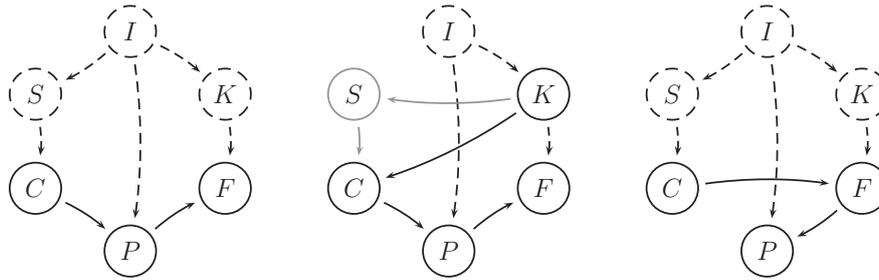


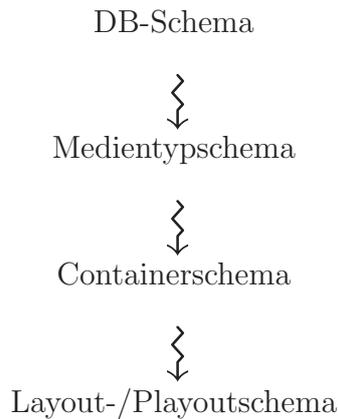
Abbildung 2.11: Dimensionen bei Infotainment sites

Nach dem DB-Schema lässt sich ein Medientypschemata entwickeln, welches den innerhalb des Interfaces benötigten Content umfasst und eine Sicht über dem DB-Schema darstellt. Zur Vermeidung einer zu starken Spezialisierung, werden die Anforderungen an den Content nicht für jedes einzelne Endgerät beschrieben, sondern abstrakt für einen Medientyp. Das Medientypschemata berücksichtigt nicht nur Restriktionen hinsichtlich der Darstellbarkeit, sondern auch allgemeine Bedürfnisse der Nutzer, die mit der Nutzung eines konkreten Mediums in Verbindung stehen. Daher ist auch die Modellierung der Navigationsstruktur Teil des Medientypschematas. Darüber hinaus besitzt ein Medium oft Beschränkungen hinsichtlich verwendbarer Funktionen und Kommunikationsformen, weshalb durch das Medientypschemata eine Verfeinerung der Szenen des Storyboards erfolgen muss und zulässige Dialogtypen durch Sichten zu benennen sind.

Das Containerschemata dient der konkreten Zusammenstellung und Bereitstellung von Content mittels Sichten, einschließlich der Funktionalität und unter Beachtung des jeweils aktiven Dialogtyps. Der Dialogtyp muss initial festgelegt werden und kann sich im Anschluss aus der Art der Interaktion des Nutzers mit dem System ergeben. Container erlauben die Be- und Entladung mit Contentobjekten. Ist ein Container beladen, kann er an das Clientsystem gesandt werden. Eine möglichst effiziente Kommunikation lässt sich erreichen, wenn der zur Entladung eines Containers bis hin zur Präsentation der Inhalte nötige clientseitige Aufwand die Fähigkeiten und Beschränkungen des Clientsystems berücksichtigt, z.B. die Leistungsfähigkeit der CPU oder bei mobilen Endgeräten die Kapazität des Akkus. Einer möglichst effizienten Kommunikation können strikte Forderungen der Nutzer und Provider aber auch der Anwendung entgegenwirken, wie z.B. die verschlüsselte Kommunikation oder die Forderung nach maximaler Performance. Die Art und der Aufbau der Contentobjekte innerhalb der Container hängt somit auch von den individuell ausgehandelten Kommunikationsbedingungen ab.

Neben der Zusammenstellung der eigentlichen Inhalte gilt es die Darstellung der Inhalte als auch den Rahmen für deren Darstellung zu spezifizieren. Hierfür ist ein Layout-/ Playoutschemata zu entwickeln, welches Festlegungen des Containerschematas als auch individuelle Präferenzen der Nutzer beachtet. Das Layoutschemata dient der Beschreibung eines Ambiente im Speziellen und umfasst infolgedessen die Darstellung der Inhalte, des Rahmens und die Kennzeichnung und Repräsentation von Funktionalität. Das Playoutschemata basiert auf der Story Spezifikation und den Aufgaben der Nutzer und bestimmt die Darstellung im Allgemeinen. Durch das Playoutschemata lässt sich festlegen, welche

Layoutelemente situationsabhängig einzusetzen sind, unter Beachtung des gewählten Styles und der Historie innerhalb der Story.



2.4.2 eBusiness sites

Im Bereich des eBusiness tritt insbesondere die Entwicklung der Story-Dimension in den Vordergrund, unter impliziter Berücksichtigung der Intention. Innerhalb der Story Dimension sind typische Business Szenario zu ermitteln, welche die vorhandenen Geschäftsprozesse abbilden. Für die spätere Layout Entwicklung sind dabei neben den Ausgaben jene Prozesse von Bedeutung, die Eingaben als steuernde Parameter erwarten oder aber eine visuelle Rückkopplung erfordern, z.B. die Überwachung des Fortschritts einer Berechnung. Szenen können durch Teilszenen und Dialogschritte konkretisiert werden und helfen Informationseinheiten voneinander abzugrenzen, die eigenständig wahrgenommen werden sollen. Ein solches Splitting beeinflusst frühzeitig die Möglichkeiten der späteren Gestaltung und verhindert, dass Nutzern zu viele Informationen gleichzeitig präsentiert werden (Content overflow), unter Berücksichtigung des gewählten Präsentationsmediums sowie der Fähigkeiten der Nutzer. Die Notwendigkeit einer umfangreichen Spezifikation der Context Dimension hängt bei eBusiness sites stark von deren Zielen ab. Beispielsweise können B2C-Shops durch die zielgerichtete und vollständige Adaption der Inhalte an die Bedürfnisse einzelner Nutzer unter Umständen nur geringfügig profitieren. Eine komplexe Unterscheidung der Nutzer ist daher nicht immer erforderlich, einschließlich des Bedarfs unterschiedlicher Rechte. Ist durch die Intention ferner das Ausgabemedium klar definiert und soll dieses strategisch nur bedingt variabel sein, so können bei der Erhebung und Sicherung der Daten Abstriche hinsichtlich der Adaptionfähigkeit der Darstellung der Inhalte hingenommen werden. Aufbauend auf den Definitionen der Story folgt die Spezifikation von Content und Funktionalität. Welche der beiden Dimensionen die andere dominiert, hängt maßgeblich von ihrem Stellenwert innerhalb der Anwendung ab. Die Reihenfolge, in der sie an der Entwicklung teilhaben, kann das Resultat des Entwicklungsprozesses stark beeinflussen, da Funktionen spezielle Content-Typen voraussetzen können und umgekehrt. Die Spezifikation der Presentation Dimension erfolgt bei eBusiness sites überwiegend im Anschluss

an die Festlegung der Geschäftsprozesse und orientiert sich an deren Vorgaben. Da diese Dimension auch durch die Definitionen der Intention getrieben wird, kann sie dennoch frühzeitig in die Entwicklung der Anwendung eingreifen.

Abbildung 2.12 zeigt zwei typische Dimensions-Hierarchien von eBusiness websites.

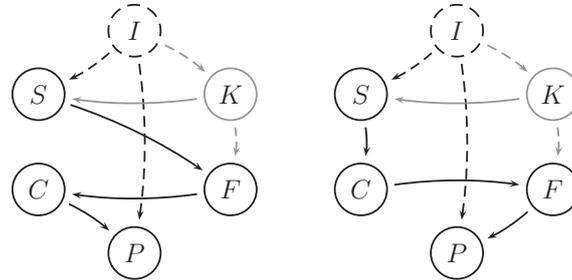


Abbildung 2.12: Dimensionen bei eBusiness sites

2.4.3 Identity sites

Identity sites dienen der Darstellung und Vorstellung der eigenen Person und weisen nicht selten einen nur geringen, inhaltlichen Umfang auf. Die Story Dimension tritt bei diesem Website-Typ lediglich indirekt in Erscheinung. Die Struktur der Stories von Identity sites ist zumeist sehr einfach, geradlinig und kaum verzweigend. Die Intention eines solchen Auftritts, aber auch die Kontextbedingungen sind implizit gegeben. Haupttreiber sind somit die Dimensionen Präsentation, Content und Funktionalität. Überwiegend werden bei der Entwicklung von Identity sites Layout-Vorlagen eingesetzt, sog. Templates, die als Grundgerüst auf dem Implementationslevel für mehrere, oft sogar alle Seiten des Webauftritts fungieren und mit Inhalten und geeigneter Funktionalität angereichert werden können. Aus pragmatischen Gründen muss sich zudem die Spezifikation der Funktionalität meist der Spezifikation der Inhalte unterordnen. Abbildung 2.13 stellt den Entwicklungsprozess für zwei typische Identity sites dar.

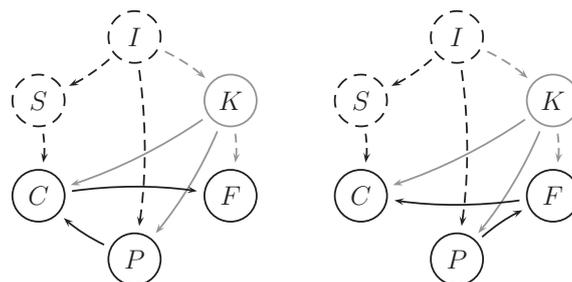


Abbildung 2.13: Dimensionen bei Identity sites

Vor allem bei geringem, inhaltlichen Umfang und seltenem Änderungsbedarf von Identity sites sollte auch der strukturelle Pflegeaufwand gering sein. Oftmals resultiert dies

im bewussten Verbleib auf dem Implementationslevel, da dieses eine geringe Komplexität bietet und infolgedessen schnelles Ändern erlaubt. Wenngleich eine solche Entscheidung den Aufwand von komplexen Änderungen erhöht, verursacht der Verzicht auf ein höheres Abstraktionsniveau mitunter dennoch einen geringeren Gesamtaufwand, wenn den Entwicklern auf dem höheren Abstraktionsniveau keine Hilfestellung bei der Spezifikation und Entwicklung von Abbildungsregeln geboten wird, z.B. anhand von Mustern typischer Lösungen.

Templates werden eingesetzt, um den Änderungsaufwand auch auf dem Implementationslevel zu reduzieren, indem durch das Trennen von Inhalt und Layout deren unabhängige Änderung und Wiederverwendung möglich wird. Die Darstellung der Inhalte kann auf diese Weise statisch an die Anforderungen des jeweiligen Templates angepasst werden, eine dynamische Adaption ist jedoch allein mit dem Templatekonzept nicht möglich. Für Identity sites ist diese Beschränkung zumeist hinnehmbar, weil eine dynamische Adaption selten erforderlich ist und die konkrete Visualisierung bewusst die Sicht des Betreibers wiedergeben soll. Die Wahrnehmung und Wirkung des Layout muss infolgedessen nicht universell einheitlich sein. Zum Zeitpunkt der Entwicklung des Layout von Identity sites werden die Bedürfnisse der Nutzer nur selten explizit berücksichtigt, obwohl die Zielgruppe in der Regel sehr genau bekannt ist.

Der Verzicht auf die dynamische Adaption birgt das Problem, dass die Darstellbarkeit der Inhalte nicht sichergestellt ist, sobald das Layout eines Templates Änderungen erfährt. Ist der inhaltliche Umfang der Identity sites gering, kann eingeschränkt manuell kontrolliert und korrigiert werden.

Im Vorfeld der Anwendungsentwicklung ist das Gewicht der Entwicklungsdimensionen zu entscheiden. Die eigentliche Identity site Entwicklung beginnt mit dem Entwurf des Layout / Playout Schema, wobei Ziele wie die Visualisierung einer Corporate Identity berücksichtigt werden. Wird das Schema auf dem Implementationslevel entwickelt, ist dessen Ergebnis eine Menge statischer Templates, welche für die Präsentation von Inhalten verwendbar sind. Werden mehrere Templates entwickelt, ist deren Verwendung und ggf. deren Gewicht zu definieren, um eindeutig die Darstellung entscheiden zu können. Dies setzt auch die Existenz einer grundlegenden Struktur der Inhalte aber auch Entscheidungen über deren zukünftige Bindungen voraus. Wie in Abbildung 2.14 dargestellt, können anschließend die Inhalte bzw. deren Struktur spezifiziert werden. Hierbei werden auch die Darstellungsmöglichkeiten des zugehörigen Templates berücksichtigt. Das Problem einer derartigen Spezifikation besteht darin, dass sich der inhaltliche Umfang der Website aber auch die Platzierungsmöglichkeiten durch ein Re-Design erheblich ändern können, Mängel hinsichtlich der Eignung auf dem Implementationslevel jedoch weder erkannt noch behandelt werden können.

In einem letzten Schritt wird das Functionality-Schema spezifiziert, um die Erreichbarkeit der Inhalte zu gewährleisten und zusätzliche Funktionen zur Verfügung zu stellen, z.B. die Suche. Die Möglichkeiten, die sich durch die Spezifikation dieses Schemas erzielen lassen, hängen stark von der Struktur der Inhalte ab, aber auch von den Darstellungsoptionen, die das bereits spezifizierte Layout bietet. Wegen der geringen Flexibilität und Leistungsfähigkeit des Layout umfasst dieses Schema zumeist nur die Entwicklung der Navigation. Vor allem bei Identity sites geringen Umfangs besteht jedoch selten zwingender Bedarf weiterführende Funktionalität zu ermöglichen.

Layout/Playoutschema

↘
↘
Content-Tree

↘
↘
Functionality-Schema

Abbildung 2.14: Schema Entwicklung bei Identity sites

2.4.4 Election sites (hybrid)

Nicht alle Websites lassen sich unproblematisch einigen, wenigen Website-Haupttypen zuordnen. Neben wenigen Haupttypen werden daher weitere Spezialtypen benötigt, die aus der Verbindung mehrerer Haupttypen hervorgehen. Ein Beispiel hierfür sind Election sites, die das Ziel verfolgen, eine sich zur Wahl stellende Person geeignet zu präsentieren. Dieser Website-Typ ist hauptsächlich durch die Herangehensweise der Identity sites geprägt, wird jedoch auch durch Infotainment und eBusiness Konzepte beeinflusst.

Ein wesentlicher Unterschied zu den Identity sites besteht darin, dass die Wirkung der Inhalte auf die Zielgruppe einen wesentlich höheren Stellenwert besitzt. Anders als bei diesen wird nicht vorrangig eine konkrete Visualisierung angestrebt, sondern eine genau definierte Wirkung durch den Abgleich mit der Wahrnehmung der Zielgruppe. Die möglichst einheitliche Wirkung bildet die Grundlage für die Präsentation der Ideen, Vorstellungen und Lösungen. Auch die Story Dimension ist ein wichtiger Bestandteil von Election sites, da die Inhalte den Benutzern in einer nicht zufälligen Reihenfolge zugänglich gemacht werden sollen, um den eigenen Standpunkt glaubhaft machen und dessen Bedeutung vermitteln und stärken zu können. Für die Benutzung von Election sites müssen wenige Kenntnisse, Fähigkeiten und Fertigkeiten notwendig sein, damit eine möglichst breite Zielgruppe erreicht werden kann. Dabei gilt es zu beachten, dass auch eine geringe Komplexität der Darstellung sowie ein geringer Informationsumfang die Benutzungskomplexität senkt.

Wie in Abbildung 2.15 dargestellt, lassen sich allgemein zwei Hauptentwicklungsrichtungen unterscheiden, wobei die Intention implizit definiert werden kann und die Spezifikation der Context Dimension optional ist. Falls die Context Dimension spezifiziert wird, so beeinflussen dessen Festlegungen alle weiteren Dimensionen. Bei sehr geringen, inhaltlichen Umfängen kann im Anschluss an eine grobe Story Spezifikation jeweils die Presentation Dimension entwickelt werden. Diese Herangehensweise ist unter anderem dann von Vorteil, wenn die Story über nur eine Szene mit einer Sequenz weniger Dialogschritte verfügt, die ihrerseits wenige und in ihrer Struktur einfache Inhalte enthalten.

Mit steigendem Umfang des darzustellenden Inhalts steigen auch die Anforderungen an die Präsentation. Aufgrund der Bedeutung der Wirkung der Inhalte auf die Zielgruppe

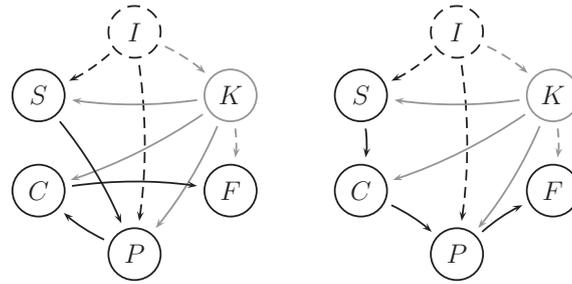


Abbildung 2.15: Dimensionen bei Election sites

ist daher bei umfangreichen oder komplexen Inhalten die Content Dimension vor der Presentation Dimension zu spezifizieren. Durch die Functionality Dimension von Election sites wird primär die Navigation definiert. Daneben ist es Aufgabe von Funktionen, die Benutzung des Webauftritts zu erleichtern und die Wirkung der Inhalte zu stärken. Zu diesen Funktionen zählen unter anderem das Bereitstellen einer Suchfunktion als auch redundante Interaktionsmöglichkeiten, um eine möglichst flexible Kommunikation mit dem System zu ermöglichen.

3 Screenography

Die Wahrnehmung einer Darstellung und deren Wirkung wird nicht allein durch objektive Faktoren bestimmt. Subjektiv beeinflussend wirken unter anderem kulturelle Prägungen als auch persönliche Präferenzen, Besonderheiten und Kenntnisse. Eine für den jeweiligen Nutzer geeignete Darstellung erfordert daher immer die Berücksichtigung seiner individuellen Bedürfnisse, Aufgaben und Ziele.

Sowohl die Wahrnehmung als auch die Erwartungen eines Nutzers an die Darstellung unterliegen permanenten Änderungen. Ein in hohem Maße adaptives System darf dabei nicht die vom Nutzer zugleich erwartete Kontinuität beeinträchtigen, da andernfalls die weitere Nutzung und Orientierung erschwert würde. Ziel der Adaption ist somit nicht die isolierte Optimierung einzelner Darstellungen. Die Suche einer geeigneten Darstellung von Inhalten sollte daher stets den erwarteten und möglichen Handlungsverlauf einer Teilaufgabe oder Szene berücksichtigen. Durch die Gegenüberstellung typischer Handlungsverläufe und der tatsächlichen Benutzungshistorie können nicht nur geeignete Lösungen gefunden, sondern auch Tendenzen erkannt und bewertet sowie Prognosen erstellt werden.

Die Screenography hat die systematische Entwicklung und Gestaltung von Websites zum Ziel. Sie ist ferner bestrebt, Inhalte und Funktionalität unter Beachtung bestehender Präsentationsmöglichkeiten und -präferenzen geeignet darzustellen. Lebensfälle, wie in [56] eingeführt und untersetzt, helfen in diesem Zusammenhang typische Handlungen und Situationen zu spezifizieren, um die pragmatische Lücke zwischen der Intention und dem Storyboarding [57] zu schließen.

Aus der Spezifikation der Anwendung, der Benutzer und des Equipments ergeben sich allgemeine Anforderungen an die Gestaltung von Websites, die systematisch zu erfassen und auszuwerten sind. Eine bessere Überschaubarkeit der existierenden Stellgrößen lässt sich erreichen, wenn sich allgemeine Dimensionen finden lassen, denen die ermittelten Einflussfaktoren zugeordnet werden können. Infolgedessen wird die Unterscheidung von 6 allgemeinen Entwicklungsdimensionen von Webinformationssystemen nach [55] verwendet. Die allgemeinen Grundlagen für das Zusammenwirken aller sechs Dimensionen auf unterschiedlichen Abstraktionsniveaus im Abstraktionsschichtenmodell [65] wurden in [66] definiert. Die Entwicklung des Layout ist nicht allein Gegenstand der Präsentationsdimension, weshalb allgemein die Berücksichtigung aller 6 Dimensionen erforderlich wird. Es wurde jedoch gezeigt (Abschn. 2.2), dass das Gewicht der Dimensionen abhängig von der Anwendung ist und überdies Abhängigkeiten zwischen den Dimensionen bestehen. Da das Resultat des Generierungsprozesses von der Anwendungsreihenfolge und Gewichtung der Dimensionen abhängt, sind diese Angaben zu Beginn des Entwicklungsprozesses zu definieren. Dazu können Präferenzen, intentionale Beschreibungen sowie Struktur- und Inhaltsanalysen genutzt werden [43].

3.1 Ursprung und Ziele

Der Begriff *Screenography* geht aus der Kombination der Begriffe '*Screen*' und '*Scenography*' hervor. Diese Bindung soll verdeutlichen, dass die Screenography als Pendant der Szenographie für Web-Anwendungen zu verstehen ist. Die dennoch bewusste, begriffliche Abgrenzung geht auf die unterschiedlichen Anforderungen, Möglichkeiten und Grenzen beider Begriffe zurück. Die Screenography soll somit nicht der 1:1 Übertragung szenographischer Konzepte in die elektronische Welt entsprechen, sondern auch die Besonderheiten und erweiterten Möglichkeiten der Gestaltung von Web-Anwendungen berücksichtigen und verwenden [6, 60].

HCI – Human-Computer-Interaction

Die Screenography ist bestrebt, die Lücke zwischen dem Web-Engineering und vorhandenen HCI-Ansätzen zu schließen, um auch im Falle großer Datenmengen und Datenvielfalt, Inhalte geeignet darstellen zu können. Dazu ist die Darstellung in einem hohen Maße an die Bedürfnisse der Nutzer und Vorgaben der Provider anzupassen. Orientiert sich die Darstellung der Inhalte ferner an deren Möglichkeiten und denen des Ausgabe-mediums, lassen sich insbesondere falsche und ungeeignete Darstellungen verhindern.

Darstellungsregeln werden bislang überwiegend auf dem Implementationslevel entwickelt, wodurch eine enge Bindung zur jeweiligen Umsetzung entsteht. Dieses Vorgehen erlaubt eine schnelle Realisierung, entsprechend der Fähigkeiten der gewählten Darstellungssprache, beschränkt aber stark die Gültigkeit und Wiederverwendbarkeit solcher Regeln. Auf diesem Abstraktionsniveau kann die anwendungsweite Konsistenz der Regeln nur selten fortgesetzt sichergestellt werden, aufgrund fehlender Konzepte, unzureichender Prüfmöglichkeiten und einer starken Abhängigkeit von den Fähigkeiten des Entwicklers. Dieser Umstand ist vor allem für die Weiterentwicklung einer Anwendung problematisch. Daher überrascht die häufige Beobachtung der Praxis kaum, dass sich die Stimmigkeit guter, initialer Entwürfe durch Änderungen sukzessive verschlechtert.

Ziel der Screenography ist eine frühe Berücksichtigung von Layoutaspekten innerhalb des Entwicklungsprozesses, sodass Entwickler während des Entwurfs oder einer Revision rechtzeitig über Darstellungsprobleme sowie mögliche Grenzen informiert werden können. Layoutaspekte sind dabei mindestens auf dem konzeptionellen Level zu betrachten, um Neugestaltungen ohne vollständige Neuentwicklung zu ermöglichen. Infolgedessen werden Ausgaben auf verschiedenstem Equipment realisierbar, unter Vermeidung kostenintensiver, paralleler Entwicklungen auf dem Implementationslevel. Darüber hinaus lassen sich Darstellungskonzepte entwickeln, die nicht nur lokale Anforderungen zu präsentierender Informationen, sondern auch globale Bedürfnisse einer Handlung berücksichtigen.

Neben den Aufgaben der Nutzer und den Intentionen der Anwendung berücksichtigt die Screenography individuelle Fähigkeiten und Fertigkeiten sowie Präferenzen der Nutzer. Diese Faktoren lassen sich innerhalb von Profilen und Portfolio [54] definieren. Während die Intention zusammen mit den Aufgaben die objektiven Erwartungen und Interessen bestimmt, können die subjektiv verschiedenen Faktoren die Geduld des Nutzers beim Umgang mit dem System beeinflussen. Hierbei lassen sich sowohl die Erfahrungen

aus dem Bereich des HCI-Design [3, 48] als auch die der Kognitiven Psychologie [40] nutzen.

Szenographie und Dramaturgie

Die Screenography hat individualisierte, gestaltete Darstellungen zum Ziel, welche den Kontext, die Fähigkeiten des Equipment, die geforderte und benötigte Funktionalität sowie den Fortgang in der Handlung berücksichtigen. Dazu erweitert die Screenography das Web Application Engineering um Aspekte der Dramaturgie und Szenographie und kann auf diese Weise die Interaktion zwischen dem System und seinen Nutzern unterstützen [35].

Die Szenographie hat ihre Wurzeln im Bereich von Theater, Film und Fernsehen und umfasst die kunstvolle Inszenierung von Gestaltungsräumen innerhalb derer konkrete Darstellungen oder Szenen stattfinden. Die tatsächliche Ausgestaltung solcher Räume hängt unter anderem von der Art der Handlung, der beabsichtigten Botschaft sowie der gewünschten Dramaturgie ab.

Die Screenography greift Konzepte der Szenographie auf, um Nutzern auch bei Webanwendungen Orientierung und den Inhalten einen angemessenen Rahmen zu bieten. Im Unterschied zur klassischen Inszenierung besteht jedoch die Möglichkeit, auf Grundlage des jeweiligen Nutzerprofils und unter Berücksichtigung der Benutzung der Anwendung, individuelle Ausprägungen eines Gestaltungsvorhabens zuzulassen.

Die Dramaturgie hat die Aufgabe, das Interesse des Betrachters an der Handlung zu wecken, aber auch dessen Aufmerksamkeit fortgesetzt zu erhalten. In Abhängigkeit vom Typ der Handlung werden dazu erprobte Handlungsverläufe eingesetzt, die eine definierte Wahrnehmung beim Betrachter provozieren und damit eine Stimmung vermitteln. Die Dramaturgie steuert sowohl die Abfolge der Szenen einer Handlung im Großen als auch die Komposition der zu präsentierenden Informationen im Kleinen. Der Verlauf einer Handlung wird in der Regel durch den Drehbuchautor und den Dramaturg definiert. Ursprünglich erfolgte dies ausschließlich und vollständig vor dem Beginn der Handlung. Moderne Formen erlauben jedoch auch eine unvollständige Beschreibung, die während der Handlung fortgeschrieben werden kann. Auch die Screenography soll diesbezüglich nicht beschränkend sein.

Im Unterschied zu den konventionellen Handlungen wird bei der Screenography der tatsächliche Handlungsverlauf durch den Nutzer bestimmt. Voraussetzung hierfür ist, dass der Entwickler der Anwendung eine Story definiert, innerhalb welcher alternative Handlungspfade existieren und die Zulässigkeit deren Nutzung durch Akteure entschieden wurde. Erweiternd kann ein Rechtesystem eingeführt werden, das die von einem konkreten Akteur nutzbaren Möglichkeiten innerhalb einer definierten Story bestimmt.

Mitunter kann die Aufgabe der Storydefinition auch bedingt auf die Nutzer übertragen werden. Innerhalb von Lernumgebungen kann dies hilfreich sein, um bestehende Daten und Teilhandlungen neu rekombinieren und damit die Präsentation der Informationen besser an die Bedürfnisse unterschiedlicher Lernertypen anpassen zu können. Auf diese Weise lassen sich individuelle Lerneinheiten schaffen, ohne diese selbständig spezifizieren zu müssen. Die Analyse der Verwendung solcher user-defined Stories kann zudem

Tendenzen hinsichtlich des Bedarfs und der Eignung aufzeigen. Ferner lassen sich neu gefundene Verlaufsformen direkt auf weitere Handlungen anwenden. Die Gültigkeit herausbildbarer Rekombinationen hängt stark von der gewählten Detailstufe ab, die den Nutzern zur Rekombination zur Verfügung steht. Sie ist sichergestellt, solange jede rekombinierbare Teilhandlung einer gewählten Detailstufe zu keiner Zeit Ergebnisse anderer Teilhandlungen implizit, aufgrund des Handlungsverlaufs, oder explizit, durch deren Verwendung, aufgreift. Da diese Forderung oft zu restriktiv ist, kann Flexibilität durch eine ergänzende Definition akzeptabler Bindungsanforderungen geschaffen werden (Kap. 2.1.2). Eine solche Definition kann auch genutzt werden, um die Sinnhaftigkeit rekombinierter Handlungen sicherstellen.

3.2 Gestaltungsgrundlagen

Grundlage der Entwicklung einer systematischen Gestaltung für Informationssysteme sind die Prinzipien der visuellen Darbietung von Informationen. Neben der Wahrnehmungsphysiologie und Wahrnehmungspsychologie sind auch Aspekte der Kognitiven Psychologie [8, 9, 40] zu berücksichtigen. Psychologische Aspekte sind abhängig von den Erfahrungen des jeweiligen Betrachters. Ein Rückgriff auf diese Faktoren innerhalb der Gestaltung ist daher nur dann empfehlenswert, wenn sie vom überwiegenden Teil der Nutzer fehlerfrei verstanden werden. Hierfür müssen die Profile der Betrachter(gruppen) [2] detailliert bekannt sein.

Ausgehend von der kleinsten Gestaltungseinheit, dem Punkt, können durch Rekombination grundlegende Elemente abgeleitet werden. Nach Kleint [25] existieren die Formelemente *Punkt*, *Linie*, *Fläche*, *Körper* und *Raum*, die schrittweise in ihrer Ausdehnung zunehmen, jedoch an Bestimmtheit verlieren. Aufgrund ihrer Sonderstellung hinsichtlich Entstehung, Verwendung und Bedeutung ergänzte Moritz [33] im Zweidimensionalen *Schriften*, *Strukturen* und *Bilder* und bezeichnet die Gesamtliste als Gestaltungselemente. Durch die Begriffsänderung löste sich auch die *Form*-Eigenschaft besser vom eigentlichen Elementbegriff.

Gestaltungselemente lassen sich abstrakt und damit unabhängig von Ihrer realen Darstellung definieren. Dies ermöglicht, auf dem Implementationslevel gleiche Inhalte in unterschiedlicher Weise darzustellen. Eingeschränkt wird dies mit *Stylesheets* bereits genutzt, beispielsweise durch Änderung der Farbwerte ausgewählter Elemente in Abhängigkeit von der durch den jeweiligen Nutzer gewählten Farbpalette.

Constraint 1. *Gestaltungselemente sind grundlegende Elemente einer Gestaltung, die sich unabhängig von Ihrer realen Darstellung definieren lassen und damit bereits einfache Adaption auf dem Implementationslevel zulassen.*

Die physiologische Wahrnehmbarkeit der Gestaltungselemente ist an die Definition von Gestaltungsmitteln [33, 41] gebunden. Gestaltungsmittel charakterisieren Elemente sowie Gruppen von Elementen losgelöst oder abhängig von anderen Elementen oder Elementgruppen. Zu den zentralen Eigenschaften eines Elements zählen dessen Größe, Form

und Farbe. Verfeinerungen dieser Gestaltungsmittel können hilfreich sein, um Besonderheiten eines Elements besser herauszuarbeiten, z.B. durch die Angabe des Materials oder der Helligkeit des Elements. Die Gestaltungsmittel Anzahl, Kontrast und Komposition sind nutzbar sobald eine Gestaltung mehrere Elemente besitzt. Ferner ist durch den Rhythmus eine verfeinerte Spezifikation der Komposition von Elementen möglich. Die intensive Nutzung des Rhythmus setzt jedoch Kenntnisse über die Zielgruppe voraus, da die Wahrnehmung von Rhythmen von den Erfahrungen des jeweiligen Nutzers abhängig ist. Eine besondere Bedeutung besitzen die Gestaltungsmittel Proportion und Perspektive. Die Aufgabe der Proportion ist es, unabhängig von den Gestaltungselementen die Grenzen der Gestaltung zu visualisieren. Typischerweise wird dies erreicht, indem sich die Farben der Gestaltungsgrenze und ihrer äußeren Umgebung unterscheiden. Die Perspektive legt den Blickwinkel auf die Gestaltung fest und beeinflusst damit die Visualisierung aller Gestaltungselemente.

Constraint 2. *Gestaltungsmittel charakterisieren Gestaltungselemente sowie Gruppen von Gestaltungselementen losgelöst oder abhängig von anderen Elementen oder Elementgruppen.*

Farben kann oft eine sehr große Bedeutung beigemessen werden, da sie als Gestaltungsmittel nicht nur die Wirkung einzelner Elemente, sondern auch maßgeblich die Stimmung der gesamten Gestaltung beeinflussen können. Infolgedessen war die Dominanz von Farben gegenüber anderen Farben bereits im frühen 19. Jahrhundert Teil der Farbenlehre von J. W. Goethe [16]. Das Ziel verfolgend, der Malerei die Auswahl und Zusammenstellung geeigneter Farben zu erleichtern, entwickelte Itten später einen zwölfteiligen Farbkreis [22]. Ferner arbeitete Itten Farbkontraste und Farbwirkungen heraus, die bei der Farbzusammenstellung und Farbverwendung zu berücksichtigen sind. Die Wirkung von Farben und Farbkombinationen auf den Betrachter ist jedoch nicht allgemeingültig und partiell konträr. Oft ist die Wirkung stark an individuelle Präferenzen und kulturelle Prägungen gebunden [10]. Die Zielgruppe einer Gestaltung sollte daher stets bekannt sein. Durch die Definition der gewünschten Farbwirkung anstelle der finalen Farbgebung kann eine entscheidende Abstraktion erzielt werden, um die Anforderungen an die spätere Gestaltung unabhängig vom jeweiligen Nutzer beschreiben zu können.

Das jeweils erste Gestaltungselement einer Gestaltung – der Gestaltungsraum – entsteht durch die Definition der Grenzen der Darstellung. Der hier verwendete Gestaltungsraum ist somit stets begrenzt, im Gegensatz zum allgemeinen Raum nach Kleint [25]. Die Beschränkung ist notwendig, da sich andernfalls die Wirkung von Gestaltungsmitteln, wie z.B. die der Proportion, nur bedingt fixieren lässt. Sie führt aber zugleich zu einer Beschränkung des maximal darstellbaren Informationsumfangs, infolgedessen die Darstellung übergroßer Contentobjekte zu behandeln ist. In Abhängigkeit von der jeweiligen Definition eines solchen Contentobjekts, kann dem Betrachter die Information jedoch typischerweise sukzessive zugänglich gemacht werden.

Neben dem Gestaltungsraum müssen auch Gestaltungselemente Grenzen besitzen, um die wahrnehmbare Eigenständigkeit zuzusichern. Eine Kreisfläche, die sich innerhalb einer anderen Kreisfläche befindet, wird jedoch bereits dann unsichtbar, wenn sich die

Position der Übergangskante zwischen den gefärbten Flächen auf der Netzhaut nicht verändert [28]. Dies gilt weitestgehend unabhängig von der Färbung der Flächen, solange die Luminanz der inneren Fläche geringer als die der Äußeren ist. Demnach ist die Wahrnehmung der Fläche nur möglich, wenn eine begrenzende Kante existiert, die in Ihrer Funktion wirkungsvoll ist [63]. Im Unterschied zur geforderten Wirkung muss eine begrenzende Kante nicht physikalisch existieren, da durch Vervollständigung von Haltepunkten auch Scheinkonturen [58] entstehen können. Ein typischer Vertreter hierfür ist das sogenannte *Kanizsa Dreieck* [24].

Constraint 3. *Im Unterschied zum allgemeinen Raum nach Kleint [25], wird der hier verwendete Gestaltungsraum als stets beschränkt betrachtet. Daher ist dessen Kapazität, Inhalte zu einem festgelegten Zeitpunkt darzustellen, ebenfalls beschränkt.*

Neben dem Gestaltungsraum ist auch das Aufnahme- und Speichervermögen der Betrachter stark begrenzt, situationsabhängig und individuell verschieden [32, 50]. Unter Berücksichtigung ihrer Komplexität [17], ihres Abstraktionsgrades, jedoch insbesondere ihrer Relevanz und Gültigkeitsdauer sind Informationen demnach auch in der Menge und Aufnahmegeschwindigkeit des jeweiligen Betrachters zur Verfügung zu stellen, um Informationsverlust zu vermeiden. Hierzu zählt unter anderem der maximale Umfang von Contentobjekten, die dem Betrachter zu einem festgelegten Zeitpunkt präsentiert werden können. Diese individuelle Grenze lässt sich durch Unterstützungsfunktionen bedingt ausweiten. So kann die abstrakte Darstellung des Bearbeitungsstandes einzelner Teilaufgaben dem Akteur einen einfachen Überblick über den aktuellen Bearbeitungsstand der Gesamtaufgabe geben und ihn zugleich hinsichtlich noch zu erledigender Aufgaben leiten.

Constraint 4. *Das Aufnahme- und Speichervermögen eines jeden Nutzers ist situationsabhängig, stark begrenzt und individuell verschieden. Eine echte Adaption der Inhalte in Form, Umfang und Geschwindigkeit ist daher nur dann möglich, wenn neben dem Profil des jeweiligen Nutzers auch der Kontext der Nutzung bekannt ist. Da beide Größen dynamisch sind, ist zudem die Historie der Benutzung zu berücksichtigen, um auch während der Benutzung auf Änderungen reagieren zu können.*

3.3 Lessons Learned

Im Bereich des Software-Engineering konnten sich eine Reihe von Modellen etablieren, die den Software-Entwicklungsprozess geeignet unterstützen. Der Versuch der Adaption dieser Modelle entsprechend der Bedürfnisse des HCI-Design stößt jedoch trotz existierender Bindungen zwischen dem Software-Engineering und Usability-Engineering oft auf Schwierigkeiten. Wenngleich der Bedarf ähnlicher Schritte existiert und sich somit die Anforderungen an ein generisches Vorgehensmodell entwickeln lassen [46], so ist gerade das Vorgehen im Bereich des HCI-Design stark an die individuellen Bedürfnisse der zu entwickelnden Anwendung gebunden und damit nicht allgemeingültig. Aufgrund der großen Zahl anwendungsbezogener Einflussfaktoren, kann auch die systematische Entwicklung und Gestaltung von Websites nicht mit Hilfe einer einzigen, universell gültigen

Vorgehensweise erreicht werden. Die Zerlegung in Teilkomponenten reduziert die Komplexität und ist hilfreich, solange diese Teilkomponenten weiterhin flexibel zusammenwirken können. Eine derartige Trennung bieten die in Abschnitt 2.1 beschriebenen 6 Dimensionen der WIS Entwicklung.

Unterschiede in der Wahrnehmung und Erwartung treten in besonderem Maße auf dem Implementationslevel hervor. Der Versuch der Behandlung solcher Situationen resultiert auf diesem Level üblicherweise in der Aufspaltung in mehrere Entwicklungszweige, deren Harmonisierung sich langfristig als problematisch erweist. Bislang fehlt die Möglichkeit einer allgemeineren Definition, die Anforderungen an die Gestaltung beschreibt, ohne konkrete Bedürfnisse der Nutzer oder Gegebenheiten einer speziellen Präsentationsform zu berücksichtigen. Die Unterscheidung von Gestaltungselementen und -mitteln zeigt aber, dass sich der Bedarf einer Gestaltung formulieren lässt, ohne die konkrete Ausprägung definieren zu müssen, welche die Verwendbarkeit der gefundenen Lösung stark beschränkt. Das allgemeine Potential übersteigt dabei die Fähigkeiten derzeitiger Template-Lösungen zur Erstellung und Gestaltung von Websites, da diese bislang nur wenige Gestaltungsmittel aufgreifen. Wenngleich die Unterscheidung von Elementen und Mitteln Individualität erlaubt, fehlt jedoch weiterhin die Möglichkeit der systematischen Entwicklung der Gestaltung auf dem konzeptionellen Level, um nicht nur lokal das Zusammenwirken von Elementen definieren zu können.

Selbst bei Außerachtlassung der Verschiedenheit der Nutzer ist eine vollständig automatisierte Berechnung geeigneter Gestaltungslösungen nicht möglich, ohne beschränkend hinsichtlich der Zahl verfügbarer und betrachteter Lösungen oder Lösungskonzepte zu sein. Darüber hinaus steigt mit zunehmender Anzahl der Optionen der Spezifikationsaufwand beträchtlich. Die Untersuchung einer Vielzahl bestehender Gestaltungen zeigt jedoch deutlich das Vorhandensein häufig wiederkehrender Lösungen, die sich nur im Detail unterscheiden. Eine Ursache für diese Häufung sind ähnliche Anforderungen an die Gestaltung. Zugleich stellt die große Menge ähnlicher Lösungen eine Hauptursache für Entscheidungsprobleme automatisierter Lösungen dar. Insbesondere weil der Lösungsraum bei ähnlichen Kern-Anforderungen typischerweise nur eine geringe Spreizung und Größe aufweist, erscheint eine abstrakte Spezifikation der Anforderungen attraktiv. Diese kann Muster definieren, um Cluster ähnlicher Lösungen zu beschreiben. Ferner kann auf diese Weise bereits frühzeitig eine Selektion geeigneter Lösungsräume erfolgen.

Gestaltungsräume begrenzen ein Gestaltungsvorhaben und lassen sich unabhängig von ihrer konkreten Nutzung beschreiben. Da eine Anwendung typischerweise mehrere Räume nutzt, sind Beziehungen erforderlich, die den Übergang zwischen bzw. das Erreichen von anderen Räumen ermöglichen. Das gesamte Raumarrangement eines Gestaltungsvorhabens kann als Muster fungieren, indem dessen Besonderheiten sowie typische Anwendungsgebiete definiert werden. Auf diese Weise kann trotz der Unabhängigkeit von der konkreten Nutzung die Wiederverwendung vorhandener Lösungen erleichtert werden.

Voraussetzung für die Nutzung von Gestaltungsräumen durch Akteure ist deren Anreicherung durch Handlungen, Ausgestaltungs- und Interaktionskonzepte (Abb. 3.1).

Bei der Entwicklung einer Anwendung ist, aufgrund wechselseitiger Beziehungen zwi-

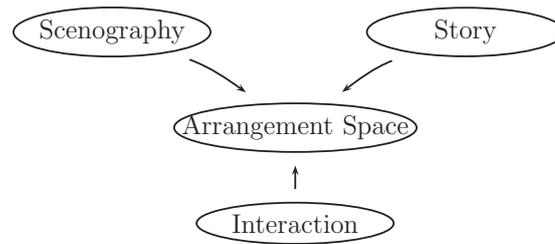


Abbildung 3.1: Gestaltungsraum Einflüsse

schen den Einflussgrößen und dem Raumarrangement, neben der Prüfung der Eignung gewählter Konzepte auch die Entscheidungshoheit bei Konflikten zu entscheiden. Ferner ist die Reihenfolge der Berücksichtigung der Einflussgrößen von Belang, da sich diese auch gegenseitig beeinflussen. Die Herausbildung eines initialen Raumarrangements wird typischerweise von äußeren Größen dominiert. Deren Einfluss nimmt im Zuge der Weiterentwicklung bestehender Anwendungen kontinuierlich ab.

3.4 Raum und Handlung

Mit dem Ziel einer systematischen Entwicklung und Gestaltung von Websites unterscheidet die Screenography die Spezifikation von Gestaltungsräumen von jener der Handlungen. Gestaltungsräume fungieren als begrenzende Rahmen eines Gestaltungsvorhabens und definieren die Möglichkeiten, Inhalte platzieren, komponieren und präsentieren zu können. Neben den Grenzen der Eignung lässt sich die gewünschte Benutzung von Gestaltungsräumen definieren, um deren Wiederverwendung zu erleichtern. Eine Handlung bestimmt hingegen, in welcher Reihenfolge und unter welchen Bedingungen den Akteuren innerhalb von Szenen Contentobjekte zugänglich gemacht werden. Handlungen können folglich unabhängig von jenen Räumen definiert werden, innerhalb derer sie erlebbar werden.

Ähnlich der Unterscheidung zwischen Gestaltungselementen und Gestaltungsmitteln lässt sich auch für Handlungen *Was* präsentiert wird von dem separieren *Wie* dies geschieht. Eine Handlung wird dadurch unabhängig von den Möglichkeiten, sie kunstvoll zu inszenieren. Andererseits erleichtert ein unabhängiges Raumkonzept nicht nur die überprüfbare und flexible Integration mehrerer Handlungen, sondern auch deren Harmonisierung sowie die frühzeitige Planung von Änderungen. Darüber hinaus bietet ein stimmiges Raumkonzept, durch geeignete Anordnung und Verbindung der Räume, den Nutzern die Möglichkeit, sich auch unabhängig von einer konkreten Handlung zu orientieren.

3.4.1 Beziehungen von Gestaltungsräumen

Ein Gestaltungsraum ist eine abgeschlossene Einheit, dem ein Thema, eine Aufgabe oder eine Funktion zugeordnet werden kann. Aufgrund der begrenzten Kapazität eines

Raumes sind in der Regel mehrere Räume für die Abbildung und Inszenierung einer Handlung erforderlich.

Der in dieser Arbeit verwendete Raumbegriff lehnt sich an dessen Verwendung innerhalb der Architektur an, da sich bei der Planung eines elektronischen Informationsgebäudes ähnliche Herausforderungen ergeben, wie beim klassischen Vorbild (Abb. 3.2). Dies ermöglicht die frühzeitige Erstellung einer abstrakten Struktur der Anwendung, die Grundlage für Weiterentwicklungen ist. Die strukturelle Anordnung kann auch innerhalb des elektronischen Informationsgebäudes durch die Zuordnung von Aufgaben und Ausstattungsmerkmalen sowie der Definition geeigneter Anwendungssituationen beeinflusst werden.

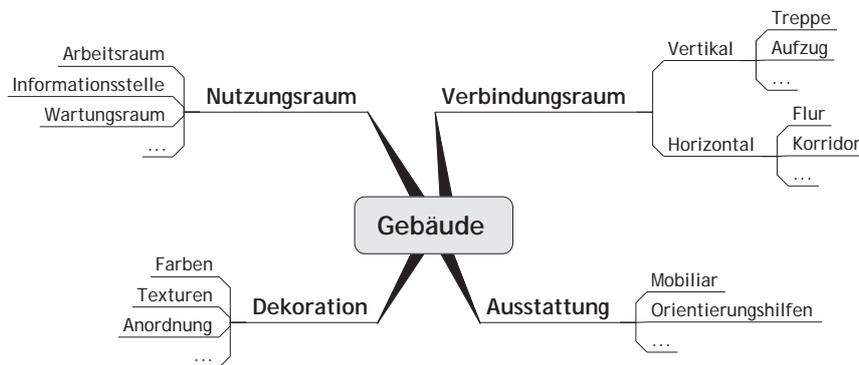


Abbildung 3.2: Räume in der Architektur

Die abstrakte Raumstruktur wird typischerweise nicht vollständig und explizit kommuniziert, sondern ist implizit während der Nutzung der Anwendung wahrnehmbar und erlernbar. Folglich gilt es die Akteure dabei zu unterstützen, schrittweise ein individuelles, mentales Modell herausbilden zu können. Der Erfolg hängt von der Komplexität der Struktur, der Häufigkeit der Nutzung der Anwendung und von den Fähigkeiten des jeweiligen Akteurs ab. Weitere Voraussetzung sind eine hinreichende, strukturelle Kontinuität sowie eine bewusste und auch geeignete Anordnung der Räume. Während eine stabile Anordnung Orientierung bietet, erleichtert ein geeigneter Zusammenhang zwischen den Räumen die angestrebte Nutzung. Um Nutzern eine fortgesetzt stabile Erkennung und Zuordnung zu ermöglichen, sind auch innerhalb eines Gestaltungsraumes thematische, inhaltliche oder funktionale Änderungen nur in begrenztem Umfang sinnvoll.

3.4.2 Erschließung des Gestaltungsraums

Insbesondere im Falle großer Informationsumfänge kann nicht vorausgesetzt werden, dass ein Nutzer eine Anwendung gut oder gar vollständig kennt. Infolge dieses Umstandes

ist er auf Hilfestellungen zum schnellen Auffinden von Informationen angewiesen. In Abhängigkeit von der Häufigkeit der Änderungen, der Komplexität der Struktur und den Fähigkeiten des Akteurs kann eine Unterstützung einmalig oder wiederholt erforderlich sein.

Unterstützungsfunktionen können einem Akteur allgemeine Hilfestellungen zur Benutzung der Anwendung geben, ihn auf dem Weg zur Information begleiten oder aber ihn direkt zum Ziel führen. Das unmittelbare Führen zum Ziel ist dann sinnvoll, wenn dies der Erwartungshaltung des Akteurs entspricht und Kenntnisse über die Struktur der Anwendung auch im Falle des wiederholten Abrufs von Informationen nicht hilfreich oder unerwünscht sind. Die Dominanz oder alleinige Verwendung dieser Unterstützungsform ist zu vermeiden, da die Realisierbarkeit und Qualität einer solchen Umsetzung vom Umfang der Informationsquelle sowie den Fähigkeiten des Akteurs abhängig ist. Ein Beispiel hierfür ist die Bereitstellung einer Suchfunktion mit Hilfe eines Suchfeldes, in welches der Akteur Schlagworte eingibt und anschließend direkt zum Ziel geführt wird. Sobald Suchterme nicht stets eindeutige Ergebnisse liefern, werden begleitende Hilfestellungen erforderlich. Diese erlauben dem Akteur lokale Entscheidungen und erleichtern damit die Benutzung der Anwendung. Während die begleitende Hilfe zielgerichtet und problemorientiert die wiederholte Nutzung des Systems in gleicher bzw. ähnlicher Weise erleichtert, erlaubt es die allgemeine Unterstützung, grundlegende Strukturen und Techniken unabhängig von konkreten Problemstellungen zu erlernen. Hilfreich ist dies gerade dann, wenn eine flexible Nutzung des Systems gefordert wird.

Den größten Einfluss auf die Benutzbarkeit einer Anwendung hat deren Struktur. Oftmals werden Inhalte und Themen klassifiziert und einer Hierarchie zugeordnet, die bei Bedarf Verfeinerungen erfährt. Steigt die Informationsvielfalt der Anwendung, wird eine eindeutige Klassifikation schwieriger und deren Verständlichkeit zunehmend nutzerabhängig. Entsprechend ist mit zunehmender Hierarchietiefe ein ansteigender Unterstützungsbedarf der Akteure zu erwarten. Üblicherweise treten diese Probleme nicht initial auf, sondern als Begleiterscheinung im Zuge der Evolution einer Anwendung.

In der Architektur wird die geplante Nutzung und der möglichst barrierefreie Zugang zu einem Gebäude [12] insbesondere bei der Planung von Gebäuden mit einer zukünftig hohen Frequentierung berücksichtigt und innerhalb eines Erschließungskonzepts fixiert. Der Begriff der *Erschließung* lässt sich auf den Bereich der Webinformationssysteme übertragen, mit dem Ziel, Konzepte für die Erkundung eines Informationsgebäudes sowie dessen Informationsebenen zu entwickeln. Erschließungskonzepte sind dabei nicht an eine konkrete Handlung oder Information gebunden, sondern erlauben durch ihren Aufbau ein bestimmte Art der Nutzung und Unterstützung. Der Unterstützungsbedarf ist im Unterschied zu den Hierarchien nicht von der Evolution der Anwendung abhängig, sondern von der Komplexität der Erreichbarkeit.

Zu den Zielen eines solchen Konzepts zählt die Harmonisierung existierender Handlungen innerhalb der Anwendung. Die Entwicklung eines Erschließungskonzepts ist daher nicht sinnvoll, wenn dies nicht möglich oder sogar unerwünscht ist. Durch die Trennung der Raumspezifikation von den Handlungen kann zu jedem Zeitpunkt geprüft werden, ob alle Handlungen einer Anwendung dem Erschließungsgesamtkonzept entsprechen. Änderungen an den Räumen einer Handlung können infolge von Überschneidungen an-

dere Handlungen beeinflussen. Konsequenzen geplanter Änderungen lassen sich durch die Trennung vorab prüfen, um fortgesetzt die Erreichbarkeit von Informationen als auch die Benutzbarkeit und Nachvollziehbarkeit des Gesamtkonzepts durch den Nutzer sicherzustellen.

Die häufige Änderung eines Erschließungskonzepts ist zu vermeiden, da dies die Orientierung der Akteure stark beeinflusst. Ist der Bedarf, Ort, Umfang oder Zeitpunkt eines geplanten Änderungsvorhabens frühzeitig bekannt, lässt sich jedoch auch dann ein Migrationskonzept entwickeln. Das direkte Kommunizieren angestrebter konzeptioneller Änderungen gegenüber den Nutzern ist dabei zumeist nicht realisierbar, da in der Regel weder der individuelle Akteur noch dessen Informationsstand bekannt ist. Ein Erschließungskonzept kann aber Akteure direkt zum Zeitpunkt der Nutzung unterstützten, präventiv, durch geeignete Hinweise, als auch reaktiv, bei Erkennen eines geeigneteren Nutzungsmusters.

In Abgrenzung zum Raumbegriff der Architektur kann sich die Anzahl und Aufgabe von Räumen innerhalb einer Webanwendung in Abhängigkeit vom Nutzer unterscheiden. Das Wissen über die Existenz einzelner Gestaltungsräume muss innerhalb eines Webinformationssystems somit nicht global verfügbar sein. Ein einheitliches Erschließungskonzept kann jedoch trotz einer teilweise nutzerabhängigen Erschließung entwickelt werden.

Allgemein lässt sich wie in der Architektur, die *Horizontale Erschließung* von der *Vertikalen Erschließung* unterscheiden.

Horizontale Erschließung

Die horizontale Erschließung ist innerhalb der Architektur Teil der Planung von Korridoren und Fluren und beeinflusst daher entscheidend die Zugänglichkeit und damit die Nutzbarkeit der Räume. Aufgrund baulicher Beschränkungen kann die horizontale Erschließung nicht unabhängig von anderen Etagen eines Gebäudes geplant werden. Insbesondere betrifft dies die direkten Nachbarn einer Etage. In Abhängigkeit von ihrer Nutzungsart können Etagen entweder zentral (Wohnungsbau) oder linear (Hotels) erschlossen werden. Ausgehend von deren Grundformen lassen sich weitere Erschließungsvarianten ableiten, z.B. Sternstrukturen, bei denen die lineare der zentralen Erschließung folgt, oder Kammstrukturen, bei denen sich an einen Haupterschließungspfad weitere lineare Erschließungsmöglichkeiten anschließen.

Übertragen auf Webinformationssysteme dient die horizontale Erschließung als Rahmen für die Konzeption und Fortentwicklung von Szenen verfügbarer Stories. Im Unterschied zur Architektur kann die Kapazität einer Ebene im Bereich der Webinformationssysteme wesentlich flexibler definiert werden. Dies ist nicht nur losgelöst von anderen Ebenen möglich, sondern erlaubt auch eine stärkere Abstraktion, infolgedessen die Raumanzahl einer Ebene nicht zwingend vorab fixiert werden muss.

Die zentrale Erschließung kann gewählt werden, wenn es Informationen verschiedener Themenbereiche zu präsentieren gilt. Diese Erschließungsvariante eignet sich daher für Portalseiten im Bereich der Infotainment sites. Ist eine für die Zielgruppe nachvollziehbare, disjunkte Kategorisierung der vorhandenen Themen möglich, bietet dies den Akteuren Orientierung und erleichtert ihnen den Einstieg in eine Handlung. Mit der

Wahl eines Themenbereichs reduziert sich typischerweise auch die Zahl der Handlungen, anhand derer der Nutzer geeignet durch die Anwendung geführt werden oder seine Aufgabe erledigen kann. Die lineare Erschließung ist dann von Vorteil, wenn sequentielle Handlungen abzubilden sind. Beispielhaft sei hier das sukzessive Betrachten einer Bilder Sammlung genannt. Werden zudem Anfang und Ende eines solchen Bereichs verbunden, lassen sich auch iterative Handlungen realisieren.

Vertikale Erschließung

Treppen und Aufzüge dienen in der Architektur der Verbindung von Ebenen und damit der vertikalen Erschließung. Auch bei Webinformationssystemen kann die vertikale Erschließung für die Planung der Verbindung von Ebenen genutzt werden, um komplexe Handlungen aber auch Sprünge, Abkürzungen sowie Querverweise realisieren zu können.

Die mögliche Flexibilität ist dabei erneut größer als im Bereich der Architektur, da Bindungen vielfältiger und umfangreicher eingegangen werden können und Änderungen jederzeit möglich sind. Andererseits erschwert eine große Zahl von Bindungen deren angemessene Kommunikation bzw. Präsentation. Die Chance birgt somit zugleich die Gefahr einer zu starken Vernetzung der vorhandenen Räume, die schließlich die Orientierung der Akteure beeinträchtigt, wenn keine wahrnehmbare Trennung zwischen wesentlichen und zusätzlichen Erschließungsrouten erfolgt. Im Extrem nimmt ein Akteur ausschließlich Querverweise und Abkürzungen einer Anwendung wahr, die sich jedoch nicht zur Herausbildung eines mentalen Modells sowie bewusster Navigation eignen. Eine starke Vernetzung erfordert daher immer auch eine hinreichende Herausstellung und Abgrenzung der wesentlichen Erschließungsrouten. Hierfür ist zunächst die jeweils aktuelle Handlung vorrangig zu berücksichtigen. Erweisen sich auch alternative Handlungen als geeignet, so sollte auf deren Existenz nur dann hingewiesen werden, wenn dies das Erkennen der bisherigen Handlung nicht beeinträchtigt oder deren Eignung als deutlich besser bewertet wird. Die Deutlichkeit ist für die Entscheidung eines Wechsels zwingend, um bei ähnlich gut geeigneten Handlungen Toggle-Effekte zu vermeiden, da dies die von den Akteuren erwartete Kontinuität und Orientierung beeinträchtigt.

3.4.3 Raumdefinition

Vereinfacht stellt ein Raumarrangement eine Menge verbundener Räume dar und lässt sich damit als ungerichteter Graph repräsentieren. Ähnlich dem Analog der Realisierung eines Notausgangs in einem Gebäude kann auch für WIS die Notwendigkeit bestehen, den Übergang zwischen Räumlichkeiten bereits konzeptionell richtungsabhängig einzuschränken. Das Raumarrangement Ra wird daher in dieser Arbeit stets als gerichteter Graph betrachtet, dessen Knoten die existierenden Gestaltungsräume R repräsentieren deren Erreichbarkeit durch die Kanten E sichergestellt wird (Gl. 3.1).

$$Ra = (R, E), E \subseteq R \times R \quad (3.1)$$

Auf diese Weise kann die prinzipielle Erreichbarkeit von Räumen festgelegt werden, ohne bereits die tatsächliche Nutzbarkeit für einen konkreten Akteur oder eine bestimmte Handlung zuzusichern. Eine weitere Einschränkung ist damit möglich, nicht jedoch eine Ausweitung und damit Umgehung der Beschränkungen des Raumarrangements.

Auch die Weiterentwicklung eines bestehenden Raumarrangements Ra_{base} kann geplant und gesteuert werden. Mögliche Weiterentwicklungen können ebenfalls als Raumarrangement Ra_{evo} beschrieben werden und sind im einfachsten Fall innerhalb von Ra_{base} allgemein zulässig. Die allgemeine Zulässigkeit verhindert jedoch die unterschiedliche Behandlung gleichartiger Situationen. Sollen Regeln nicht für das gesamte Arrangement gelten, so können Ausgangsorte bestimmt werden, die als Referenzpunkt für die Prüfung und Einhaltung einer Entwicklung dienen. Lokale Entwicklungen $Ra_{ext_{static}}$ sind daher durch Beschränkung der Gültigkeit auf einen oder mehrere Räume $R_i \in Ra_{base}$ realisierbar (Gl. 3.2).

$$Ra_{ext_{static}} = (Ra_{evo}, R_i) \quad (3.2)$$

Darüber hinaus kann die Flexibilität durch das Loslösen von einer konkreten Raumanzahl weiter erhöht werden. Durch den Einsatz regulärer Ausdrücke lassen sich auf einfache Weise zulässige Raumfolgen beschreiben, die mögliche Weiterentwicklungen ausschließlich oder in Ergänzung zu den Ra_{evo} definieren. Die Gültigkeit solcher Weiterentwicklungen $Ra_{ext_{regex}}$ bzw. der Ausdrücke $Expr$ kann wiederum auf einen oder mehrere Räume $R_i \in Ra_{base}$ beschränkt werden (Gl. 3.3).

$$Ra_{ext_{regex}} = (Expr, R_i) \quad (3.3)$$

Durch explizite Kennzeichnung kann auch in diesem Fall ein beschränkender, richtungsabhängiger Übergang zwischen Räumen realisiert werden. Dies erlaubt konzeptionelle Beschränkungen der Nutzung und betrifft damit auch die mögliche Weitergabe von Informationen. Auf einfache Weise ist eine solche Beschränkung erreichbar, indem der jeweilige Übergang durch den Operator ' \rightarrow ' bzw. ' \leftarrow ' gekennzeichnet wird.

Exemplarisch erfordere der Ausdruck $Expr_1$ (Gl. 3.4) im Falle einer Erweiterung zwingend die Einrichtung eines Folgeraums R_a . Diesem darf sich eine beliebige Anzahl von Räumen des Typs R_b anschließen, solange diese ihrerseits ausschließlich den vorwärts gerichteten Übergang und damit die Weitergabe nicht jedoch die Rückgabe von Informationen erlauben. Schließlich sei die Erweiterung nur dann erlaubt, wenn abschließend genau ein Raum des Typs R_c vorhanden ist, mit dem die Erweiterung schließt.

$$Expr_1 = /(R_a)(R_b^*)(R_c)/ \quad (3.4)$$

Darüber hinaus kann die Weiterentwicklung auch explizit unterbunden werden, gerichtet durch die Operatoren ' \nrightarrow ' bzw. ' \leftarrow ' und gänzlich durch die Verwendung von ' \nleftrightarrow '. Dies erlaubt finale Beschränkungen der Weiterentwicklung und verhindert damit auch ungewollte, zukünftige Erweiterungen. Da der Ausdruck $Expr_2$ (Gl. 3.5) keine Erweiterungen zulässt, bei denen einem Raum R_a ein Raum R_b folgt, wird durch ihn $Expr_1$ nur noch dann erfüllbar, wenn dem Raum R_a unmittelbar Raum R_c folgt.

$$Expr_2 = /(R_a) \nrightarrow (R_b)/ \quad (3.5)$$

Ausdruck $Expr_3$ (Gl. 3.6) lässt grundsätzlich keine Erweiterungen zu, bei denen dem erweiternden Raum R_a Informationen von weiteren Räumen zurückgegeben werden können. Hieraus resultiert eine weitere Beschränkung möglicher Erweiterungen.

$$Expr_3 = /(R_a) \leftarrow / \quad (3.6)$$

Die größtmögliche Beschränkung besteht im Verbot jeglicher Erweiterungen und kann durch den Ausdruck $Expr_4$ (Gl. 3.7) erreicht werden. Der sinnvolle Einsatz dieses Ausdrucks setzt dabei voraus, dass der Ursprungsraum $R_i \in Ra_{base}$, dem dieser Ausdruck zugeordnet wird, innerhalb des bisherigen Raumarrangements bereits erreichbar ist.

$$Expr_4 = / \nleftrightarrow / \quad (3.7)$$

Raumarrangements als auch Ausdrücke können genutzt werden, um Muster zu beschreiben, die Raumlösungen für typische Problemstellungen darstellen. Einerseits erleichtert und beschleunigt dies die Entwicklung, andererseits lassen sich durch den Einsatz erprobter Lösungen Entwicklungsfehler vermeiden.

Die Erweiterung des Raumarrangements um neue Räume kann auf Basis eines Musters sowohl *statisch* als auch *dynamisch* erfolgen. Eine statische Lösung kann abstrakt zulässige Raumbeziehungen überprüfen oder aber sich konkret auf einen Raum bzw. eine Folge solcher Räume einschließlich ihrer Bindungen stützen. Eine eingeschränkte Dynamik lässt sich erreichen, indem für ein Erweiterungsvorhaben aus einer Menge zulässiger Räume oder Raumfolgen gewählt werden kann. Dabei kann das bewusste Erschweren der Vorhersagbarkeit der Entwicklung der Gestaltung durch die jeweils zufällige Auswahl der Erweiterungsoption die Attraktivität der Anwendung erhöhen. Beeinträchtigt dies jedoch die Nachvollziehbarkeit der Entwicklung, mindert dies die Attraktivität. Eine echt dynamische Entwicklung berücksichtigt bei der Wahl weiterer Raumsegmente auch die

Historie der Entwicklung. Im einfachsten Fall wird lediglich der jeweilige Vorgängerraum einer geplanten Erweiterung betrachtet. Um den Umfang des gesamten Erweiterungsreichs oder die aktuelle Position innerhalb dieses Bereichs visualisieren zu können, ist jedoch die gesamte Historie zu berücksichtigen.

3.5 Szenographie des Gestaltungsraums

Aufgabe der Szenographie ist die künstlerische Ausgestaltung von Räumen. Typischerweise sind die Räume und deren Bindungen eng an eine spezielle Handlung gebunden. Eine Inszenierung basierend auf einem Szenographiekonzept stellt auf diese Weise eine mögliche Ausprägung dar, welche nicht nur die Darbietung von Inhalten und damit deren Wirkung bestimmt, sondern auch die Sicht des Künstlers auf die Handlung offenbart. Eine solche Ausprägung wird durch allgemeine und spezielle Faktoren bestimmt, wobei diese global und lokal innerhalb der Anwendung wirken können.

Innerhalb der Screenography werden Szenographiekonzepte unabhängig von konkreten Handlungen, Inhalten und Darstellungsformen entwickelt. Einerseits ermöglicht dies den flexiblen und mehrfachen Einsatz bewährter Konzepte, einschließlich Kombination und Weiterentwicklung. Andererseits erlaubt eine abstrakte Bestückung und Ausgestaltung von Räumen das Berücksichtigen individueller Besonderheiten. Dies wird nicht nur der Heterogenität der Nutzer besser gerecht, sondern stärkt die Robustheit der Konzepte, insbesondere im Hinblick auf potentielle neue Anforderungen.

Die Aufgabe der Szenographie besteht somit darin, den allgemeinen Rahmen für die Visualisierung von Räumen und der darin enthaltenen Elemente festzulegen. Unter Berücksichtigung der möglichen Quantität und Komplexität von Inhalten eines Inhaltstyps entstehen hierbei auch Anforderungen an die Visualisierung, deren Ursache auf die Begrenztheit der Gestaltungsräume zurückgeht. Probleme infolge einer hohen Quantität von Elementen eines Inhaltstyps lassen sich typischerweise durch Kategorisierung reduzieren, indem jeder geschaffenen Kategorie ein Gestaltungsraum exklusiv zur Verfügung gestellt wird, sich jedoch für alle Kategorien die Art der Visualisierung gleicht. Bei großer Komplexität oder großem Umfang einzelner Elemente entsteht hingegen der Bedarf, Inhaltstypen über mehrere Gestaltungsraumgrenzen hinweg ausdehnen zu können.

Infolgedessen sind bei der Entwicklung eines Szenographiekonzepts auch die zur Darstellung erforderlichen Räume und Raumbeziehungen zu berücksichtigen. Ein Szenographiekonzept spannt hierfür ein virtuelles Raumarrangement Ra_{virt} (Gl. 3.8) auf, das den Bedarf gegenüber dem realen Raumarrangement Ra definiert. Die Eignung des virtuellen Raumarrangements kann auf ausgewählte Handlungstypen beschränkt werden. Damit auch kunstvolle Inszenierungen möglichen werden, sei jedoch nicht der Bedarf der darzustellenden Inhalte Voraussetzung, um mehrere Räume innerhalb eines Szenographiekonzepts nutzen zu können.

$$Ra_{virt} = (R_{virt}, E), E \subseteq R_{virt} \times R_{virt} \quad (3.8)$$

Wird ein Szenographiekonzept starr an eine bestimmte Raumanzahl oder Raumanordnung gebunden, so ist dessen Flexibilität hinsichtlich der möglichen Nutzung beschränkt. Bereits geringe, aber stete Änderungen des Informationsumfangs können während der Betriebsphase einer Anwendung dazu führen, dass Änderungen am Konzept erforderlich werden oder sich dieses nicht mehr für die Präsentation der Inhalte eignet. Die Situation ist vor allem dann unbefriedigend, wenn die steten Änderungen gewünscht sind und bereits zum Zeitpunkt der Entwicklung erwartet werden, jedoch kein Endpunkt der Weiterentwicklung erkennbar oder vorhanden ist. Sobald ein Konzept mehrfach genutzt wird, sind zudem direkte Änderungen an diesem nur noch sehr eingeschränkt möglich. Wird dennoch eine fortwährend gute Eignung angestrebt, resultiert dies rasch in einer großen Zahl ähnlicher Konzepte. In gleicher Weise wie beim Raumarrangement kann allerdings auch das virtuelle Raumarrangement eines Szenographiekonzepts durch den Einsatz von Mustern flexibilisiert werden. Hierbei sind jedoch neben den Räumen und deren Bindungen auch deren Eigenschaften zu berücksichtigen infolgedessen sich deren Eignung ergibt.

3.5.1 Entwicklungsdimensionen der Szenographie

Das Gewicht der Szenographie innerhalb der Gesamtentwicklung eines WIS hängt von den individuellen Zielsetzungen einer konkreten Entwicklung ab. Die Entwicklung eines Szenographiekonzepts kann durch eine Vielzahl von Aspekten beeinflusst werden, die auch gegenseitige Abhängigkeiten aufweisen. Exemplarisch seien hier einige Aspekte der Entwicklungsdimensionen genannt:

I Ambiente und Visualisierungsstil

F Interaktionskonzept und Erschließungskonzept

K Zielgruppe und Equipment

C Art und Umfang der Inhalte

S Handlungsform und Handlungsverlauf

P Gestaltungselemente und Gestaltungsmittel

Die Entwicklung des Szenographiekonzepts wird insbesondere durch die Dimension *Intention* beeinflusst, die ihrerseits an der Herausbildung der Dimension *Presentation* beteiligt ist (Abb. 3.3).

Allgemein besteht jedoch kein Zwang, dass die *Intention* Dimension Treiber der Entwicklung ist. Diese Flexibilität ist erforderlich, um spezielle künstlerische Anforderungen realisieren zu können, da eine kunstvolle Inszenierung nicht der pragmatisch geeigneten Inszenierung entsprechen muss. Ohnehin sind an der Herausbildung der *Intention* Dimension nicht selten weitere Entwicklungsdimensionen beteiligt, beispielsweise in-

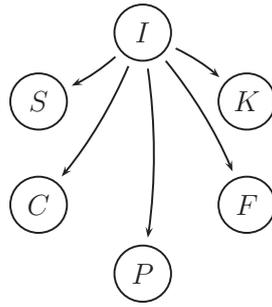


Abbildung 3.3: Szenographie Entwicklung

dem Aspekte bekannter Lösungen oder Kontextbedingungen bereits frühzeitig implizit Berücksichtigung finden (Abb. 3.4).

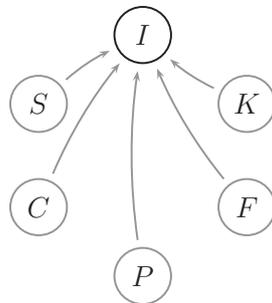


Abbildung 3.4: Szenographie Entwicklung - Einflüsse

Jede stärkere Konkretisierung der Ausgestaltungsregeln ist mit Einschränkungen der Verwendbarkeit des Konzepts verbunden. Oft betrifft dies Aspekte der *Presentation* Dimension und führt zu Beschränkungen hinsichtlich der Wahl des Präsentationsmediums. Eine geringe Abstraktion erlaubt somit einerseits das Realisieren spezieller Anforderungen beschränkt aber andererseits die Flexibilität und Adaptivität.

3.6 Dramaturgie des Gestaltungsraums

Das Erlangen der Aufmerksamkeit der Nutzer stellt eine initiale Voraussetzung für das Vermitteln von Informationen dar. Insbesondere im Falle umfangreicher Informationen besteht ferner die Notwendigkeit, ein hinreichend großes Interesse beim handelnden Akteur hervorzurufen. Dieses fortgesetzt während der gesamten Handlung zu erhalten, ist Aufgabe der Dramaturgie.

Eine *ganze Handlung* besitzt nach Aristoteles [14] einen *Anfang*, eine *Mitte* und ein *Ende*. Fehlt nur einer dieser Teile kann der für die Dramaturgie typische Spannungsbogen in der Regel nicht umgesetzt werden. Im Ergebnis beeinträchtigt dies die Aufmerksamkeit

und Bereitschaft, weitere Informationen aufzunehmen. Eine Ausnahme scheinen einige, modernere Handlungsformen darzustellen, die trotz fehlendem oder offenem Ende attraktiv und spannungsvoll wirken. Der Fokus solcher Handlungen liegt jedoch nicht auf der Gesamthandlung, sondern auf ihren Teilhandlungen, die ihrerseits den Regeln der *ganzen Handlung* entsprechen. Die Gesamthandlung ist in diesem Fall ein offener Rahmen, der Teilhandlungen organisiert und ggf. vorantreibt, nicht aber vervollständigt. Die Anwendbarkeit dramaturgischer Konzepte lässt sich daher allgemein durch nachfolgende Regeln prüfen.

Constraint 5. *Jeder Handlungsschritt einer Handlung muss zu jedem Zeitpunkt der Gesamt- oder einer ihrer Teilhandlungen zuordenbar sein, die im Aristoteles'schen Sinne als Ganz bezeichnet werden kann. Die dramaturgische Darbietung einer Handlung kann an jenen Stellen unterbrochen werden, die mindestens einer aktiven Teilhandlung zuordenbar sind, welche zum Zeitpunkt der Unterbrechung endet. Soll die Handlung nicht final unterbrochen, sondern zu einem späteren Zeitpunkt fortgeführt werden, muss ferner am Punkt der Unterbrechung mindestens eine neue Teilhandlung beginnen.*

Auch für WIS stellt die Dramaturgie eine Möglichkeit der kunstvollen Inszenierung dar und kann genutzt werden, um Akteure zu überraschen, zu leiten und zu informieren. Da der nötige Einfluss der Dramaturgie auf die Entwicklung stets von den individuellen Zielsetzungen der jeweiligen Anwendung abhängig ist, lässt er sich nicht allgemein festlegen.

Die Dramaturgie kann innerhalb eines WIS durch ein Dramaturgiekonzept beschrieben werden. Dessen Entwicklung wird insbesondere durch die *Story* Dimension geprägt, die ihrerseits die Entwicklung der anderen Dimensionen beeinflussen kann (Abb. 3.5).

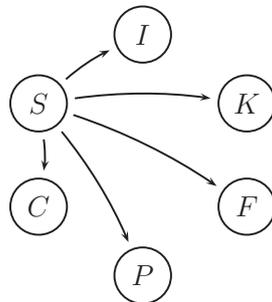


Abbildung 3.5: Dramaturgie Entwicklung

Ein Dramaturgiekonzept kann unabhängig von konkreten Handlungen und Räumen entwickelt werden, indem es sich nicht am individuellen Verlauf orientiert, sondern Handlungsmuster beschreibt. Da hieraus Platzbedarf entsteht, sind allgemeine Anforderungen hinsichtlich der Beschaffenheit geeigneter Gestaltungsräume zu definieren. In gleicher Weise wie für die Szenographie ist auch für die Dramaturgie die Definition eines virtuellen Handlungsraums Ra_{virt} (Gl. 3.8) möglich. Ebenfalls losgelöst vom realen Raumarrangement, innerhalb dessen schließlich die Präsentation der Handlung erfolgt, beschreibt

damit das Dramaturgiekonzept die Anordnung und Nutzbarkeit von Räumen aus dramaturgischer Sicht.

Die Eignung eines Dramaturgiekonzepts kann anhand des Bedarfs der jeweiligen Handlung und dem Potential eines konkreten Raumarrangements entschieden werden. Die grundsätzliche Anwendbarkeit dramaturgischer Konzepte ist daher, in Anlehnung an die obige Festlegung (Constr. 5), auch innerhalb von WIS überprüfbar (Def. 2).

Definition 2. Sei Sc eine Menge von Szenen und $S_{r_{full}}$ eine Menge vollständig terminierender Stories. Die Anwendbarkeit dramaturgischer Konzepte sei genau dann sichergestellt, wenn jede Szene des Handlungsraums $s_i \in Sc$ Teil einer echten, vollständig terminierenden Story ist und damit gilt: $\forall s_i \in Sc : s_i \in S_{r_{full}}$

Das Unterbrechen von Handlungen ist auch im Kontext von WIS möglich, erfordert jedoch erweiternd die Berücksichtigung des Interaktionsumfangs des Akteurs. Da der Akteur den Verlauf der Handlung typischerweise aktiv steuert, ihn jedoch oft nicht kennt, wird er, ohne hinreichende Unterstützung, die Handlung zu einem unbestimmten Zeitpunkt unterbrechen. Dies kann die Sicherung von Daten gefährden und den Wiedereinstieg in die Handlung erschweren. Durch das Definieren von Unterbrechungspunkten lässt sich ein kontrolliertes Pausieren erreichen, indem dem Akteur begleitende Informationen bereitgestellt werden, wann und unter welchen Umständen eine geeignete Unterbrechung möglich ist.

Lassen sich innerhalb einer Handlung Teilhandlungen bestimmen, die als *Ganz* im Aristoteles'schen Sinne gelten, so können diese genutzt werden, um geeignete Unterbrechungspunkte Ξ zu ermitteln (Gl. 3.9). Allgemein gelte, dass der Abschluss jeder Teilhandlung eine Unterbrechung erlaubt, wenn zugleich sichergestellt ist, dass ausgehend von diesem Punkt eine neue Teilhandlung beginnt oder aber die Gesamthandlung endet (Gl. 3.10).

$$S_{sub} \subseteq S_{r_{full}} \quad (3.9)$$

Sei $s_a \in S_i, s_b \in S_j$ mit $s_a \prec s_b$ und $s_a \notin S_j, s_b \notin S_i$ sodass gilt:

$$\xi(s_a, s_b) \in \Xi : \begin{cases} \xi_1(s_a, s_b), & S_i, S_j \neq \emptyset \\ \xi_2(s_a, \perp), & S_i \neq \emptyset, S_j = \emptyset \\ \perp, & \text{sonst} \end{cases} \quad (3.10)$$

Entsprechend kann die Beispielhandlung S (Abb. 3.6) nur nach Abschluss der Teilhandlung S_1 unterbrochen werden als auch final nach Beendigung von S_2 und S_4 bzw. mit der vollständigen Beendigung der Gesamthandlung S .

Lösung: Das Problem des ungewollten Handlungswechsels besteht nicht, wenn nach dem Einstieg in eine Handlung überschneidende Alternativhandlungen grundsätzlich unberücksichtigt bleiben. Diese restriktive Vorgehensweise verhindert jedoch auch dann den Wechsel, wenn während der Nutzung eine dieser Handlungen als geeigneter erkannt wird. Alternativ kann durch Anpassen der Darstellung eine sichere Interaktion ohne Beschränkung gewährleistet werden. Im einfachsten Fall geschieht dies durch Separation jener Pfade, die nicht der bisherigen Handlung zugeordnet werden können. Die Integration ist möglich, wenn zugleich empfohlene Wege der aktuellen Handlung gegenüber den Alternativen hervorgehoben werden können.

Offen: In welcher Form erwartet der Akteur die Darbietung von Informationen?

Handlungsform:

Die Erwartungen der Akteure reichen von einer ergebnis-orientierten Darbietung der Inhalte bis hin zu Lernumgebungen, die Wiederholen und das Überprüfen von Erlerntem erlauben.

Problem: Neben der Wahl eines geeigneten Story-Typs ist die angemessene Detaillierung der Inhalte zu entscheiden.

Lösung: In Abhängigkeit von den Bedürfnissen des Akteurs kann dieser direkt oder schrittweise zum Ergebnis geführt werden.

Offen: Wie lassen sich Seiteneinstiege behandeln?

Einstiegspunkt:

Typischerweise existieren mehrere Einstiegspunkte in eine Handlung, wobei zwischen geplanten als auch ungeplanten Einstiegen unterschieden werden kann.

Problem: Eine Herausforderung stellt der ungeplante Einstieg in die Handlung dar, der unter anderem durch Suchmaschinen-Einträge verursacht werden kann. Aufgrund des hohen Spezifikationsaufwands kann zumeist nicht für jeden potentiellen Einstiegspunkt eine eigene Handlung definiert werden. Daher gilt es zu entscheiden, wie im Falle eines solchen Einstiegs in die Handlung zu verfahren ist.

Lösung: Eine einfache Lösung besteht in der automatischen Weiterleitung (*Redirect*), die den Akteur stets zu dem geplanten Einstiegspunkt in die Handlung führt. Problematisch ist jedoch der Einstieg an Überschneidungspunkten von Handlungen, da für diesen Fall die Weiterleitung nicht ohne weitere Maßnahmen automatisiert erfolgen kann. Ist der Akteur lediglich an einem Resultat interessiert ist, erschwert ihm eine solche Weiterleitung sogar den Weg zur Information.

Der partielle Verzicht auf Weiterleitungen ist möglich, wenn der Verlauf von Handlungen dynamisch angepasst werden kann (*Reordering*). Voraussetzung hierfür sind Kenntnisse über die Beziehungen der Handlungsschritte einer Handlung. Diese lassen sich nutzen, um Beschränkungen zu bestimmen bzw. die Menge aller Sequenzen von Handlungsschritten, die eine festgelegte Ausführungsreihenfolge erfordern.

Offen: Insbesondere durch die dynamische Adaption des Handlungsverlaufs kann nicht ermittelt werden, ob eine Handlung dem Akteur vollständig präsentiert wurde.

Entwicklung:

Durch die gewünschte Flexibilität sind alternative Handlungen für das vollständige Lösen von Aufgaben einer Anwendung möglich.

Problem: Durch die Existenz alternativer Pfade ist nicht sichergestellt, ob und wann Handlungen vollständig abgeschlossen und Aufgaben erfüllt werden.

Lösung: In [64] wurde das Verlassen einer Szene erstmals an Bedingungen geknüpft, wodurch überprüfbar wurde, welche Teilaufgaben ein Akteur bereits erfüllt hat. Entsprechend gelten die Bedingungen eines Handlungsschritts im Kontext der Dramaturgie für jeden zulässigen Verlauf der Handlung. Die Menge der Bedingungen beschreibt somit das Minimum, das ein Akteur erfüllen muss, um die Handlung beenden zu können.

Offen: Beschränkungen des zeitlichen Verlaufs der Handlung bleiben bislang unberücksichtigt.

Zeit:

Das erfolgreiche Beenden einer Aufgabe unterliegt mitunter zeitlichen Beschränkungen, z.B. um die Sicherheit oder Synchronität mit anderen Inhalten zu gewährleisten.

Problem: Eine zu langsame und auch die zu schnelle Bearbeitung kann einen Abbruch einschließlich *Rollback* provozieren, sodass eine erneute teilweise oder vollständige Bearbeitung der Aufgabe erforderlich wird.

Lösung: Die Definition von Art und Umfang erlauben in Verbindung mit einer Markierung des betroffenen Handlungsabschnitts, den Akteur angemessen über die Beschränkung zu informieren. Hierbei gilt es mehrere Informationszeitpunkte und -ziele zu unterscheiden:

- Vor: dem Eintritt in den beschränkten Bereich wird der Akteur auf die anstehende Beschränkung als auch deren Bedingungen und ggf. Gründe hingewiesen.
- Während: des Aufenthalts im beschränkten Bereich ist der Akteur über die verbleibende Zeit oder die bisherige Verweildauer zu informieren. Dies ist nicht nötig, wenn zwar die Zeit zur Interaktion beschränkt ist, der Fortgang der Handlung jedoch nicht von der Interaktion des Akteurs abhängt.
- Nach: dem Verlassen des beschränkten Bereichs ist der Akteur über den Erfolg oder Misserfolg der Bearbeitung zu informieren.

Offen: Die Bearbeitung komplexer und zeitaufwendiger Aufgaben weist indirekte Zeitbeschränkungen auf, die nicht durch die Handlung, sondern den Akteur verursacht werden.

Unterbrechung:

Die Definition von Szenen und Handlungen ist hinsichtlich deren Länge und Komplexität nicht beschränkt.

Problem: Aufgrund der unbeschränkten Länge und Komplexität können Unterbrechungen nötig werden. Insbesondere für komplexe und umfangreiche Handlungen kann der Aufwand des manuellen Zuordnens geeigneter Unterbrechungspunkte zu hoch sein.

Lösung: Handlungsabschnitte mit direkten Zeitbeschränkungen sind weiterhin nicht unterbrechbar.

Geeignete Unterbrechungspunkte können durch das Ermitteln von Teilhandlungen gefunden werden. Hierfür kann in einem ersten Schritt nach zusammenhängenden Sequenzen von Handlungsschritten (Einstiegspunkt) gesucht werden. Mit der Beendigung jeder Sequenz ist eine Unterbrechung möglich, da durch aufgrund der Flexibilität des Verlaufs die enge inhaltliche Bindung nicht mehr vorausgesetzt werden kann. Daher ist auch nach Beendigung jedes Handlungsschritts eine Unterbrechung möglich, wenn dieser eigenständig und damit kein Teil einer Sequenz ist. Weitere geeignete Unterbrechungspunkte lassen sich innerhalb der Sequenzen finden. Hierfür kann die Menge der Bedingungen, die ein Akteur erfüllen muss, um die Handlung beenden zu können (Entwicklung) berücksichtigt werden. Mit dem Erfüllen der Bedingungen einer Szene ist auch eine Unterbrechung möglich, da dies einen inhaltlichen Abschluss darstellt und auch derart vom Akteur wahrgenommen werden kann.

3.6.2 Story-Typen

Die Struktur eines jeden Szenengraphs beeinflusst den möglichen Verlauf der Handlung und damit deren Wirkung und Visualisierungspotential. Folgend werden relevante Story-Typen unterschieden, die auch die bestehenden dramaturgischen Besonderheiten der Entwicklung von WIS berücksichtigen. Die Visualisierung erfolgt auf der Basis grundlegender Story-Elemente (Abb. 3.7) und wird schrittweise erweitert. Da die gewählte Notation von Story-Typen eine schwache Abstraktion erlaubt, die lediglich zulässige sowie erforderliche Transitionsbedingungen und Szenenelemente umfasst, kann auch die Evolution von Stories eines definierten Typs beschränkt und gesteuert werden. Andererseits lassen sich mögliche Problemstellen einer Entwicklung frühzeitig erkennen, vermeiden und beheben. Story-Typen können auch statische und damit starre Teilverläufe oder Gesamthandlungen beschreiben, wenngleich Letzteres nicht der Intention von Story-Typen gerecht wird.

Neben der visuellen Notation wird eine einfache, semi-formale Darstellung eingeführt, die sich insbesondere für die Spezifikation von Szenario eignet, jedoch nicht dahingehend beschränkt ist. Mit zunehmender Zahl und Länge alternativer Handlungspfade sinkt jedoch deren Übersichtlichkeit. Bei dieser Darstellungsform wird der Beginn einer Story mit dem Symbol \nearrow und das Ende mit dem Symbol \searrow gekennzeichnet. Die dem Story-Beginn nachgestellte Szene Sc ist die Startszene, während die dem Story-Ende vorangestellte Szene als Endszene fungiert. Durch die auch hier bestehende Trennung einer Szene von ihren Aufgaben wird diese kontextabhängig mehrfach verwendbar.

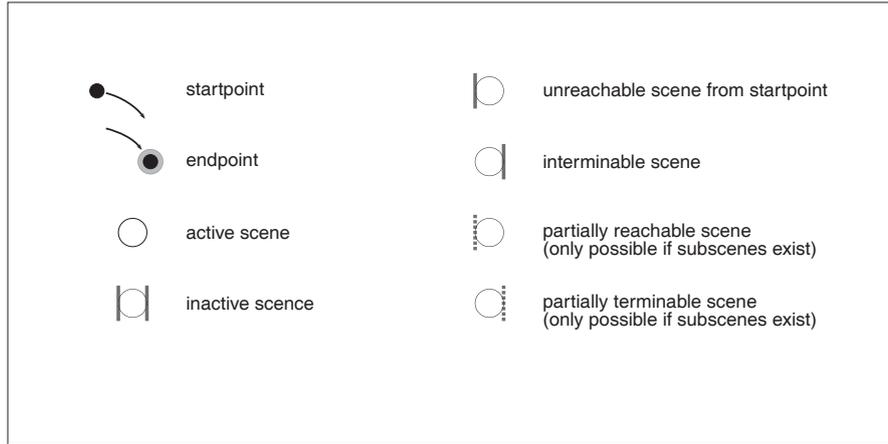


Abbildung 3.7: Story-Elemente

- S - Story
- \nearrow - Story-Beginn
- \searrow - Story-Ende
- Sc - Szene einer Story

Für viele Handlungen als auch zum Erhalt der Übersichtlichkeit der Notation kann es von Vorteil sein, Rückschritte stets implizit zu erlauben. Für diesen Fall ist jedoch die gewünschte Semantik solcher Rückschritte zu definieren, da Transitionen hin zu vormaligen Szenen als echtes Rollback verstanden werden können oder aber innerhalb einer Session erlangte Informationen durch einen Rückschritt nicht verloren gehen sollen. Ferner bergen Rückschritte Konfliktpotential, sobald eine Szene Daten einer vorangegangenen Szene ändert oder weiterverarbeitet und die Ursprungsdaten nicht mehr vorhanden oder rekonstruierbar sind.

Vollständige Story:

Jede *vollständige Story* S_c (Gl. 3.11) besitzt genau eine Startszene und mindestens einen Endszene, die ausgehend von der Startszene über mindestens einen Handlungspfad erreichbar ist.

S_c - vollständige Story (complete)

$$\begin{aligned}
 &\forall S_c : \exists Sc_{start} \wedge Sc_{end}, Sc \in S_c \\
 &if(Sc_{start} \neq Sc_{end}) \\
 &then \exists T_{Sc_{start}, Sc_{end}, constraints}(S_c) : Sc_{start} \rightarrow \dots \rightarrow Sc_{end}
 \end{aligned} \tag{3.11}$$

Semiformal lässt sich eine einfache, ausschließlich sequentiell ablaufende, vollständige Story $S_{c_{seq}}$ mit dem folgenden Term (Gl. 3.12) beschreiben.

$$S_{c_{seq}} = \nearrow SC_1 \rightarrow \dots \rightarrow SC_n \searrow \quad (3.12)$$

Eine minimale, vollständige Story $S_{c_{min}}$ liegt vor, wenn Startszene und Endszene identisch sind, wobei in diesem Fall echtes Handeln nicht möglich ist. Die Notwendigkeit einer solch starren Präsentation ist für manche Anwendungen dennoch gegeben. Aufgrund der sehr geringen Beschränkungen erlaubt der Story-Typ S_c neben Sequenzen $S_{c_{seq}}$ auch Zyklen $S_{c_{cyclic}}$. Ein Beispiel für die minimale, eine sequentielle und eine zyklische Story S_c findet sich in Abbildung 3.8.

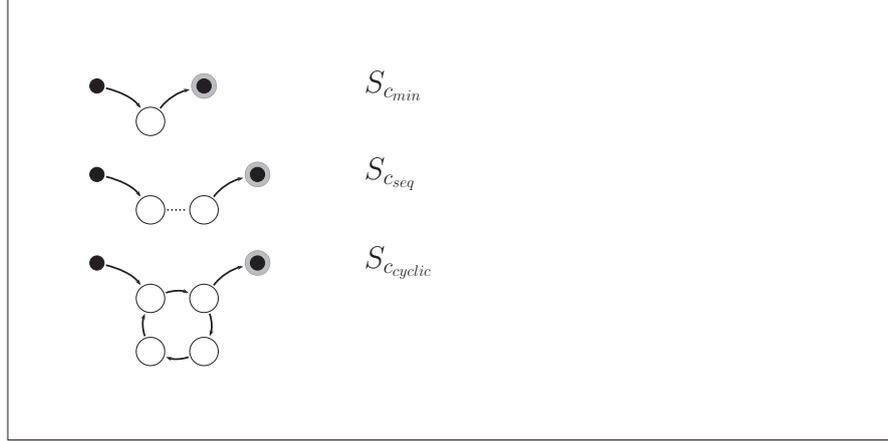


Abbildung 3.8: Vollständige Story

Unvollständige Story:

Bei der Spezifikation einer Story wird mitunter auf die Angabe einer Endszene verzichtet, um eine Handlung nachträglich weiterentwickeln bzw. ergänzen zu können. In einigen Fällen ist ein finaler Evolutionsschritt zu einem späteren Zeitpunkt sogar ganz bewusst nicht vorgesehen. Stories ohne Endszene erlauben das Spezifizieren von Handlungen ähnlich dem *soap opera* Ansatz, aber auch das Erzeugen unendlicher Zyklen. Ist keine Endszene vorhanden oder ist keine vorhandene erreichbar, wird die Story als *unvollständige Story* S_i (Gl. 3.13) bezeichnet.

S_i - unvollständige Story (incomplete)

$$\begin{aligned} \forall S_i : \exists SC_{start}, SC \in S_i \\ \text{if}(\exists SC_{end}) \\ \text{then } SC_{start} \neq SC_{end} \wedge \nexists T_{SC_{start}, SC_{end}, constraints}(S_i) : SC_{start} \rightarrow \dots \rightarrow SC_{end} \end{aligned} \quad (3.13)$$

Entsprechend der obigen Bedingungen ist eine sequentielle, unvollständige Story $S_{i_{seq}}$ (Gl. 3.14) ausführbar, kann jedoch nicht terminieren, da keine Szene der Sequenz als Endszene benannt wird und ihr damit das Story-Ende (\searrow) fehlt.

$$S_{i_{seq}} = \nearrow SC_1 \rightarrow \dots \rightarrow SC_n \quad (3.14)$$

Eine minimale, unvollständige Story $S_{i_{min}}$ liegt vor, wenn lediglich eine Szene existiert, die als Startszene markiert wurde. Da die Szene in diesem Fall nicht zugleich Endszene sein darf, kann die Story nicht terminieren und nur durch vollständigen Abbruch der Applikation verlassen werden. In gleicher Weise wie bei den vollständigen Stories sind auch bei unvollständigen Stories Sequenzen $S_{i_{seq}}$ und Zyklen $S_{i_{cyclic}}$ beschreibbar. Ein Beispiel für die minimale, eine sequentielle und eine zyklische Story S_i findet sich in Abbildung 3.9.

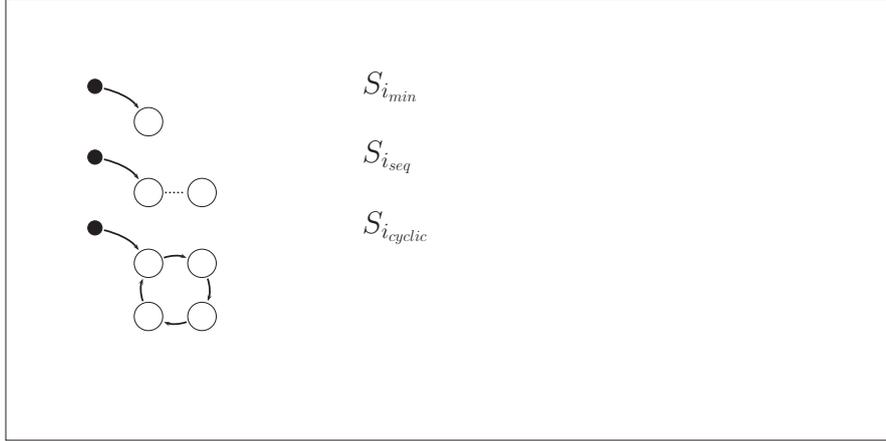


Abbildung 3.9: Unvollständige Story

Echte Story:

Jede *echte Story* S_r (Gl. 3.15) erlaubt den Akteuren der Anwendung echtes Handeln. Eine echte Story besitzt daher mindestens eine Transition, die ausgehend von der Startszene zu einer von ihr verschiedenen Szene führt. Die Transitionsbedingung einer echten Story S_r kann dabei bereits durch Teilszenen erfüllt werden.

S_r - echte Story (real)

$$\begin{aligned} \forall S_r : \exists T_{S_{c_1}, S_{c_2}, constraints}(S_r) : S_{c_1} \rightarrow S_{c_2}, \\ S_{c_1}, S_{c_2} \in S_r, S_{c_1} \neq S_{c_2} \end{aligned} \quad (3.15)$$

Die minimale, echte Story $S_{r_{min}}$ (Gl. 3.16) terminiert nicht und genügt daher lediglich den Anforderungen einer unvollständigen Story S_i , aufgrund der erforderlichen Transition, jedoch nicht denen der minimalen, unvollständigen Story $S_{i_{min}}$.

$$S_{r_{min}} = \nearrow S_{c_1} \rightarrow S_{c_2} \quad (3.16)$$

Beispielhaft werden in Abbildung 3.10 eine nicht echte Story $\neg S_r$ als auch zwei minimale echte Stories $S_{r_{min}}$ gezeigt. Um keine echte Story $\neg S_r$ handelt es sich, wenn nur eine einzige Szene vorhanden ist, aber auch dann, wenn keine Transition zu einer von der Startszene verschiedenen Szene führt. Eine minimale, echte und vollständige Story

$(S_{r_{min-complete}})$ liegt vor, sobald $S_{r_{min}}$ um die Möglichkeit des Terminierens erweitert wird. Das reale Handeln des Akteurs ist dabei nicht verpflichtend, weshalb trotz einer gültigen Transition die Startszene zugleich Endszene der Story sein darf.

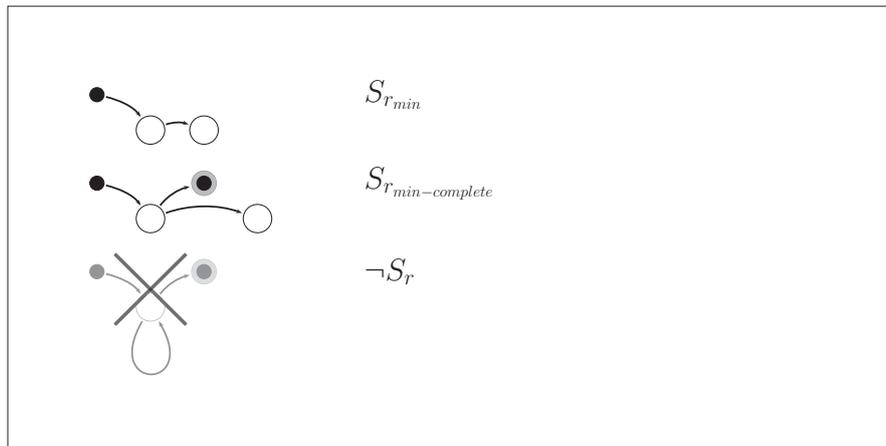


Abbildung 3.10: Echte Story

Personalisierbare Story:

Durch die in dieser Arbeit verwendete Erweiterung des Storybegriffs um Handlungsoptionen, beschreibt eine Story nicht mehr zwingend eine einzige, festgelegte Handlung. Um diese Spezifikationsoption nicht zu verlieren, können Szenario [65] genutzt werden, die stellvertretend einen konkreten Weg durch die Story beschreiben. Szenario helfen dabei, typische Lebensfälle [56] zu beschreiben, um Nutzer individuell und situationsgerecht durch eine Anwendung führen zu können. Verschiedene Wege durch die Story sind notwendig, weil nicht alle Nutzer in gleicher Weise agieren und ihr Profil und Portfolio sich stark unterscheiden kann. Sollte ein Nutzer während der Benutzung seine Strategie ändern, so muss die Möglichkeit bestehen, auch während der Benutzung das Szenario zu wechseln, um der veränderten Situation besser gerecht werden zu können. Gezielte Reaktionen auf Veränderungen sind unter anderem durch die Analyse des gewählten Pfades des Nutzers möglich, indem Abweichungen vom Standard-Pfad analysiert und andere Szenario-Typen auf ihre Eignung überprüft werden.

Handlungsoptionen sind Bestandteil einer *personalisierbare Story* S_p und erlauben dem Akteur mindestens an einem Punkt, den Verlauf der Handlung zu beeinflussen. Eine Entscheidung des Akteurs und damit die Realisierung eines individuellen Handlungsverlaufs ist möglich, sobald die Handlung mindestens eine erreichbare Szene mit mehr als einer ausgehenden oder eingehenden Transition besitzt (Gl. 3.17). Die Nutzbarkeit eines Übergangs von einer Quell- zur Zielszene kann an Bedingungen geknüpft sein, z.B. an die Abhängigkeit von den Rechten und Aufgaben des Nutzers, seinen Angaben in der Quell-szene sowie den bisherigen Handlungsverlauf. Ferner regeln die Transitionsbedingungen welche Daten unter welchen Bedingungen von der Quell-szene an die Zielszene weitergegeben werden dürfen. Folglich kann es sinnvoll sein, zwischen zwei Szenen mehrere

Transitionen gleicher Richtung zuzulassen. Zur Vermeidung von Redundanzen, ist es jedoch dann sinnvoll, die Bedingungen für all diese Transitionen eindeutig unterscheidbar zu halten.

S_p - personalisierbare Story (personalise)

$$\forall S_p : \exists T_1, T_2(S_p), T_1 \neq T_2$$

mit

$$T_{1;S_{c_q},S_{c_r},constr_u}(S_p)$$

$$T_{2;S_{c_s},S_{c_t},constr_v}(S_p)$$

$$((S_{c_q} = S_{c_s}) \oplus (S_{c_r} = S_{c_t})) \vee ((S_{c_q} = S_{c_s}) \wedge (S_{c_r} = S_{c_t}) \wedge (constr_u \neq constr_v)) \quad (3.17)$$

Eine einfache Beispielhandlung $S_{p_{split-and-join}}$ erfordert entsprechend des folgenden semiformalen Terms (Gl. 3.18) an der Stelle S_{c_1} eine Entscheidung des Akteurs zugunsten des Pfades S_{c_2} oder S_{c_3} . Die Wahl beider Pfade sei hierbei nicht möglich.

$$S_{p_{split-and-join}} = \nearrow S_{c_1} \rightarrow S_{c_2} \rightarrow \dots \rightarrow S_{c_n} \searrow \oplus \nearrow S_{c_1} \rightarrow S_{c_3} \rightarrow \dots \rightarrow S_{c_n} \searrow \quad (3.18)$$

Eine minimale personalisierbare Story $S_{p_{min}}$ ist nicht *echt* und nicht *vollständig* und umfasst daher nur eine Szene, die als Startszene, jedoch nicht als Endszene fungiert. Die für diesen Fall minimal erforderlichen Transitionen werden im folgenden Beispiel (Abb. 3.11) als T_1 und T_2 bezeichnet. Typischerweise werden alternative Handlungspfade für die Flexibilisierung des Verlaufs und Abweichungen im Detail verwendet, infolgedessen die selbe Endszene erreicht werden kann. Dies gilt beispielsweise für Story $S_{p_{split-and-join}}$ (Abb. 3.11). Ein diesbezüglicher Zwang existiert jedoch nicht, sodass durch die Spezifikation alternativer Handlungspfade auch mehrere Endszene erforderlich werden können. Ein Beispiel hierfür ist die Story $S_{p_{split-only}}$ (Abb. 3.11).

Terminierende Story:

Eine *terminierende Story* S_t gibt darüber Auskunft, ob und in welchem Umfang eine Story terminieren kann. Im eigentlichen Sinne repräsentiert sie daher keinen eigenen Story-Typ, sondern stellt vielmehr eine Verfeinerung der Unterscheidung vollständiger und unvollständiger Stories dar. Zweck dieser Unterscheidung ist das leichtere Erkennen von Deadlocks und unendlichen Zyklen.

Bereits eine einfache minimale, echte und vollständige Story $S_{r_{min-complete}}$ wie in Abbildung 3.10 offenbart eine Klassifikationsproblematik, da die dargestellte Startszene stets terminieren kann, die sich ihr anschließende jedoch keinesfalls. Für personalisierbare Stories kommt erschwerend hinzu, dass sie mehrere erreichbare Endszene erlauben, die über einfache, unterbrechbare Handlungssequenzen hinausgehen. Immerhin kann in beiden Fällen die Möglichkeit des Terminierens einheitlich für die Story bzw. den Story-Beginn entschieden werden. Dies ist erforderlich, um allgemein die Nutzbarkeit von Handlungen bewerten zu können.

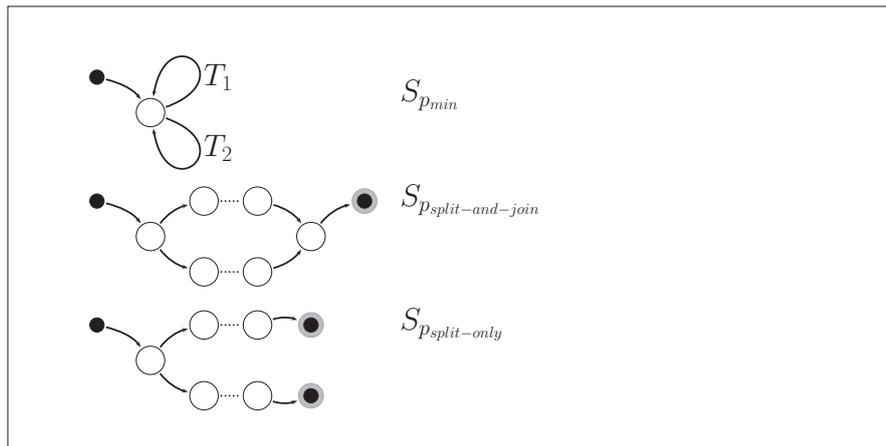


Abbildung 3.11: Personalisierbare Story

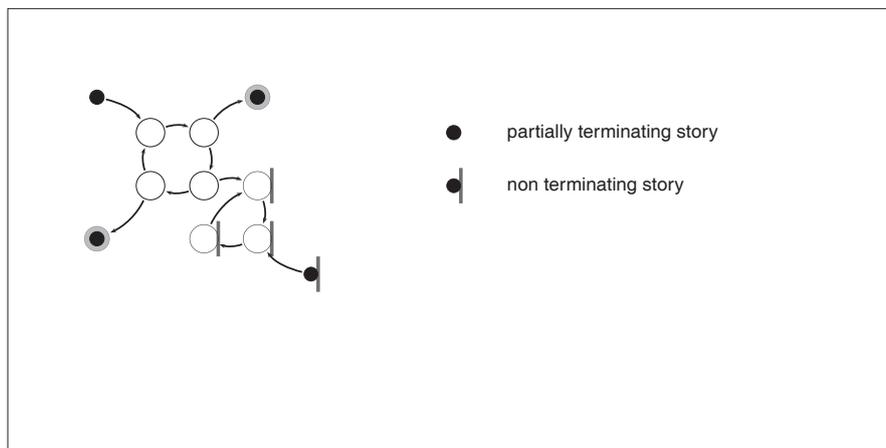


Abbildung 3.12: Nicht und partiell terminierende Story

Wird ein flexibler Einstieg in die Handlung gestattet, ist die Möglichkeit des Terminierens für jeden *Story-Beginn* und damit jede Teilhandlung gleicher Ebene separat entscheidbar. Mitunter beschränkt dieser Umstand die mögliche Ausführungsreihenfolge von Teilhandlungen, weshalb folgend unterschieden wird, ob die Handlung ausgehend von einem *Story-Beginn* *vollständig*, *partiell* oder *nicht* terminieren kann. Auch die grafische Notation greift diese Unterscheidung auf und kennzeichnet entsprechend den *Story-Beginn* von Handlungen, wenn diese nicht terminieren können (Abb. 3.12).

Kann, ausgehend von einem gewählten *Story-Beginn*, stets eine Endszene erreicht werden, so handelt es sich um einen vollständig terminierenden *Story-Beginn*. Auch diese Situation hebt die grafische Notation gesondert hervor (Abb. 3.13), da in diesem Fall den Akteuren immer ein sicherer Ausstieg aus der Handlung möglich ist.

Wenngleich dies für einfache Sequenzen attraktiv erscheint, kann die letzte Szene einer Sequenz nicht automatisch Endszene sein, da andernfalls eine bewusste und wirkungsvolle Platzierung, unter Beachtung der Ziele der Dramaturgie, hinfällig wäre. Ferner

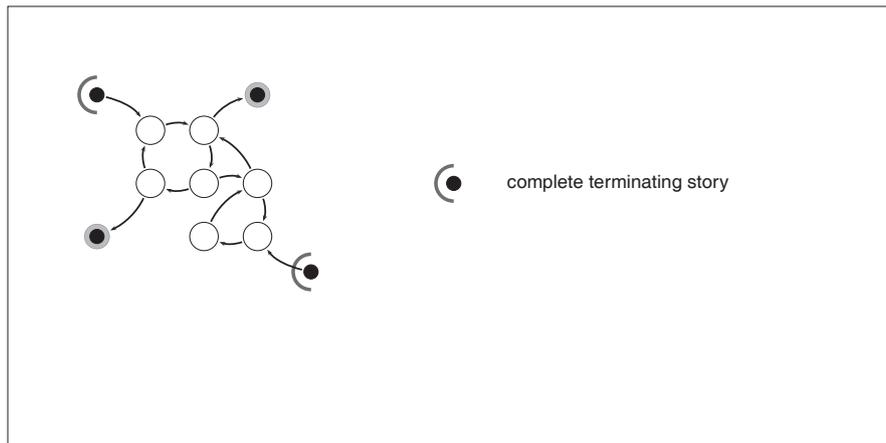


Abbildung 3.13: Vollständig terminierende Story

bestünde für Zyklen ein Entscheidungsproblem und die Existenz und nötige Platzierung von Endszenen würde von der Reihenfolge von Änderungen abhängig.

3.6.3 Szenen-Eigenschaften

Mit der Einführung der obigen Story-Typen entsteht auch für die Notation von Szenen ein weiterer Präzisierungsbedarf. Dies ist einerseits notwendig, um die Behandlung von Teilhandlungen entscheiden zu können, und andererseits, um das Auffinden möglicher Problemstellen zu erleichtern.

Aktivität:

Jede *aktive Szene* einer Story muss Teil des Story-Graphen sein und ist daher wenigstens durch eine Kante mit diesem verbunden. Um den Forderungen der Flexibilität und Adaptivität zu genügen, lassen sich Bindungen zwischen Szenen nachträglich ändern und lösen. Infolgedessen kann für jede Szene einer Story nicht stets vorausgesetzt werden, dass diese *aktiv* im obigen Sinne ist. Besitzt eine Szene Teilszenen und sind weder keine noch alle Teilszenen mit dem Story-Graphen verbunden, so wird die Szene als *partiell aktive Szene* bezeichnet.

Erreichbarkeit und Reichweite:

Da die Bindungen zwischen Szenen als gerichtet betrachtet werden und infolge der Flexibilität der Akteure, den Verlauf der Handlung zu beeinflussen, kann global die Erreichbarkeit einzelner Szenen sowie lokal die Erreichbarkeit des Story-Endes beschränkt sein. Bestehende Beschränkungen sind nicht immer hinreichend gut erkennbar weshalb folgend Einstiegs- und Ausstiegsbeschränkungen von Stories gesondert gekennzeichnet werden.

Jede *Szene* einer Story muss wenigstens durch einen Handlungsverlauf erreichbar sein, um als verwendbarer Bestandteil der Story zu gelten. Eine Szene ist unerreichbar (*unreachable scene*), wenn sie ausgehend vom Story-Beginn der Handlung nicht erreichbar ist

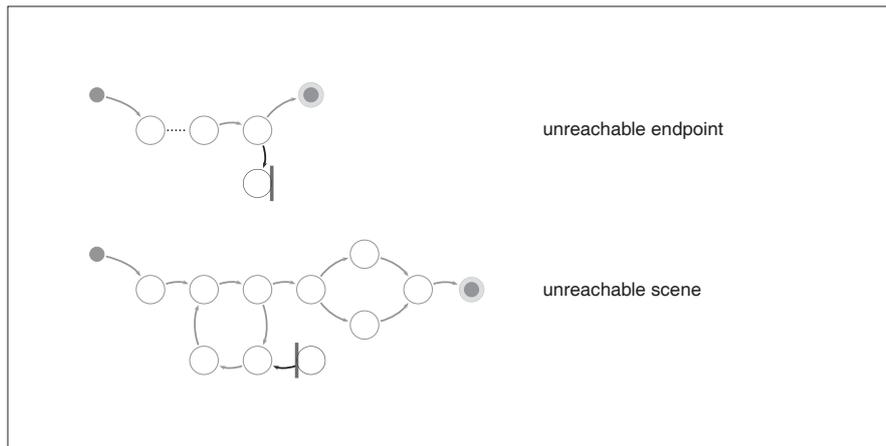


Abbildung 3.14: Unerreichbare Szene / unerreichbares Story-Ende

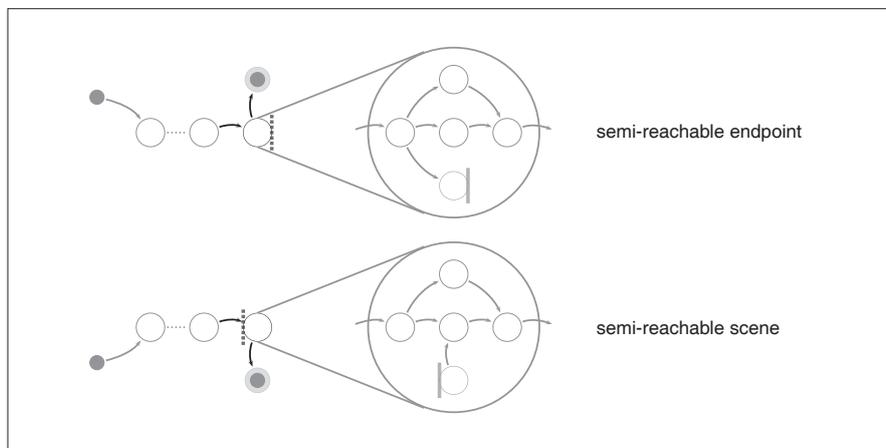


Abbildung 3.15: Partiiell erreichbare Szene / partiiell erreichbares Story-Ende

(Abb. 3.14). Analog kann die Handlung nicht terminieren (*unreachable endpoint*), wenn ausgehend von einer ausgewählten Szene das Story-Ende nicht erreichbar ist (Abb. 3.14).

Partiiell-erreichbar (*semi-reachable scene*) ist eine Szene, wenn sie über Teilszenen verfügt, die nur teilweise, vom Story-Beginn ausgehend, erreichbar sind (Abb. 3.15). Analog ist das Story-Ende partiiell erreichbar (*semi-reachable endpoint*), wenn die Erreichbarkeit weder für alle noch für keine der Teilszenen gilt (Abb. 3.15).

Im Unterschied zum urtypischen Gebrauch von Szenen, ist im Bereich von WIS auch deren mögliche Situationsabhängigkeit zu berücksichtigen, die durch den flexiblen Einstieg in die Handlung als auch Ausstieg aus dieser entstehen kann. Existieren mehrere Einstiege in die Handlung als auch Ausstiege aus der Handlung, kann ergänzend annotiert werden, von wie vielen Startpunkten ausgehend eine Szene erreichbar ist und wie viele Endpunkte von der Szene ausgehend erreichbar sind (Abb. 3.16). Einerseits erleichtert dies das Erkennen allgemeiner Beschränkungen der Erreichbarkeit bereits zum Zeitpunkt der Anbindung einer Szene an den Story-Graphen. Andererseits lassen

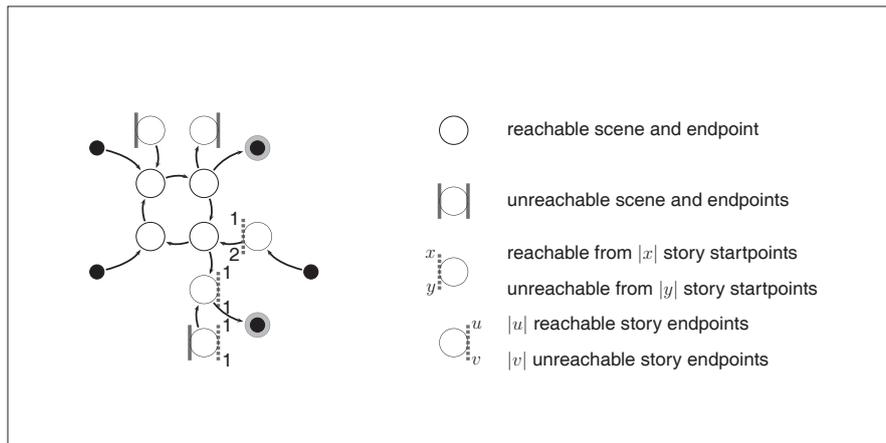


Abbildung 3.16: Szenenbehandlung bei mehreren Start- und Endscenen

sich auf diese Weise bei Änderungen eines bestehenden Story-Graphen auch entstandene Änderungen der Erreichbarkeit und damit die Konsequenzen für einzelne Stories aufzeigen.

3.7 Screenography im Detail

Die Entwicklung der Darstellung bzw. der damit verbundene Entwicklungsprozess wird von verschiedensten Rand- und Vorbedingungen, subjektiven Präferenzen des Entwicklers und Auftraggebers als auch den Erwartungen und Bedürfnissen der späteren Nutzer beeinflusst. Von der Art der zu entwickelnden Anwendung hängt beispielsweise ab, ob und wann Abhängigkeiten zwischen den Einflussfaktoren der Entwicklungsdimensionen existieren und ob diese Bindungen gerichtet sind. Auch die Berücksichtigung der individuellen Bedeutung eines Faktors, und damit dessen Gewicht für den Entwicklungsprozess, beeinflusst das spätere Ergebnis.

Eine allgemeingültige und insbesondere wiederverwendbare Lösung mit hinreichender Flexibilität kann nicht gefunden werden, wenn alle Einflussgrößen von Beginn an gelten. Um diese Starre zu überwinden, wird ein schrittweiser Prozess vorgeschlagen, innerhalb dessen die Parameter und deren Beziehungen zu ermitteln sind. Dabei werden die im Sinne der Screenography eingeführten Konzepte verwendet und Konzepte aus Szenographie und Dramaturgie systematisch mit Techniken der Entwicklung von Webseiten verbunden.

Zunächst sind geeignete Website-Typen zu suchen, welche die Ziele und Anforderungen der Darstellung im Kleinen und jene der Handlung im Großen bestmöglich erfüllen. Hierfür sind die obligatorischen und relevanten Parameter jeder WIS-Dimensionen zu fixieren (Gl. 3.19-3.24).

$$Intention(p_1^I, \dots, p_{k_I}^I) \tag{3.19}$$

$$Story(p_1^S, \dots, p_{k_S}^S) \tag{3.20}$$

$$\text{Context}(p_1^K, \dots, p_{k_K}^K) \quad (3.21)$$

$$\text{Content}(p_1^C, \dots, p_{k_C}^C) \quad (3.22)$$

$$\text{Functionality}(p_1^F, \dots, p_{k_F}^F) \quad (3.23)$$

$$\text{Presentation}(p_1^P, \dots, p_{k_P}^P) \quad (3.24)$$

Allgemein werden damit die nennenswerten Einflussgrößen aller WIS-Dimensionen berücksichtigt (Gl. 3.25). Indes besteht hingegen keine Notwendigkeit, tatsächlich jede WIS-Dimension zu berücksichtigen.

$$\text{Dimension}(p_1^{D_i}, \dots, p_{k_{D_i}}^{D_i}) \quad (3.25)$$

Ohnehin passen eine ähnliche Motivation und ähnliche Ziele eines Gestaltungsvorhabens nur selten vollumfänglich zur Spezifikation eines einzelnen Website-Typs. Durch Gewichtung der Einflussgrößen und schrittweise Verfeinerung kann aber eine Reduktion möglicher Optionen erreicht werden. Eine eindeutige Abgrenzung aller Website-Typen anhand der Definition der Anforderungen ist wegen des hohen Abstraktionsgrades dieses Entwicklungsschritts nur bei starker Beschränkung der Optionen möglich oder mündet in einer Vielzahl schwer unterscheidbarer, ähnlicher Website-Typen.

Die Szenographie-Sicht kann einen konkreten Website-Typ vollständig repräsentieren. Die Szenographie definiert dann ein Mockup einer geplanten Inszenierung. Aufgrund ihrer Aufgabe und Stellung innerhalb der Entwicklung, muss das mögliche Zusammenwirken der Dimensionen und deren Faktoren nicht zwingend konfliktfrei konzipiert sein. Aufgrund existierender Wechselwirkungen zwischen einzelnen Faktoren der Entwicklungsdimensionen ist der tatsächliche Einfluss eines Faktors neben seinem Gewicht auch vom Zeitpunkt seiner Berücksichtigung abhängig.

Die Auflösung existierender Konflikte ist unabdinglich bzw. Voraussetzung dieser Vorgehensweise. Entsprechend der definierten Präferenzen und Abhängigkeiten zwischen den Parametern kann eine schrittweise Annäherung an eine realisierbare Lösung erfolgen. Dabei werden die nur jene Abhängigkeiten zwischen Parametern berücksichtigt, die für die gewählte Szenographie-Sicht von Bedeutung sind (Gl. 3.26).

$$p_{k_{D_i}}^{D_i} \mapsto p_{l_{D_j}}^{D_j} \quad (3.26)$$

Die Auflösung von Konflikten lässt sich oft nur bedingt automatisieren, aufgrund der großen Zahl kombinatorischer Lösungen, der möglichen Widersprüche in der Definition und des Vorhandenseins gegensätzlicher Abhängigkeiten. Eine automatisierte Auflösung von Konflikten ist für gleichartige Problemstellungen möglich, wenn eine starke Beschränkung der möglichen Lösungen und Lösungswege akzeptabel ist.

3.7.1 Screenography am Beispiel einer Museumsanwendung

Folgend wird exemplarisch eine Museumsanwendung beschrieben, die zwei Zugangsarten der Benutzung unterscheidet. Als Hauptzugangsarten sollen die *Suche* und der *Rundgang* unterschieden werden. Die Zugangsarten sollen auf die gleiche Datenbasis zurückgreifen,

sich aber in ihren Aufgaben und Zielen unterscheiden. Hiervon ist das Gewicht einzelner Faktoren und deren Abhängigkeiten betroffen sowie bedingt verschiedene strategische Einflussgrößen.

Suche

Mit dem Begriff der Suche wird in der Regel eine mehr oder minder komplexe Stichwort- und Themensuche verbunden. Soweit spezifiziert, können auch Stories berücksichtigt bzw. der Einstieg in passende Handlungen empfohlen werden. Ziel der Suche ist das möglichst einfache und schnelle Auffinden von Informationen, sodass im besten Fall die Suchhandlung einer minimalen echten Story $S_{r_{min}}$ entspricht. Die Unterstützung des Akteurs bei der Herausbildung seines mentalen Modells des Handlungsraums ist in der Regel keine Aufgabe der Suche. Beim gewählten Beispiel der Museumshandlung umfasst die Suche das Auffinden von Werken und Informationen zu diesen bzw. über diese. Für die Entwicklung der Darstellung reicht diese Zielstellung allein nicht aus, da sie unter anderem keine Aussage darüber trifft, ob die gesamte Zielgruppe oder der konkrete Akteur *proaktiv* oder *reaktiv* agiert. Daher werden folgend diese zwei Varianten separat betrachtet.

Die *proaktive Suche* setzt Wissen und Annahmen des Akteurs über die gesicherten Informationen der Anwendung voraus und eignet sich z.B. für den Abruf von Messgrößen, Definitionen oder bekannten Objekten. Dieser ergebnisorientierte Ansatz ist hilfreich, wenn Suchen eine zentrale Funktion der Anwendung ist und der Akteur hinreichend präzise Suchterme formulieren kann. Letzteres ist vor allem in Gegenwart von Deskriptoren möglich, die nicht künstlich erzeugt wurden und den Akteuren gut bekannt sind. Beispielhaft seien hier Online-Shops (Produktsuche), -Enzyklopädien und -Wörterbücher genannt. Eine Vielzahl von Anwendungen eignet sich jedoch nicht für diese Form der Unterstützung, weil nur künstliche Deskriptoren existieren, sie nicht allgemeingültig oder allgemein bekannt sind. Bei einer Volltextsuche ist mit zunehmendem Informationsumfang eine abnehmende Treffergenauigkeit und Qualität der Suchergebnisse zu erwarten, da die Präzision der vom Akteur formulierten Suchterme typischerweise konstant bleibt, jedoch oft die Verschiedenheit der Inhalte abnimmt. Ferner kann mit der proaktiven Suche ein Zielkonflikt entstehen, da eine hohe Treffergenauigkeit zwar einen schnellen Weg zur Information ermöglicht, der Akteur aber nicht darin unterstützt werden kann, den Aufbau der Anwendung kennenzulernen. Infolgedessen trägt der wiederholte Besuch eines solchen Informationssystems nur geringfügig dazu bei, das Orientierungsvermögen des Akteurs zu steigern und ihm damit die Benutzung zu erleichtern. Für eine Museumsanwendung bedeutet die Verwendung der proaktiven Suche, dass für den Akteur Wissen über geeignete Schlüsselworte vorausgesetzt wird, anhand derer sich Inhalte oder Handlungen finden lassen. Beispielhaft seien hier der Name eines Künstlers oder der einer speziellen Epoche genannt.

Mit Blick auf die Entwicklungsdimensionen (Abb. 3.17) ist für das Realisieren einer proaktiven Suche die *Functionality*-Dimension von zentraler Bedeutung, mitunter in Berücksichtigung von Vorgaben der *Context*-Dimension. Da die *Functionality*-Dimension entscheidet in welcher Weise den Nutzern Zugang zu den Inhalten gewährt wird, do-

minierte sie auch direkt die *Presentation*-Dimension, die überdies durch die *Intention*-Dimension beeinflusst sein kann. Handlungen lassen sich für diesen Suchtyp nicht realisieren, weshalb die *Story*-Dimension entfällt. Der *Content*-Dimension obliegt in diesem Fall schließlich nur noch die Aufgabe, Inhalte derart abzulegen, dass sie beim späteren Suchen entsprechend der verfügbaren Funktionen gefunden und in der von der Darstellung geforderten Form zur Verfügung gestellt werden können.

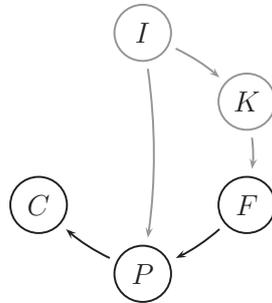


Abbildung 3.17: Museums Story - Proaktive Suche

Das *reaktive Suchen* erfordert keine Vorkenntnisse über die Anwendung oder geeignete Schlüsselworte. Da dem Akteur die verfügbaren Themen, Handlungen und Filter zur Auswahl geboten werden, ist ihm die reaktive Suche auch dann möglich, wenn ihm Eigenschaften bislang unbekannt oder nur unscharf bekannt waren. Diese Möglichkeit ist vor allem dann von Bedeutung, wenn sich der Akteur schrittweise über Themengebiete großen Umfangs zu informieren versucht, deren Gliederung er nicht kennt. Eine Schwierigkeit besteht jedoch darin, anhand eines gefundenen Themas einen geeigneten Einstiegspunkt in dieses zu wählen als auch darin, ein passendes Detail-Level für die verfügbaren Inhalte zu entscheiden. Eine Schwäche dieses Ansatzes wird mit wachsendem Informationsumfang und sich in zunehmendem Maße überschneidenden Themengebieten deutlich, weil dann auch die geeignete Darbietung der Themen schwieriger wird. Infolgedessen wird die mögliche Interaktionsqualität sinken und die Länge des Weges zur Information ansteigen. Durch eine geeignete Gewichtung der Themen, kann diese Problematik abgemildert werden, jedoch zu Lasten der Auffindbarkeit gering priorisierter oder selten erfragter Themen. Für die Museumsanwendung eignet sich dieser Ansatz insbesondere dann, wenn die Akteure initial höchstens sehr allgemeine Suchkriterien angeben können. Die reaktive Suche ermöglicht in diesem Fall das Auffinden eines Werks durch schrittweise Verfeinerung. Dies ist auch dann hilfreich, wenn dem Akteur die Existenz eines Werks zwar bekannt ist, er jedoch nicht dessen eindeutige Identifikatoren benennen kann.

Für die reaktive Suche ist hinsichtlich der Entwicklungsdimensionen vor allem die *Content*-Dimension bedeutsam (Abb. 3.18). Während bei der proaktiven Suche für das Auffinden der Inhalte die Existenz einer thematischen Klassifikation der Inhalte ausreichend ist, muss diese bei der reaktiven Suche den Akteuren zugänglich sein, da andernfalls unterstützendes Leiten nicht möglich ist. Die erforderliche Funktionalität wird maßgeblich durch die *Content*-Dimension sowie bedingt durch die *Context*-Dimension be-

stimmt. Einflüsse durch die *Story*-Dimension können entstehen, wenn der Handlungspfad der Akteure berücksichtigt und mit bekannten Handlungsmustern verglichen werden soll. Diese Muster können auch in Entscheidungen einbezogen werden, ob eine weitere Verfeinerung der Suchanfrage nötig oder der Einstieg in eine Handlung möglich ist, aber auch bewerten, an welcher Stelle ein Einstieg in die Handlung sinnvoll erscheint. Die Entwicklung der *Presentation*-Dimension wird vor allem durch die zur Verfügung zu stellende Funktionalität bestimmt. Weil die Darstellung der gesuchten Inhalte streng genommen nicht mehr Teil der Suche ist, sondern deren Ergebnis, besitzt die *Content*-Dimension keinen direkten Einfluss. Mindestens implizit ist aber an der Entscheidung der geeigneten Darstellung die *Intention*-Dimension beteiligt, da die dem Akteur dargebotenen Themen, zur Verfeinerung seiner Suche, auch dessen Kenntnisse beachten müssen.

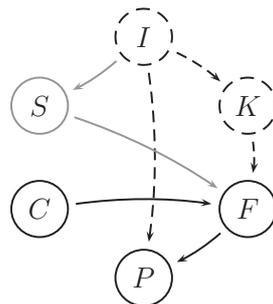


Abbildung 3.18: Museums Story - Reaktive Suche

Rundgang

Ziel eines Rundgangs ist das Erleben und Entdecken von Informationen im Rahmen einer abgeschlossenen Handlung. Um dies zu erreichen, wird die Handlung eines Rundgangs inhaltlich und dramaturgisch aufbereitet und sollte stets terminieren können. Die inhaltliche Beschränkung und Fokussierung der Sicht wird durch das Rundgang-Thema vorgegeben. Diese Selektion bietet Orientierung und erleichtert dem Akteur die Informationsaufnahme und in der Konsequenz das Bearbeiten von Aufgaben und Lösen von Problemstellungen. Da ein Rundgang verschiedenen Zwecken dienen kann, sollen folgend für das Museumsbeispiel *einfache Rundgänge* mit fixiertem Handlungsverlauf von Solchen unterschieden werden, deren Handlungsverlauf individuell und zielgerichtet vorangetrieben werden kann, um u.a. Wissen vermitteln und den Wissensstand überprüfen zu können. Voraussetzung für *individuelle Rundgänge* ist die Möglichkeit einer starken, aktiven Beteiligung des jeweiligen Akteurs.

Der *einfache Rundgang* ist dann geeignet, wenn Inhalte ähnlicher Art präsentiert werden sollen, die zumeist losgelöst voneinander verstanden werden können. Begleitende Informationen als auch eine uneinheitliche Informationstiefe, die über das bereits definierte Maß hinausgeht, sind diesem Rundgang-Typ fremd. Die mögliche dramaturgische Aufbereitung ist daher stark beschränkt, ebenso die erforderliche Funktionalität. Das Ziel einfacher Rundgänge ist das Kennenlernen und Motivieren ausgewählter Themen.

Im Unterschied zur parameter-getriebenen Suche sind daher nicht alle zum Rundgang-Thema passenden Inhalte Teil des späteren Rundgangs. Vielmehr erfolgt durch den Rundgang-Ersteller, als Dramaturg, eine subjektive Auswahl und Anordnung geeigneter Repräsentanten, mit dem Ziel das Interesse an der Handlung hervorzurufen und es ferner zu erhalten. Der Handlungsverlauf des einfachen Rundgangs ist nicht akteursspezifisch. Daraus erwächst jedoch der Vorteil, dass dieser Rundgang-Typ außerhalb des Informationssystems kommunizierbar und referenzierbar bleibt. In einer Museumsanwendung bietet der einfache Rundgang seinen Interessenten einen leichten Einstieg in vorbereitete Themen. Darüber hinaus kann mit diesem Rundgang-Typ das vorhandene Inventar ohne großen, zusätzlichen Aufwand aus verschiedenen, thematischen Blickwinkeln betrachtet werden, wodurch die Attraktivität für die Nutzer steigt und weitere Zielgruppen angesprochen werden können.

Der einfache Rundgang wird üblicherweise in Kenntnis existierender Lösungen entwickelt und basiert auf Rahmenbedingungen und Zielen geringer Komplexität. Da auch der Verlauf der Handlung nicht über ein geringes Maß an Komplexität hinaus gehen soll, sind die Spezifikationen der Dimensionen *Intention*, *Story* und *Context* nicht erforderlich (Abb. 3.19) und können vielmehr als implizit vorhanden betrachtet werden. Für den einfachen Rundgang ist die Auswahl und Anordnung der Inhalte von zentraler Bedeutung. Daher ist zunächst die *Content*-Dimension zu spezifizieren, die Ihrerseits direkt die Entwicklung der *Presentation*-Dimension beeinflusst. Der Fokus auf Einfachheit führt zu einem bewusst geringen Einfluss der *Functionality*-Dimension, deren Entwicklung infolgedessen erst spät entschieden werden kann.

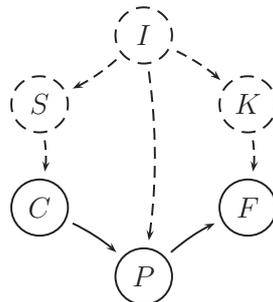


Abbildung 3.19: Museums Story - Einfacher Rundgang

Der *individuelle Rundgang* ist unter anderem dann hilfreich, wenn Wissen vermittelt werden soll, sich das Interesse der Akteure aber anfänglich oder auch final vor allem auf Teilaspekte verfügbarer Themengebiete beschränkt. Zur Vermeidung der Spezifikation einer Vielzahl stark redundanter einfacher Rundgänge, erlaubt der individuelle Rundgang nicht nur mehrere Einstiegs- und Endpunkte für die Handlung, sondern ermöglicht dem Akteur ferner, den Verlauf der Handlung aktiv zu steuern. Da Inhalte nicht in beliebiger Reihenfolge verstanden werden können, wirken Abhängigkeiten aufeinander aufbauender Inhalte beschränkend hinsichtlich der Flexibilität. Der Aufnahme und Annotation der Inhalte ist daher eine starke Bedeutung beizumessen, um diese Abhängigkeiten auf

ein Mindestmaß zu beschränken. Ziel sollte es sein, dass Inhalte möglichst eigenständig verstanden werden können, ohne zugleich vermehrt Redundanzen zu erzeugen.

Beim individuellen Rundgang zählen ergänzende und weiterführende Informationen zu den Erwartungen der Nutzer. Diese können genutzt werden, um in Abhängigkeit vom beschrittenen Pfad sowie den erlangten und bereits vorhandenen Kenntnissen zu steuern, ob und welche Informationen ergänzend zu präsentieren sind. In einer Museumsanwendung bietet der individuelle Rundgang seinen Nutzern damit die Flexibilität, sich Themen aus verschiedenen Perspektiven nähern zu können, ohne dem Zwang vorgefertigter Handlungen zu unterliegen. Das Beeinflussen des Handlungsverlaufs ist nicht nur erwünscht, sondern zählt beim individuellen Rundgang zu den Aufgaben des Akteurs. Verglichen mit den bisherigen Zugängen erfordert das Realisieren des individuellen Rundgangs einen erkennbar größeren Aufwand, aufgrund des größeren Aufwands für das Erstellen, Ablegen und Annotieren von Inhalten. Andererseits ergibt sich jedoch der Vorteil, auch bisher unbeachtete Zugänge zu Themen bedienen zu können und diese nicht gesondert spezifizieren zu müssen. Ferner bietet sich den Nutzern der Vorteil, dass Lernen auch außerhalb eines starren Ablaufs möglich wird und daher unterschiedliche Formen des Lernens abbildbar werden.

Für das Realisieren individueller Rundgänge, ist zunächst die *Intention*-Dimension zu spezifizieren, da diese durch die Wahl von Zielen maßgeblich den erforderlichen Entwicklungsaufwand mitentscheidet (Abb. 3.20). Dies wirkt einerseits direkt auf die *Presentation*-Dimension, beeinflusst andererseits aber auch die Entwicklung der *Story*-Dimension. Die *Story*-Dimension ist zu entwickeln, um dem Wunsch eines individuellen Handlungsverlaufs Rechnung tragen zu können. Die *Content*-Dimension wird von dieser Entwicklung beeinflusst, aber auch durch die *Functionality*-Dimension. Der Einfluss der *Functionality*-Dimension ist der Tatsache geschuldet, dass die Art der geplanten Interaktion und Navigation auf Basis der *Story*-Dimension darüber entscheidet, welche Inhalte in welcher Weise vorzuhalten sind. Die Entwicklung der *Presentation*-Dimension erfolgt beim individuellen Rundgang zuletzt und erlaubt dadurch größtmögliche Flexibilität und Adaptivität.

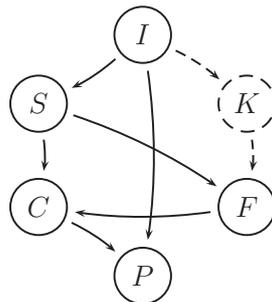


Abbildung 3.20: Museums Story - Individueller Rundgang

4 Patterns und Grids

4.1 Grids

Bereits in den zwanziger Jahren des 20. Jahrhunderts entstanden neue theoretische und gestalterische Überlegungen für eine moderne Typografie, die auch die Entwicklung von allgemeinen Ordnungssystemen umfasste. Zu den bedeutendsten Pionieren dieser Strömung zählte Jan Tschichold [69]. Beeinflusst von seinen und anderen Arbeiten, wurden in den vierziger Jahren *Raster* bzw. *Grids* als Ordnungssystem entwickelt und verwendet. Explizit erwähnt wurden Grids erstmals 1961 in einem Werk von Müller-Brockmann [38]. Müller-Brockmann betrachtete Grids stets als Hilfsmittel, um Inhalte in Übereinstimmung mit objektiven und funktionalen Kriterien platzieren zu können [39]. Die Organisation der Darstellungsfläche durch Grids ist inzwischen ein nahezu unverzichtbarer Bestandteil der Entwicklung von professionellen Drucksachen. Häufig teilen die Grids die Darstellungsfläche durch horizontale und vertikale Linien in mehrere Teilbereiche, wodurch eine allgemeine Vorlage mit Halte- und Orientierungspunkten entsteht, an denen sich die späteren Inhalte in einfacher Weise ausrichten lassen. Oft beinhalten die entwickelten Vorlagen sogar mehrere Gruppen zusammengehöriger Haltepunkte und integrieren somit mehrere Organisationsalternativen. Aufgrund der fehlenden, abgrenzenden Kennzeichnung, erschwert dies jedoch das intuitiv richtige Erkennen der Zusammenhänge und den Umgang mit solch komplexen Vorlagen.

Allgemein lässt sich feststellen, dass Grids mehrdimensionale Präsentationsbereiche in visuelle Teilbereiche teilen, mit dem Ziel die Gestaltung zu flexibilisieren und durch Wiederverwendbarkeit zu erleichtern. Die Teilbereiche müssen dabei nicht zwingend vollständig disjunkt sein. Existieren jedoch mehrfach genutzte Bereiche, so ist die Hierarchie der Bereiche zu definieren, um Darstellungsfehler, z.B. infolge transmissiver Elemente, und Verdeckungsprobleme, infolge von Ebenenüberdeckungen, zu vermeiden.

Anders als bei den Nicht-elektronischen Dokumenten beschreibt ein Grid G für die Gestaltung von Weboberflächen jeweils eine einzelne, konkrete Organisationsalternative. Die Mächtigkeit der Nicht-elektronischen Pendanten kann erhalten werden, indem geeignete, alternative Grids zu einem Grid-Bündel G_B vereint werden. Typischerweise weisen Grids eines solchen Bündels nur geringe Unterschiede in kleinen Teilbereichen auf. Da ferner jedes Grid aus anderen Grids bestehen kann, lassen sich Redundanzen vermeiden, indem organisationsalternative Grids nur jene Bereiche beschreiben, die von Änderungen betroffen sind. Der genaue Einsatzzeitpunkt der Grids eines Grid-Bündels lässt sich durch Regeln definieren. Dies erlaubt abwechslungsreiches Präsentieren von Inhalten, ohne dabei die Gestaltung einzelner Seiten eines Webauftritts von Hand festlegen zu müssen. Überdies ist es hilfreich, wenn Grids für Weboberflächen nicht nur die

Position seiner Elemente innerhalb des Handlungsraums festlegen, sondern diesen auch Eigenschaften zuweisen sowie Bedingungen und Restriktionen definieren.

Das Definieren der Gridelemente eines Grids $e_n \in G_i$ mit $n, i \in \mathbb{N}$ kann auf unterschiedliche Weise erfolgen. Im einfachsten Fall können Grids für Weboberflächen den gesamten Präsentationsbereich durch Parkettierung (überlappungsfrei) vollständig in 2-dimensionale Rechteckflächen [37] unterteilen, wie beispielhaft dargestellt in Abbildung 4.1. Da jedes Grid aus mehreren Subgrids bestehen kann, erleichtert dies die Integration und Wiederverwendung von Teilbereichen bereits entwickelter Grids.

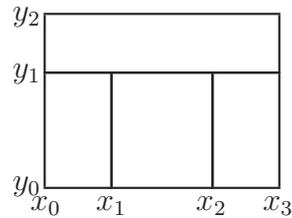


Abbildung 4.1: 3-Spalten Grid: Aufbau

Typischerweise werden horizontale und vertikale Achsen als Hilfsmittel zur Festlegung der Gridpunkte unterschieden, um Rechteckflächen beschreiben zu können. Die horizontalen Gridpunkte umfassen den Bereich $x_{min} = x_0 < x_1 < \dots < x_m = x_{max}$, die vertikalen Gridpunkte $y_{min} = y_0 < y_1 < \dots < y_n = y_{max}$. Eine Rechteckfläche soll definiert sein als $[x_i, x_j] \times [y_k, y_\ell]$. Dies erlaubt eine einfache Partitionierung der Bildschirmfläche in obige Teilbereiche, indem 4 horizontale Gridpunkte $x_0 < x_1 < x_2 < x_3$ und 3 vertikale Gridpunkte $y_0 < y_1 < y_2$ verwendet werden. Mittels dieser Gridpunkte lassen sich die folgenden 4 Teile definieren:

$$\begin{array}{ll} \text{up} = [x_0, x_3] \times [y_1, y_2] & \text{left} = [x_0, x_1] \times [y_0, y_1] \\ \text{middle} = [x_1, x_2] \times [y_0, y_1] & \text{right} = [x_2, x_3] \times [y_0, y_1] \end{array}$$

Gewöhnlich wird der “up” Bereich für ein Menü verwendet, der “left” Bereich für die Navigation, der “middle” Bereich für den Hauptinhalt und der “right” Bereich für Zusatzoptionen.

4.1.1 Grid Geometrie

Das oben beschriebene Teilungsverfahren ist sehr einfach, da es lediglich Zeilen und Spalten ohne weitere Einschränkungen unterscheidet. Anspruchsvoller wird das Verfahren sobald die Elemente des Grids rhythmischen Strukturen folgen sollen, um die Wahrnehmbarkeit der Bedeutung eines Bereichs zu verbessern. So werden unter anderem größere Elemente einer Reihe als wichtiger empfunden als kleine Elemente, insbesondere dann, wenn es sich um eine monotone Reihe handelt.

Ein Beispiel für eine geeignete Reihe sind die Fibonacci-Zahlen, die in der Kunst und Architektur seit Jahrhunderten von großer Bedeutung sind, um den Blick des Betrachters durch Längen-, Größen- und Abstandsverhältnisse gezielt zu lenken. Die Fibonacci-Zahlen können durch Rekursion über das Bildungsgesetz $f_{n+2} = f_n + f_{n+1}$ ermittelt

werden, unter Verwendung der initialen Werte $f_0 = 0$ und $f_1 = 1$. Dies führt zu der gut bekannten Folge $0, 1, 1, 2, 3, 5, 8, 13, \dots$

In Kunst und Architektur wird der *Goldene Schnitt* (lat. *sectio aurea*) oft als *ideale Proportion* aufgefasst. Dieser wird erzielt, wenn das Verhältnis der größeren Strecke zur Kleineren dem der Gesamtstrecke (Summe der Einzelstrecken) zur Größeren entspricht.

$$\phi = \frac{a}{b} = \frac{a+b}{a} = \text{Goldener Schnitt} \quad (4.1)$$

Aus der Verhältnis-Beziehung ergibt sich folgendes Teilungsverhältnis:

$$\phi = \frac{1 + \sqrt{5}}{2} \approx 1,618 \quad (4.2)$$

Wird das Verhältnis zweier, aufeinander folgender Fibonacci-Zahlen berechnet, so entspricht dieses in Näherung dem Wert des *Goldenen Schnitts* (Gl. 4.2). Je größer die ins Verhältnis gesetzten Fibonacci-Zahlen sind, desto exakter stimmt das Verhältnis mit dem *Goldenen Schnitt* überein. Abbildung 4.2 zeigt ein einfaches Karteireiter-Grid, bei dem der Abstand der horizontalen Gridpunkte gemäß des Teilungsverhältnisses (Gl. 4.2) monoton ansteigt. Infolgedessen entstehen harmonische, nicht-linear breiter werdende Karteireiter.



Abbildung 4.2: Rhythmische Struktur von Gridelementen

Eine weitere Möglichkeit der Nutzung der Fibonacci-Folge zur Parkettierung rechteckiger Präsentationsflächen besteht in der ausschließlichen Nutzung von Quadraten als Grundform für Gridelemente. Die Konstruktion eines solchen Grids beginnt mit einem Quadrat dessen Kantenlänge normierend der ersten Fibonacci-Zahl $f_1 = 1$ entspricht. Ein Vorteil der Normierung ist die spätere, beliebige Skalierbarkeit des Ergebnisses. Ferner sei definiert, dass die Kantenlänge zweier, aufeinander folgender Gridelemente im Verhältnis gemäß der Fibonacci-Zahlen ansteigen soll, wodurch die Platzierung des jeweils nächsten Elements f_{n+1} immer an einer der längsten Kanten des bisherigen Gesamtobjekts erfolgen kann und soll. Auf diese Weise entsteht ab dem dritten Element ($f_3 = 2$) als Gesamtobjekt ein sogenanntes *Goldenes Rechteck*, dessen Seitenverhältnis dem *Goldenen Schnitt* entspricht. Erfolgt die Platzierung bzw. das Anfügen neuer Elemente stets unter Beibehaltung einer Drehrichtung, so entsteht eine Grid-Spirale, ähnlich jener in Abbildung 4.3. Die Drehrichtung der Spirale entscheidet sich mit dem Anfügen des dritten Elements und ist daher ab dem vierten Element einzuhalten.

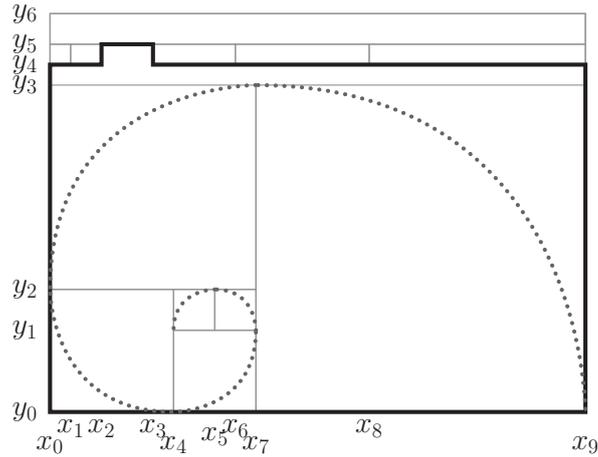


Abbildung 4.3: Fibonacci Grid Modell

Das Fibonacci-Grid der Abbildung 4.3 wurde durch Moritz [34] als flexibles Grid für Weboberflächen eingeführt und kann ebenfalls mittels der oben eingeführten Gridpunkt Definition beschrieben werden.

$$\begin{aligned}
 \text{head} &= [x_0, x_9] \times [y_5, y_6] & \text{nav_spot1} &= [x_0, x_1] \times [y_4, y_5] \\
 \text{nav_spot2} &= [x_1, x_2] \times [y_4, y_5] & \text{nav_spot3} &= [x_2, x_3] \times [y_4, y_5] \\
 \text{nav_spot4} &= [x_3, x_6] \times [y_4, y_5] & \text{nav_spot5} &= [x_6, x_8] \times [y_4, y_5] \\
 \text{nav_spot6} &= [x_8, x_9] \times [y_4, y_5] & \text{nav_active} &= [x_0, x_9] \times [y_3, y_4] \\
 \text{square1} &= [x_4, x_5] \times [y_1, y_2] & \text{square2} &= [x_5, x_7] \times [y_1, y_2] \\
 \text{square3} &= [x_4, x_7] \times [y_0, y_1] & \text{square4} &= [x_0, x_4] \times [y_0, y_2] \\
 \text{square5} &= [x_0, x_7] \times [y_2, y_3] & \text{square6} &= [x_7, x_9] \times [y_0, y_3]
 \end{aligned}$$

4.1.2 Grid Typen

Bei der Anwendungsentwicklung sind typischerweise globale Richtlinien wie das Sicherstellen des Erfüllens von Corporate Identity Anforderungen zu berücksichtigen. Zur besseren Abgrenzung wird daher zwischen globalen und lokalen Grids unterschieden:

- frame grids / tiles
- body grids / tiles

Frame grids sind starre Teile der Präsentation, die innerhalb aller Seiten eines Webauftritts unverändert zur Anwendung kommen. Diese Grids bilden das Fundament für die Orientierung und sollen das Bewegen im Handlungsraum erleichtern. Frame grids gilt es mit Inhalten anzureichern, die sich höchstens geringfügig ändern, wobei etwaige Änderungen intuitiv nachvollziehbar sein müssen. Insbesondere große Webauftritte benötigen Orientierungshilfen, da eine große Navigationstiefe und -breite schnell zur Orientierungslosigkeit führen kann.

Body grids betreffen die Repräsentation von nicht-globalen Contentobjekten. Sie erlauben gezielt Individualisierungen, um eine abwechslungsreiche Darstellung sowie Adaption zu ermöglichen. Die Anwendung eines konkreten Body grids hängt maßgeblich vom Story-Verlauf ab, unter Berücksichtigung globaler Definitionen und Restriktionen.

4.1.3 Grid Spezifikation

Dieser Abschnitt spezifiziert den Aufbau und Umgang mit Grids und führt in diesem Zusammenhang geeignete Manipulatoren ein. Ziel dieser Spezifikation ist es, möglichst flexibel Grids für beliebige Ordnungssysteme erzeugen zu können. Obwohl der Fokus auf der Entwicklung von Weboberflächen liegt, soll die Spezifikation der Grids nicht auf die derzeitigen Möglichkeiten, Anforderungen und Grenzen dieses Bereichs beschränkt sein, sondern auch den Bedürfnissen anderer Domänen gerecht werden sowie potentielle, zukünftige Anforderungen berücksichtigen.

Jedes Grid G_i kann beliebig viele Gridelemente $e \in G$ enthalten. Innerhalb eines Grids sollen alle Gridelemente eindeutig unterscheidbar sein, weshalb die Gridelemente um den Index $n \in \mathbb{N}$ zu ergänzen sind. Jedes Gridelement darf als Vaterelement auftreten und seinerseits weitere Gridelemente enthalten. Darüber hinaus können jedem Grid G_i neben den Gridelementen e_n auch Instanzen von bereits existierenden Grids G_{x_j} zugeordnet werden. Das Löschen eines Grids ist infolgedessen nur möglich, wenn für dieses keine Gridinstanzen existieren oder alle Verwendeten gelöscht werden. Alle eingebetteten Gridinstanzen müssen eindeutig von anderen Instanzen des Grids unterscheidbar sein, um beispielsweise Mehrfacheinbettungen innerhalb des Darstellungsraums erzielen zu können. Unabhängig von der Hierarchieebene darf ein Grid G_i selbstverständlich keine Instanzen von sich selbst enthalten, weshalb für jede eingebettete Gridinstanz G_{x_j} gilt:

$$\forall G_{x_j} \subset G_i : G_{x_j} \neq G_i \quad (4.3)$$

Alle Gridelemente einer Gridinstanz sind für das einbettende Grid sowie für andere Gridinstanzen dieses Grids transparent, sodass die Forderung nach eindeutig unterscheidbaren Gridelementen innerhalb eines Grids nicht die Gridelemente der Gridinstanzen einschließt. Allgemein lässt sich ein Grid als Vereinigung von Gridelementen und Gridinstanzen definieren.

$$G_i - \text{Grid } i \quad (4.4)$$

$$G_{i_j} - \text{Instanz } j \text{ eines Grids } G_i \quad (4.5)$$

$$G_i = \bigcup_{x=1}^m G_{x_j} \wedge \bigcup_{n=1}^z e_n \quad x, j, n \in \mathbb{N} \quad (4.6)$$

Folgend werden verschiedene Grid-Manipulatoren eingeführt, die den Aufbau, die Erweiterung, Reduktion und Veränderung von Grids ermöglichen. Der besseren Veranschaulichung wird in diesem Zusammenhang ein Beispielgrid eingeführt und schrittweise

weiterentwickelt. Das Beispielgrid besteht initial aus 6 Elementen, umfasst 2 Hierarchielevel. Vereinfachend enthält das Beispiel keine Gridinstanzen, da diese eine Erweiterung, Flexibilisierung und Erleichterung der Gridentwicklung bedeuten, jedoch keine Notwendigkeit darstellen.

$$G_0 = \{e_1, e_2, e_3\{e_5, e_6\}, e_4\}$$

Einfügen eines Gridelements (*ins*): Um einem Grid ein neues Element hinzuzufügen, wird der Operator *ins* verwendet. Neben dem Operator ist das neu einzufügende Element anzugeben, welches eindeutig innerhalb des Grids sein muss. Optional kann zudem die Position angegeben werden, an der das neue Element im bestehenden Elementbaum eingefügt werden soll. Die Positionsangabe identifiziert das unmittelbare Eltern-element, infolgedessen diese Angabe als *Einfügen unterhalb* interpretiert werden kann. Beim Einfügen eines neuen Elements an der Wurzel (Rootlevel) kann auf die Positionsangabe verzichtet werden. Allgemein wird das Einfügen mit der folgenden Operation beschrieben:

$$ins_{element_{e_identifizier}(\curvearrow parent_element_{pe_identifizier})}(G_{G_identifizier})$$

Soll ein neues Element nicht auf dem Rootlevel des gewählten Grids eingefügt werden, so muss das angegebene *parent_element* zum Zeitpunkt des Einfügens tatsächlich existierendes Element des Grids sein. Für das Einfügen von Elementen unter Angabe eines *parent_elements* muss somit gelten:

$$parent_element_{pe_identifizier} \in G_{G_identifizier}$$

Exemplarisch, kann e_7 in G_0 unterhalb von e_4 mittels folgender Operation eingefügt werden:

$$\begin{aligned} G_1 &= ins_{e_7 \curvearrow e_4}(G_0) \\ G_1 &= \{e_1, e_2, e_3\{e_5, e_6\}, e_4\{e_7\}\} \end{aligned}$$

Das Einfügen von Elementen gleicher Identifikation, wie sie innerhalb des Grids bereits verwendet werden, soll unabhängig vom Ort des Einfügens verboten sein, um die Eindeutigkeit der Elemente absichern zu können. Folglich bewirkt das Einfügen von e_6 in G_1 keine Änderungen in G_1 .

$$G_1 = ins_{e_6 \curvearrow e_4}(G_1)$$

In einigen Situationen kann die Reihenfolge der Elemente einer Ebene im Elementbaum von Bedeutung sein. Der Operator *ins* legt jedoch bewusst nicht fest, an welcher Position innerhalb der Ebene ein neues Element unterhalb des angegebenen Elternelements angefügt wird, um eine einfache und intuitive Verwendung sicherzustellen. Zur Vermeidung von Beschränkungen und Konflikten, wird daher später in diesem Kapitel der Verschiebe-Operator *shift* eingeführt, der das Ändern der Reihenfolge von Elementen

einer Ebene erlaubt. Trotz der prinzipiellen Möglichkeit, werden neue Elemente weder zufällig noch regelfrei in den Elementbaum eingefügt. Existiert beim Einfügen kein Kindelement zum angegebenen Elternelement, so wird eine neue Ebene erzeugt, der das neue Element hinzugefügt wird. Sobald auf der Kindebene bereits Elemente existieren, verhält sich der Operator *ins* wie ein *append*, wodurch ein neues Element jeweils dem letzten Element der Kindebene angehängt wird.

Löschen eines Gridelements (*del*): Das Löschen eines Gridelements lässt sich durch Benutzung des Operators *del* erreichen, ergänzt um den Identifikator des betroffenen Elements. Die Verwendung dieser Operation ist nur dann möglich, wenn ein Element angegeben wurde, das innerhalb des gewählten Grids existiert. Andernfalls ist die Operation wirkungslos. Das Löschen wird allgemein mit folgender Operation beschrieben:

$$del_{element_{e.id}}(G_{G.id})$$

$$element_{e.id} \in G_{G.id}$$

Soll ein Element gelöscht werden, das seinerseits über Kindelemente verfügt, so werden auch die Kindelemente gelöscht. Zukünftige Implementierungen sollten den Nutzer darauf hinweisen, ob vom Löschen eines Elements auch weitere Elemente betroffen sind, da trotz einer hierarchischen Vater-Kind-Beziehung die visuelle Position eines Kindelements nicht immer innerhalb des Väterelements gelegen sein muss. Beispielhaft wird das Element e_3 aus dem Grid G_1 gelöscht, wodurch auch die Elemente e_5 und e_6 entfernt werden:

$$G_2 = del_{e_3}(G_1)$$

$$G_2 = \{e_1, e_2, e_4\{e_7\}\}$$

Das Löschen eines Elements ohne dessen Kindelemente lässt sich nicht vollständig automatisieren ohne zu beschränken, da im Falle dynamischer Positionierung die automatische Integration in die übergeordnete Ebene nicht immer dem gewünschten Ergebnis entspricht. Soll dennoch ein Element in dieser Weise gelöscht werden, so sind zunächst alle weiterhin benötigten Kindelemente aus dem zu löschenden Element heraus mit dem *move* Operator zu verschieben bevor das Element tatsächlich gelöscht werden kann. Die Einführung des *move* Operators erfolgt später in diesem Kapitel.

Verschieben eines Gridelements (*shift*): Entsprechend der obigen Definition, erlaubt der Operator für das Einfügen von Gridelementen *ins* nur bedingt, die exakte Position eines neuen Elements zu wählen. Da jedoch in einigen Situationen die Reihenfolge der Elemente innerhalb einer Hierarchie-Ebene von Bedeutung ist, wird ein Verschiebe-Operator *shift* benötigt, der es gestattet, die Reihenfolge von Geschwister-Elementen zu ändern. Dazu ist nicht nur das Element zu benennen, welches verschoben werden soll, sondern auch die Richtung anzugeben, in die das gewählte Element geschoben werden soll. Demgemäß ist zu unterscheiden zwischen dem Verschieben nach links bzw. nach

vorn und dem Verschieben nach rechts bzw. nach hinten in der Liste der Geschwister. Das Verschieben nach links wird gekennzeichnet durch das Symbol '<' und das Verschieben nach rechts durch das Symbol '>'. Allgemein lässt sich das Verschieben von Elementen wie folgt ausdrücken:

Verschieben eines Elements um eine Stelle nach vorn:

$$\mathit{shift}_{\mathit{element}_{e_{id}},<}(G_{G_id})$$

Verschieben eines Elements um eine Stelle nach hinten:

$$\mathit{shift}_{\mathit{element}_{e_{id}},>}(G_{G_id})$$

Das folgende Beispiel verschiebt das Element e_4 innerhalb von G_2 um eine Stelle nach vorn in der Liste der Geschwisterelemente.

$$\begin{aligned} G_3 &= \mathit{shift}_{e_4,<}(G_2) \\ G_3 &= \{e_1, e_4\{e_7\}, e_2\} \end{aligned}$$

Analog lässt sich das Verschieben nach hinten erreichen, wie das folgende Beispiel zeigt. In diesem Fall wird das Element e_1 innerhalb des Grids G_3 um eine Stelle nach hinten verschoben.

$$\begin{aligned} G_4 &= \mathit{shift}_{e_1,>}(G_3) \\ G_4 &= \{e_4\{e_7\}, e_1, e_2\} \end{aligned}$$

Mit den bisherigen Operatoren lässt sich nun jedes Element innerhalb des Grids exakt positionieren. Erweiterungen um spezielle Verschiebe-Operatoren sind nicht zwingend erforderlich, aber hilfreich. Soll beispielsweise ein Element um mehrere Positionen verschoben werden, so ist dies mit der bisherigen Notation nicht in knapper Form möglich. Daher lohnt die Erweiterung der Notation um die Option, die Anzahl der Iterationen der Verschiebung angeben zu können. So bewirkt das Angeben von '> $_l$ ', dass das Element um l Stellen nach rechts verschoben wird. Analog bedeutet '< $_m$ ' ein Verschieben des Elements um m Stellen nach links.

Beispielhaft wird das Element e_4 innerhalb des Grids G_4 um 2 Stellen nach rechts verschoben.

$$\begin{aligned} G_5 &= \mathit{shift}_{e_4,>_2}(G_4) \\ G_5 &= \{e_1, e_2, e_4\{e_7\}\} \end{aligned}$$

Die Vorteile der erweiterten Notation sind bereits bei einer Verschiebung um 2 Stellen erkennbar, da diese Redundanzen vermeidet und im Falle von Mehrfachverschiebungen leichter zu verstehen und anzuwenden ist. Wird anstelle der erweiterten Notation die

Basisnotation verwendet, um das Element e_4 innerhalb von G_4 um 2 Stellen nach rechts zu verschieben, so entsteht folgender Ausdruck:

$$G_5 = \mathit{shift}_{e_4, >}(\mathit{shift}_{e_4, >}(G_4))$$

Neben dem gezielten Angeben der Iterationen kann das direkte Verschieben von Elementen an den Anfang oder das Ende der Liste von Geschwister-Elementen bedeutsam sein. Dies erleichtert das Verschieben, da die Anzahl der Iterationen nicht bekannt sein muss. Realisieren lässt sich diese Erweiterung durch Ergänzung des Symbols für die Verschieberichtung um den Parameter ' $<_b$ ' (b - begin), zum Verschieben eines Elements an den Anfang, und den Parameter ' $>_e$ ' (e - end), für das Verschieben an das Ende der Liste von Geschwister-Elementen.

Beispielhaft wird das Element e_2 an das Ende der Liste der Geschwister-Elemente verschoben.

$$G_6 = \mathit{shift}_{e_2, >_e}(G_5)$$

$$G_6 = \{e_1, e_4\{e_7\}, e_2\}$$

Das Verschieben von Elementen ist nur dann möglich, wenn in Verschieberichtung mindestens ein weiteres Geschwister-Element existiert. Bei Mehrfachverschiebungen müssen entsprechend der Anzahl der Iterationen Geschwister-Elemente existieren. Andernfalls ist das Verschieben entweder nicht zuzulassen oder kann nur bedingt umgesetzt werden. Innerhalb einer Anwendung ist im Falle der bedingten Umsetzung der Nutzer über diese Tatsache zu informieren.

Gridelement Umzug (*move*): Zum Verschieben eines Gridelements in eine andere Ebene wird der komplexe Operator *move* benötigt, der sich auf die Kombination der Operationen *del* und *ins* zurückführen lässt. Es gilt zu beachten, dass das Löschen jeweils vor dem Einfügen stattfinden muss, da andernfalls die Eindeutigkeit der Elemente nicht gewährleistet werden kann. Ferner muss sichergestellt sein, dass immer beide Operationen durchgeführt werden, beispielsweise durch Einbettung in eine Transaktion. Ergänzt wird der *move* Operator um den Identifikator des Elements, das verschoben werden soll, und den Identifikator des Elements unterhalb dessen das zu verschiebende Element einzufügen ist. Soll ein Element in die Rootebene verschoben werden, entfällt die Angabe des Zielelements. Keinesfalls darf ein zu verschiebendes Element zugleich als Zielelement angegeben werden, da sich ein Grid nicht selbst enthalten darf (Gl. 4.3) und zudem durch das Verschieben die Referenz verloren gehen würde. Allgemein lässt sich der *move* Operator beschreiben als:

$$\mathit{move}_{element_{e.id} \rightarrow (target_element_{te.id})}(G_{G.id})$$

$element_{e.id} \neq target_element_{te.id}$ In folgendem Beispiel wird das Element e_1 als Kindelement in das Element e_2 verschoben. Da e_2 über keine Kindelemente verfügt, wird die Kindebene mit dem Einfügen neu geschaffen.

$$G_7 = move_{e_1 \rightarrow e_2}(G_4)$$

$$G_7 = \{e_4\{e_7\}, e_2\{e_1\}\}$$

Entspricht das *target_element* dem Elternelement des zu verschiebenden Elements ($parent_element(e_x) = e_y$), so entsteht der Sonderfall der Verschiebung des Elements innerhalb der Ebene. Aufgrund der Rückföhrbarkeit auf die Operatoren *ins* und *del*, wird in diesem Fall das Element am Ende der Liste der Geschwister-Elemente eingefügt. Somit entspricht unter solchen Umstünden die Operation $move_{e_x \rightarrow e_y}(G_z)$ der Operation $shift_{e_x, > e}(G_z)$.

Grid(element) Einbettung (*embed*): Soll ein Grid G_x bzw. sollen dessen Elemente $e_i \in G_x$ in ein Grid G_y integriert werden, wobei $G_x \neq G_y$ gilt, ist der Operator *embed* zu verwenden. Dieser erfordert mindestens das Angeben des Grids G_x , welches integriert werden soll, als auch die Benennung des integrierenden Grids G_y . Ohne weitere Angaben werden alle Elemente des Grids G_x auf dem Rootlevel von G_y als neue Elemente eingefügt. Da zu jedem Zeitpunkt die Eindeutigkeit der Gridelemente gewährleistet sein muss, sind etwaige Namenskonflikte vor der Einbettung durch Umbenennung aufzulösen. Das Ergänzen des zu integrierenden Grids G_x , um die Angabe eines Elements $e_i \in G_x$ bewirkt, dass lediglich dieses Element e_i einschließlich seiner Kindelemente in G_y eingefügt wird. Ferner lässt sich durch das Benennen eines Zielelements $e_j \in G_y$ erreichen, dass alle Elemente des Grids G_x oder das konkrete Element e_i als Kindelement von e_j in G_y eingefügt werden. Allgemein lässt sich das Einbetten von Grids beschreiben als:

$$embed_{G_{from_id}, (element_{e_id}) \rightarrow (target_element_{te_id})}(G_{to_id})$$

Beispielhaft sollen folgend alle Elemente des Grids G_8 in das Grid G_7 unterhalb des Elements e_2 eingebettet werden. Da die Elemente des Grids G_8 bei einer direkten Integration die Eindeutigkeitsforderung verletzen würden, sind diese innerhalb eines temporären G'_8 vor dem Einbetten derart umzubenednen, dass sie sowohl innerhalb von G_7 als auch G'_8 eindeutig sind, z.B. $e_1 \rightarrow e_8$ und $e_2 \rightarrow e_9$. In diesem Fall entspricht somit das Einbetten der Elements somit dem Einfügen eines neuen Elements

$$G_7 = \{e_4\{e_7\}, e_2\{e_1\}\}$$

$$G_8 = \{e_1, e_2\}$$

$$G_9 = embed_{G_8 \rightarrow e_2}(G_7)$$

mit den transparenten Zwischenschritten:

$$\forall e_i, e_j \in \mathbb{N}, e_i \in G_7 \wedge G_8, e_j \notin G_7 \vee G_8 : \varrho_{e_i \rightarrow e_j}(G_8)$$

$$G'_8 = \varrho_{e_1 \rightarrow e_8, e_2 \rightarrow e_9}(G_8)$$

$$G'_8 = \{e_8, e_9\}$$

reale Einbettung nach der Umbenennung:

$$G_9 = \text{embed}_{G'_8 \rightarrow e_2}(G_7)$$

$$G_9 = \{e_4\{e_7\}, e_2\{e_1, e_8, e_9\}\}$$

Beim Einbetten eines Grids G_x wird keine Referenz, sondern eine Kopie der Elemente in das integrierende Grid G_y eingefügt, wodurch sich nachträgliche Änderungen am Grid G_x nicht auf das Grid G_y auswirken. Referentielle Einbettungen werden durch das Verwenden von Gridinstanzen möglich, die später in diesem Kapitel Erwähnung finden.

Gridelement Gruppen (*group* / *ungroup*): Innerhalb komplexer Grids kann es hilfreich sein, zusammengehörige Teilbereiche gesondert kenntlich zu machen, da nicht in jedem Fall eine offensichtliche Element-Hierarchie definiert oder die Aufgabe und inhaltliche Zugehörigkeit eines Elements erkennbar sein muss. Allgemein besteht zwar die Möglichkeit, zusammengehörige Elemente jeweils einem Gridelement ohne explizite Aufgabe unterzuordnen, jedoch kann dies die Verständlichkeit des Grids sogar erschweren, da dann nicht mehr entscheidbar ist, ob ein Gridelement nur der Zusammenführung logisch zusammengehöriger Gridelemente dienen soll oder wichtiger, struktureller Bestandteil des Grids ist. Zur Vermeidung dieses Problems wird der *group* Operator eingeführt, der eine logische Gruppierung zusammengehöriger Elemente erlaubt. Eine Gruppe kann Gridelemente als auch weitere Gruppen enthalten und ist selbst kein die Präsentation betreffendes Element.

Das Erstellen einer Gruppe innerhalb eines Grids erfordert neben der Wahl einer eindeutigen Bezeichnung für die Gruppe die Angabe der Gridelemente, die Bestandteil dieser Gruppe sein sollen. Die Bezeichnung sollte aussagekräftig sein und sich von jener für Grids, Gridinstanzen, Gridelementen und Operatoren hinreichend unterscheiden, um Fehldeutungen zu vermeiden. Ist ein Element e_i Bestandteil der Gruppe, so sind auch Kindelemente von e_i Bestandteil der Gruppe. Allgemein lässt sich das Erstellen von Gruppen wie folgt beschreiben:

$$\text{group}_{\text{group_name}' : e_{e_id_1}, \dots, e_{e_id_z}}(G_{G_id})$$

Das Hinzufügen weiterer Elemente zu einer bestehenden Gruppe wird ebenfalls mittels obigem Term erreicht. In diesem Fall ist eine bestehende Gruppenbezeichnung zu verwenden. Beispielhaft sollen die Gruppen *Grp1* und *Grp2* innerhalb des Grids G_9 eingefügt werden. *Grp1* soll Gruppe des Elements e_4 sein, während die Gruppe *Grp2* die Elemente e_1, e_8 umfasst. Ergänzend wird dieser Gruppe das Element e_9 hinzugefügt.

$$G_{10} = \text{group}_{\text{Grp1}' : e_4}(G_9)$$

$$G_{10} = \{\text{Grp1}(e_4\{e_7\}), e_2\{e_1, e_8, e_9\}\}$$

$$G_{11} = \text{group}_{\text{Grp2}' : e_1, e_8}(G_{10})$$

$$G_{12} = \text{group}_{\text{Grp2}' : e_9}(G_{11})$$

$$G_{12} = \{Grp1(e_4\{e_7\}), e_2\{Grp2(e_1, e_8, e_9)\}\}$$

Gruppen dürfen nur Elemente eines einzigen Hierarchie-Levels und des gleichen Zweiges im Elementbaum umfassen. Folglich dürfen sowohl Element e_7 (Zweig-Konflikt) als auch Element e_4 (Hierarchie-Level-Konflikt) nicht zusätzlich der Gruppe $Grp2$ zugeordnet werden. Wird jedoch der Gruppe $Grp2$ das übergeordnete Element e_2 , anstelle der einzelnen Kindelemente, zugeordnet, so kann das Element e_4 Bestandteil der Gruppe werden, da dann keiner der obigen Konflikte auftritt. Ein weiteres Problemfeld stellen sich einander überkreuzende Gruppen dar, die innerhalb eines Zweiges und Hierarchie-Levels auftreten können. Grundsätzlich gilt es diese zu vermeiden, um die Übersichtlichkeit der Grids und der Notation nicht zu beeinträchtigen. Typischerweise lassen sich Überschneidungen verhindern, indem Elemente durch Verwendung des *shift* Operators vor dem Gruppieren adäquat geordnet werden. Sind Überschneidungen nicht vermeidbar oder zwingend erforderlich, so ist die Notation zu erweitern. Unter anderem kann zugelassen werden, dass eine Gruppenbezeichnung mehrfach innerhalb des Grid auftreten darf, um die Zusammengehörigkeit einzelner Gruppenelemente kenntlich zu machen.

Beispielhaft werden die Gruppen $Grp3$ und $Grp4$ eingeführt. Während der Gruppe $Grp3$ die Elemente e_1 und e_8 zugeordnet werden, sollen e_8 und e_9 Elemente der Gruppe $Grp4$ sein, wodurch e_8 Element beider Gruppen ist. Da sich Gruppe $Grp4$ nicht ohne Überschneidungen einführen lässt, erfolgt das Aufspalten der Gruppe.

$$G_{13} = group'_{Grp3':e_1,e_8}(G_{12})$$

$$G_{13} = \{Grp1(e_4\{e_7\}), e_2\{Grp2(Grp3(e_1, e_8), e_9)\}\}$$

$$G_{14} = group'_{Grp4':e_8,e_9}(G_{13})$$

$$G_{14} = \{Grp1(e_4\{e_7\}), e_2\{Grp2(Grp3(e_1, Grp4(e_8)), Grp4(e_9))\}\}$$

Entsprechend dem Hinzufügen von Gridelementen zu einer Gruppe, ist es mit dem *ungroup* Operator möglich, Elemente von dieser Liste zu entfernen. Soll eine bestehende Gruppe aufgelöst werden, so sind mittels des *ungroup* Operators alle Elemente der Gruppe von dieser zu lösen. Mit dem Entfernen des letzten Elements von einer Gruppe wird diese gelöscht, da eine Gruppe ohne Elemente nicht der strukturierenden Funktion gerecht wird. Analog dem Hinzufügen von Gridelementen zu einer Gruppe sind die Anforderungen für das Löschen. Neben dem Gruppennamen, der innerhalb des Grids $G_{G.id}$ vorhanden sein muss, werden alle Gridelemente $e_{e.id} \in G_{G.id}$ angegeben, die Bestandteile der Gruppe sind und entfernt werden sollen. Dabei ist unerheblich, ob die Gruppe zur Vermeidung von Überschneidungen aus mehreren Teilen besteht. Allgemein lässt sich das Entfernen von Elementen von Gruppen beschreiben durch:

$$ungroup'_{group_name':e_{e.id_1}, \dots, e_{e.id_z}}(G_{G.id})$$

Da das Entfernen von Elementen einer Gruppe nicht die Elementreihenfolge beeinflussen soll, ist gegebenenfalls eine Gruppe zu trennen, wenn einzelnen Elemente von dieser entfernt werden. Umgekehrt kann durch das Entfernen auch das Zusammenführen von

Gruppenteilen möglich werden. So kann durch das Entfernen der Gruppe *Grp3* bzw. durch das Entfernen ihrer Elemente e_1 und e_8 , eine Zusammenlegung der Gruppe *Grp4* erfolgen.

$$\begin{aligned} G_{15} &= \text{ungroup}_{Grp3':e_1,e_8}(G_{14}) \\ G_{15} &= \{Grp1(e_4\{e_7\}), e_2\{Grp2(e_1, Grp4(e_8, e_9))\}\} \end{aligned}$$

Durch das Entfernen von e_8 aus der Gruppe *Grp2* wird hingegen ein Splitting erforderlich.

$$\begin{aligned} G_{16} &= \text{ungroup}_{Grp2':e_8}(G_{15}) \\ G_{16} &= \{Grp1(e_4\{e_7\}), e_2\{Grp2(e_1), Grp4(e_8, Grp2(e_9))\}\} \end{aligned}$$

Behandlung von Gridinstanzen: Gridinstanzen werden grundsätzlich wie Gridelemente behandelt, sodass für das Einfügen, Löschen, Verschieben, Umziehen, Einbetten und Gruppieren von Gridinstanzen die gleichen Regeln gelten, wie für die oben eingeführten Gridelemente. Gridinstanzen stellen jedoch Referenzen zu Grids dar, wodurch sich Änderungen an den Grids unmittelbar auf alle verwendeten Gridinstanzen auswirken. Dies gilt es bei der Verwendung von Gridinstanzen als auch bei Änderungen an den Grids zu beachten. Ferner sind die Gridelemente der Gridinstanzen für das einbettende Grid und anderen Instanzen des Grids transparent, sodass keine Namenskonflikte auftreten und Umbenennungen nicht erforderlich sind.

Beispielhaft sollen folgend 2 Instanzen des Grids G_8 in das Grid G_9 eingefügt werden, auf dem Rootlevel und innerhalb von e_4 .

$$\begin{aligned} G_{17} &= \text{ins}_{G_{8_1} \curvearrowright e_4}(G_9) \\ G_{17} &= \{e_4\{e_7, G_{8_1}\}, e_2\{e_1, e_8, e_9\}\} \end{aligned}$$

$$\begin{aligned} G_{18} &= \text{ins}_{G_{8_2}}(G_{17}) \\ G_{18} &= \{e_4\{e_7, G_{8_1}\}, e_2\{e_1, e_8, e_9, G_{8_2}\}\} \end{aligned}$$

Bei vollständiger Auflösung der Instanzen ergibt sich die folgende Gridbeschreibung:

$$G_{18} = \{e_4\{e_7, G_{8_1}\{e_1, e_2\}\}, e_2\{e_1, e_8, e_9, G_{8_2}\{e_1, e_2\}\}\}$$

Strukturelle Änderungen können an den Instanzen nicht vorgenommen werden, sondern nur am jeweiligen Ursprungsgrid. Lokale Änderungen oder Individualisierungen sind daher nur möglich indem eine Instanz von ihrem Ursprungsgrid gelöst wird und anschließend entweder in einem neuen Grid mündet oder aber die Gridelemente des Grids mittel *embed* Operator in ein übergeordnetes Vatergrid integriert werden.

Eine weitere Flexibilisierung des Einsatzes von Gridinstanzen ist denkbar, indem diesen ergänzend lokal gültige Gridelemente oder andere Gridinstanzen hinzugefügt werden können. Darüber hinaus ließen sich Elemente des Grids benennen, die als gelöscht innerhalb der Gridinstanz betrachtet werden sollen. Dies erhöht den Grad der Flexibilität

birgt jedoch die Gefahr von Inkonsistenzen bei Änderungen an den zugrundeliegenden Grids. Insbesondere bei umfänglicher Nutzung eines Grids durch Gridinstanzen ist diese Problematik zu erwarten. Werden zudem nur die Veränderungen gespeichert, treten Organisationsprobleme auf, da die Benutzbarkeit des *shift* Operators nicht gegeben ist. Abhilfe kann die Einführung einer Versionierung der Gridentwicklung schaffen, so dass sich Änderungen am Grid nicht direkt auf die Gridinstanzen auswirken. Diese Arbeit wird auf eine derartige Erweiterung verzichten und falls nötig mittels *embed* Operator die Elemente eines Grids einbetten, um Individualisierungen vornehmen zu können.

4.1.4 Grid Erweiterung - (grid cluster)

Grids ermöglichen, entsprechend ihrer bisherigen Einführung, die Beschreibung der allgemeinen Aufteilung des Darstellungsbereichs in einer konkreten Situation. Hängt die Eignung oder Positionierung eines Grids jedoch von einem anderen Grid und der Szene innerhalb der Story ab, so stößt der bisherige Ansatz an Grenzen. Insbesondere dann, wenn sich die Entwicklung der Handlung dynamisch entscheiden kann und die Präsentation diesen Verlauf dokumentieren soll, lässt sich selten ein geeignetes statisches Grid entwickeln. Die begrenzende Größe ist dabei nicht allein die visuelle Darstellung, da, je nach Erforderlichkeit, mehrere Darstellungsvarianten entwickelt werden könnten, einschließlich der Regeln für deren Anwendung. Die Tatsache, dass diese Herangehensweise nicht den logischen Zusammenhang der Grids beschreiben kann, wiegt weit schwerer. Separiert betrachtet, lässt sich zwar feststellen, dass auch auf der Story-Ebene ein solcher Zusammenhang definiert werden kann, allerdings wird die Story-Spezifikation rasch unbeherrschbar, sobald zugleich eine Vielzahl von Darstellungsvarianten und Situationen behandelt werden muss. Aus diesem Grund soll die Gridentwicklung um die Fähigkeit der Clusterbildung erweitert werden. Anders als die Grid-Bündel dokumentieren Grid-Cluster nicht die Ähnlichkeit von Grids zur Herausbildung und Nutzung von Alternativen, sondern definieren welche Grids, in welcher Reihenfolge und in welcher Art und Weise verbunden werden können. Dies befähigt Grid-Cluster zum expliziten spezifizieren von Handlungsverläufen, um Präsentationsanforderungen, wie dargestellt in Abbildung 4.4, besser gerecht werden zu können.

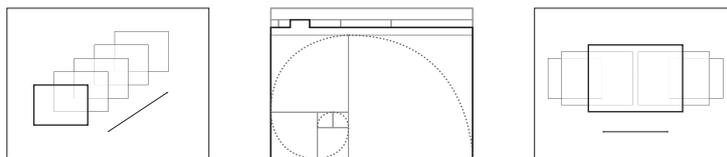


Abbildung 4.4: Evolutions Grids

Die Grids eines solchen Grid-Clusters erlauben infolgedessen nicht nur das Betrachten der Handlung zum aktuellen Zeitpunkt, sondern schaffen Raum für Ausschnitte aus Vergangenheit und/oder Zukunft. Das Cluster definiert dabei nicht nur die Anordnung der Grids, sondern auch die zeitlichen Blickrichtungsmöglichkeiten. So lässt sich beispielsweise mit einem Grid-Cluster (Abb. 4.4) entweder nur in die Vergangenheit blicken

(Abbildung links) oder aber zu gleichen Teilen in die Vergangenheit und Zukunft (Abbildung rechts).

Allgemein lassen sich Grid-Cluster als eine Liste von Grids $G_c = [G_x | x \in \mathbb{N}]$ im Handlungsraum definieren. Für jedes Grid des Clusters muss der absolute oder relative Zeitpunkt der Sicht festgelegt werden. Dies ist auch implizit möglich, jedoch darf jedes Grid des Handlungsraums nur einer einzigen Sicht zugeordnet werden. Subgrids müssen nicht zwingend der zeitlichen Sicht des Elterngrids folgen. Bei der Definition von relativen Sichten, ist zwingend eine Referenzsicht anzugeben. Jedes Grid kann mehrfach als Referenzsicht fungieren, um beispielsweise unterschiedliche zeitliche Entwicklungen in die Vergangenheit und Zukunft definieren zu können. Soll der zeitliche Abstand der Grids eines Grid-Clusters rhythmischen Strukturen folgen, so ist zusätzlich zum Referenzzeitpunkt eines Referenzgrids die Reihenfolge der Grids zu definieren, deren Sicht durch die beschriebene Folge definiert wird. Jedes Grid darf nur ein einziges Mal als Nicht-Referenzgrid auftreten. Grid-Cluster können in ein übergeordnetes Grid aber auch in andere Grid-Cluster eingebettet werden, um weitere Flexibilität zu ermöglichen.

4.1.5 Ausblick

Die bisher eingeführten und gezeigten Grids basieren auf den Überlegungen der klassischen Grid-Erstellung durch Teilung einer Grundfläche in Rechteckflächen mittels Gridpunkten. Obwohl sich diese Teilungsform bislang für die überwiegende Mehrheit zu präsentierender Inhalte eignet, ist die Eignung im Allgemeinen nicht zwingend gegeben. Folglich besteht kein Zwang, ausschließlich Rechteckflächen zuzulassen, da beispielsweise auch eine Parkettierung mit Hilfe von Sechsecken ausreichend Orientierung bieten kann. Im Allgemeinen ist bei der Wahl der Parkettierung eines Raumes auch die Art des darzustellenden Inhalts sowie die gewünschte Erschließung zu berücksichtigen, unter Beachtung der Prinzipien der visuellen Kommunikation.

Grids von Weboberflächen werden in der Regel für die Ebene entwickelt, da die überwiegende Zahl verfügbarer Präsentationssysteme lediglich zur 2D-Darstellung fähig ist. In Abhängigkeit von der Dimension des Gestaltungsbereichs kann jedoch allgemein zwischen 2-dimensionalen und 3-dimensionalen Grids unterschieden werden. Mangels geeigneter Präsentationssysteme, ist die direkte, räumliche Darstellung oft nicht möglich. Räumliche Tiefe kann den Nutzern jedoch durch den gezielten Einsatz von Gestaltungselementen und -mitteln suggeriert werden. Hierbei gilt es jedoch Aufwand und Nutzen genau abzuwägen, da der Aufwand von ad-hoc Transformationen sehr groß sein kann.

4.2 Templates

Als Templates werden oft Grids verstanden, die zu einer vollständigen Layoutvorlage fortentwickelt wurden. Derartige Templates setzen gezielt Gestaltungselemente und Gestaltungsmittel [33, 41] ein und stellen eine konkrete visuelle Ausprägung ohne Inhalte dar. Sie definieren somit Abmessungen und Proportionen, aber auch das mögliche interaktive Verhalten. Durch Anreichern dieser Templates mit Inhalten entstehen gestaltete

Informationen, die dem Stil des im Template definierten Ambiente [34, 41] folgen. Im Ergebnis stellt der Template-Ansatz eine einfache und geeignete Lösung dar, um Inhalte weniger, ausgewählter Inhaltstypen zu präsentieren und globale Layout-Änderungen zu ermöglichen. Innerhalb dieses definierten Rahmens kann zudem die jeweils korrekte Darstellung der Inhalte sichergestellt werden.

Templates werden nur selten modular und überwiegend als monolithische Komponente entwickelt. Dies ermöglicht eine schnelle Entwicklung und verursacht typischerweise wenige, initiale Umsetzungsprobleme. Insbesondere für die Entwicklung von Prototypen ist dies von Vorteil. Innerhalb des Entwicklungsprozesses von WIS sind Templates Bestandteil des Implementationslevels [65], unter anderem durch die Bindung an konkrete Auszeichnungssprachen. Der Tribut der einfachen Entwicklung ist der weitgehende Verlust von Flexibilität und Adaptivität. Bei bestehenden Anwendungen lassen sich zwar kleinere Anpassungen und Ergänzungen der Templates mit vertretbarem Aufwand durchführen, nicht jedoch umfangreiche Änderungen, Individualisierungsvorhaben oder vollständige Neuausrichtungen der Gestaltung. Typischerweise folgt daher einer Anpassungs- und Erweiterungsphase von Templates stets die vollständige Neuentwicklung der Gestaltung.

Sollen Teile der Gestaltung wiederverwendbar innerhalb der gleichen Anwendung sein, ist die Spezifikation eines einzigen, globalen Templates ungeeignet. Content Management Systeme, z.B. Typo3 [70], unterstützen daher im Gebrauch mehrerer Templates und bieten den Nutzern Funktionalität, um Template-Hierarchien bilden als auch verwalten zu können. Dies bietet Flexibilität und neben der leichten Wiederverwendbarkeit auch die Möglichkeit der Integration von Templates in Andere. In der Folge erlaubt dies weniger und weniger komplexe Templates und begünstigt die Wartbarkeit und Pflege.

Die erleichterte Wiederverwendbarkeit hilft nur sehr eingeschränkt bei der Personalisierung der Gestaltung. Eine individualisierte Darstellung, die über die Auswahl weniger, vorkonfigurierter Layout-Varianten hinausgeht, ist mit dem Template-Ansatz wegen des erforderlichen hohen Aufwands selten umsetzbar und praktikabel.

Im Zuge einer Neuausrichtung der Gestaltung oder dem Wechsel des CMS, wird meist deutlich erkennbar, dass gerade die Implementierungsdetails von Templates deren direkte Wiederverwendbarkeit verhindern. Ferner stellt man beim Einsatz vieler integrierbarer Templates fest, dass zwar deren korrekte Funktion mit vertretbarem Aufwand sichergestellt werden kann, nicht aber deren Darstellbarkeit in allen Situationen auf diesem niedrigen Abstraktionsniveau vollständig testbar ist. Die Konsequenz großer Portale besteht schließlich allzu oft in der Beschränkung eigener Tests auf für die Anwendung essentielle Bereiche und der Verlagerung von weitergehenden Tests auf die Benutzer im Rahmen offener Beta-Phasen.

Während hoher Aufwand bei Neuausrichtungen oder einem System-Wechsel zumeist erwartet wird, überrascht dieser umso mehr, wenn er bei kleineren Änderungsvorhaben entsteht. Wird eine Anwendung schrittweise weiterentwickelt, können die jeweils kleinen Evolutionsschritte die Abhängigkeiten zwischen Templates stetig vergrößern und damit die Flexibilität einschränken. Häufiger sind sogar Abhängigkeiten zwischen Templates und ihren Inhalten, durch Annotation der Inhalte mit Template-spezifischen Darstellungsdetails auf dem Implementationslevel. Dem Content-Autor kann auf diesem Weg

mehr Flexibilität geboten werden und auch der Einsatz plattformbezogener Tools wird möglich, z.B. JavaScript-basierte Rich-Text-Editoren. Aber gerade dieses Entgegenkommen erschwert spätere Änderungen am Template erheblich und macht sie teilweise sogar unmöglich. Bei hinreichender Evolution und intensiver Verwendung von Templates entfällt daher oft ein Großteil des Änderungsaufwand auf die Sicherstellung der Darstellbarkeit.

Abhängig vom Typ der Anwendung und deren Umfang kann eine Template-Lösung auch fortgesetzt geringen Aufwand verursachen. Unter anderem weisen sogenannte *Identity sites* einen nur geringen Umfang auf, erfordern nur wenige zu unterstützende Contentobjekttypen und sind damit für Template-Lösungen prädestiniert.

Die Screenography verfolgt das Ziel, dass nicht der Gestalter oder Content-Produzent definieren muss, welche Layout-Komponenten zusammengefügt und präsentiert werden, sondern sich dies aus den Bedürfnissen ableiten lässt. Der Gestalter definiert unabhängig von Inhalten die Fähigkeiten der von ihm entwickelten Layout-Komponenten und grundlegende Kompositionsregeln. Für die Inhalte werden vom Content-Manager jeweils grundlegende Bedürfnisse für ihre Darstellung annotiert. Im Ergebnis kann das Informationssystem über die tatsächliche Komposition und Verwendung anhand der Annotationen und der jeweiligen Situation entscheiden. Durch eine Trennung der Bedürfnisse von den Darstellungsmöglichkeiten wird die Prüfung der Darstellbarkeit für bestehende aber auch neue Inhaltstypen möglich. Auswirkungen geplanter Änderungen am Layout lassen sich daher besser kontrollieren. Deutlich erschwert wird mit dieser Vorgehensweise punktuell eingreifen in die Gestaltung durch den Künstler, bleibt aber über die Layout-Definitionen weiterhin möglich.

Insbesondere dann, wenn eine Vielzahl von Gerätetypen unterstützt werden soll, ist es einfacher, die Grenzen und Möglichkeiten des jeweiligen Präsentationsmediums zu definieren, als für jede Präsentation separat geeignete Layout-Varianten zu entwickeln und vor allem weiterzuentwickeln. Nur wenn wenige oder keine Gemeinsamkeiten existieren, kann nicht von der Wiederverwendbarkeit einzelner Layout-Komponenten profitiert werden.

Aufgrund der Vielzahl von Einflussfaktoren kann die Layout-Entwicklung sehr aufwendig werden und überfordert die meisten Entwickler, wenn diese ihre Komponenten vollständig eigenständig annotieren sollen. Da sich die überwiegende Mehrheit von Websites wenigen Clustern gleicher oder ähnlicher Ziele zuordnen lässt, können Gemeinsamkeiten aber als Muster (Pattern) definiert und eine Bibliothek zum Entwickeln von Standard-Anwendungen aufgebaut werden.

4.3 Patterns

Infolge der oft gleichartigen Ziele und Herausforderungen, treten bei der Entwicklung von Webinformationssystemen Probleme wiederholt auf. Mehrfachentwicklungen lassen sich gerade dann vermeiden, wenn eine geeignete Lösung nicht auf dem Implementationslevel entwickelt wird, da es nur selten ein exakt gleiches Problem nochmals zu lösen gilt. Durch die Wahl eines höheren Abstraktionsniveaus, beispielsweise dem konzeptionellen Level,

können allgemeinere Lösungen für Problemstellungen gefunden werden, die anstelle der speziellen Lösung ein Muster (*Pattern*) formulieren, anhand dessen eine konkrete Lösung abgeleitet werden kann.

4.3.1 Verwandte Arbeiten

Die Möglichkeit Muster zu erkennen, zu entwickeln und anzuwenden ist eine ureigene Fähigkeit des Menschen, die überwiegend intuitiv angewandt wird. Das explizite Entwickeln von Mustern als Lösungsschablone für Problemstellungen wird vor allem mit Christopher Alexander verbunden, der Entwurfsmuster (*design patterns*) [1] einführte, um Problemfelder der Architektur mittels universeller Konzepte beschreiben zu können. Das Ziel dieser Muster ist es, gleiche oder in ähnlicher Weise wiederkehrende Probleme einfacher lösbar zu machen.

”Each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice” - Christopher Alexander [1].

Die sog. Viererbande [15] griff diesen Ansatz auf und adaptierte ihn für Problemstellungen der objektorientierten Softwareentwicklung. Auch die Entwicklung von Webinformationssystemen kann von der Wiederverwendung bekannter Konzepte profitieren, indem durch die Muster konzeptionelle Layoutanforderungen und -restriktionen beschrieben werden. Folgend werden solche Muster als *Layout Patterns* bezeichnet. Durch die Abstraktion der *Layout Patterns* ist eine Gestaltungslösung nicht mehr untrennbar mit einer konkreten Visualisierung verbunden. Diese Flexibilität ist Voraussetzung um mehrfaches Wiederverwenden ohne die Probleme von Template-Lösungen realisieren zu können.

Verschiedenste HCI Ansätze [72, 45, 60, 73] haben Regeln eingeführt, die es erlauben, ansprechende und gut bedienbare Webauftritte zu entwickeln. Typischerweise werden Aussagen darüber getroffen, wie Bestandteile des Layout arrangiert werden sollten und welche Kriterien erfüllt sein müssen, um die objektive und subjektive Wahrnehmung zu optimieren. Dabei reicht die Bandbreite von allgemeinen Empfehlungen bis hin zu unmittelbar anwendbaren Lösungen für ein konkretes Medium.

Shneiderman [60] beispielsweise empfiehlt beim Entwurf von Interfaces mehrere grundlegende Regeln zu berücksichtigen, die sogenannten ”eight ’golden’ rules of interface design”. Leider wurden diese Regeln derart allgemein formuliert, dass sie sich ohne weitere Verfeinerung nur zum Überprüfen bestehender Interfaces eignen, nicht jedoch zum Entwickeln solcher. Ferner lassen die Regeln Fehleinschätzungen zu, da sie kontextabhängig wirken und sich Werte, die mit den Regeln in Verbindung stehen, teilweise nur subjektiv festlegen lassen. So ist beispielsweise die nutzbare Kapazität des Kurzzeitgedächtnisses von der Aufmerksamkeit, dem Alter und den Fähigkeiten des Nutzers abhängig, aber auch vom Typ der Anwendung. Darüber hinaus verursachen kleine Informationseinheiten eine geringere Belastung pro Bildschirmausgabe jedoch lange Wege zur Information.

Dies stellt höhere Anforderungen an den Nutzer bzw. dessen Fähigkeit sich zu orientieren als im umgekehrten Fall. Aufgrund der fehlenden Allgemeingültigkeit sind Shneidermans Regeln insbesondere für geübte und talentierte Entwickler hilfreich.

Van Welie [73] entwickelte eine *pattern language*, die von den Beziehungen zwischen speziell entwickelten Layout Patterns profitiert. Im Unterschied zu Shneidermans Ansatz, weisen van Welies Patterns einen hohen Detailierungsgrad auf und werden Geschäftszielorientiert entwickelt. Die Basis der Spezifikation bilden gut orchestrierbare Layoutelemente auf unterschiedlichen Abstraktionsleveln. Die feingranulare, problemorientierte Herangehensweise bietet ein Höchstmaß an Anwendungsbezogenheit. Da die Patterns nicht ausschließlich konzeptionell oder abstrakter entwickelt werden, sondern Implementierungsspezifika aufweisen oder aber diese implizieren, wird jedoch die Wiederverwendbarkeit bestehender Lösungen erschwert. Die problemorientierte Herangehensweise erschwert ferner das Vermeiden und Erkennen gleichartiger Patterns, infolgedessen eine große Anzahl von Patterns und Patterns-Kombinationen entsteht. Überdies steigt mit zunehmender Patternmenge der Aufwand, um geeignete Transitionen zwischen den vorgeschlagenen Abstraktionsschichten zu identifizieren und nicht mehr Transitionen als nötig zu definieren. Allgemein lässt sich die Suche geeigneter Patterns erleichtern, wenn die Klassifikation von Patterns lediglich konzeptionell erfolgt und zwischen den Patterngruppen möglichst geringe Überschneidungen zu erwarten sind. In [55] werden 6 Entwicklungsdimensionen unterschieden (intention, context, story, content, functionality, and presentation), um Entscheidungen bei Anwendungen auf Basis mehrerer Anwendungsdomänen zu erleichtern. Van Welie [73] unterscheidet Patterns hinsichtlich *Context*, *Application needs* und *User needs*. Diese Patterns eignen sich insbesondere, um neue Websites für die bereits definierten Anwendungsbereiche zu erstellen. Sie bieten jedoch keine Unterstützung bei sich ändernden Anforderungen oder den Wechsel des Website-Typs und berücksichtigen nicht den Verlauf bzw. die Historie der Benutzung. Final münden Anforderungsänderungen daher stets in einer Neuentwicklung, anstelle eines alternativen Content-Playout, das auf die geänderte Situation adaptiv reagiert.

Neben den HCI Publikationen existiert eine große Zahl von Publikationen, z.B. Itten [22] und Skopec [61], die den Layoutentwicklungsprozess und Layoutanforderungen aus der Sicht von Künstlern und Designern diskutieren und Entwickler darin unterstützen, welche graphischen Elemente in welchen Situationen zu verwenden sind und wie diese komponiert werden können. Das Fundament dieser Ansätze bilden die Prinzipien der kognitiven Psychologie [37], die implizit als auch explizit Berücksichtigung finden. Die Prinzipien erlauben das Ableiten allgemeiner Regeln der Layoutentwicklung, deren Gewicht sich innerhalb der Anwendung kontextabhängig entfaltet.

4.3.2 Pattern Klassifikation

In [37] werden drei Prinzipien der kognitiven Psychologie unterschieden, das Prinzip der visuellen Kommunikation, das Prinzip der visuellen Wahrnehmung und das Prinzip des visuellen Designs.

Die visuelle Kommunikation umfasst Aspekte der Interaktion und stellt dabei auch die Bedienbarkeit der Anwendung durch den Nutzer sicher, z.B. durch Suchen und Auswahl

einer für das jeweilige Gerät geeigneten Bereitstellungsform. Im Falle wechselseitiger Kommunikation, bei der die Anwendung das Bindeglied der Kommunikation darstellt, ist zudem sicherzustellen, dass sich die Kommunikationspartner auch dann verständigen können, wenn sie auf unterschiedliche Kommunikationsmittel zurückgreifen. Somit sind lokale und globale Fähigkeiten und Restriktionen bei der Wahl der Kommunikationsform zu berücksichtigen und gegebenenfalls geeignete Transformationsformen zu suchen.

Das Prinzip der visuellen Wahrnehmung trägt Sorge dafür, dass sich ein Nutzer innerhalb des Gestaltungsraums orientieren und zielgerichtet interagieren kann. Daher ist vor allem die Anordnung und graphische Gewichtung der Inhalte bedeutsam, durch den Einsatz geeigneter Gestaltungselemente und -mittel [41]. Primäres Ziel ist dabei die Herausbildung eines Gestaltungsrahmens, der als stabiles Grundgerüst nicht nur die Integration verschiedenster Inhalte erlaubt, sondern den Akteuren visuelle Orientierungshilfen bietet. Hierzu sind sowohl Vordergrund-Hintergrund Beziehungen der Inhalte zu berücksichtigen als auch die Wirkung von Farben [34] sowie bestehende Wechselwirkungen zwischen Farben [16]. Eine allgemeingültig optimale Darstellung kann allerdings nicht gefunden werden, da die Wahrnehmung von Inhalten durch Nutzer individuell verschieden ist. Werden jedoch die sensorischen, motorischen und kognitiven Fähigkeiten der Nutzer berücksichtigt, kann zumeist eine eingeschränkt individuelle Darstellungsform gefunden werden, die eine Balance zwischen den Bedürfnissen des Betreibers und denen des Nutzers erlaubt.

Das Prinzip des visuellen Designs bewirkt eine Ausdrucksform durch gezielte Komposition von Gestaltungselementen. Die Grundlage dazu bildet die Gestaltpsychologie, die maßgeblich durch Wertheimer [76], Köhler [27] und Koffka [26] geprägt wurde. Das bewusste Ausnutzen menschlicher Wahrnehmungsfähigkeiten kann die Intuitivität der Kommunikation mit dem System steigern und somit die Benutzung erleichtern.

Auf der Grundlage der Prinzipien der kognitiven Psychologie lässt sich eine Klassifikation erzielen, die sich für die Entwicklung und Zuordnung von *Layout Patterns* eignet. Ohne eine Ergänzung ermöglicht die Klassifikation jedoch nur das Entwickeln von Patterns für zeitinvariante Visualisierungen. Ferner ist es nicht möglich, Visualisierungsmuster zu beschreiben, die sich zur Senkung der Darstellungs- und Benutzungskomplexität aus mehreren Teilvisualisierungen zusammensetzen, soweit sie nicht losgelöst voneinander betrachtet werden können. Um der Präsentations- und Interaktionsdynamik von Webinformationssystemen gerecht zu werden, ist die Klassifikation daher um die Möglichkeiten des visuellen Voranschreitens *work progress* zu erweitern [44]. Dies beinhaltet das geeignete Visualisieren der Weiterentwicklung und der Historie. Abbildung 4.5 veranschaulicht die Klassifikation der Layout Patterns einschließlich ausgewählter Teilbereiche.

4.3.3 Composition patterns

Composition patterns dienen primär der Platzierung und Anreicherung des Gestaltungsraums mit Gestaltungselementen und Contentobjekten. Das gezielte Arrangement von Inhalten greift dabei Aspekte der Ästhetik und Struktur auf und kann, durch Beachtung der Grundlagen der Gestaltpsychologie, eine objektive Darstellungswirkung erzie-

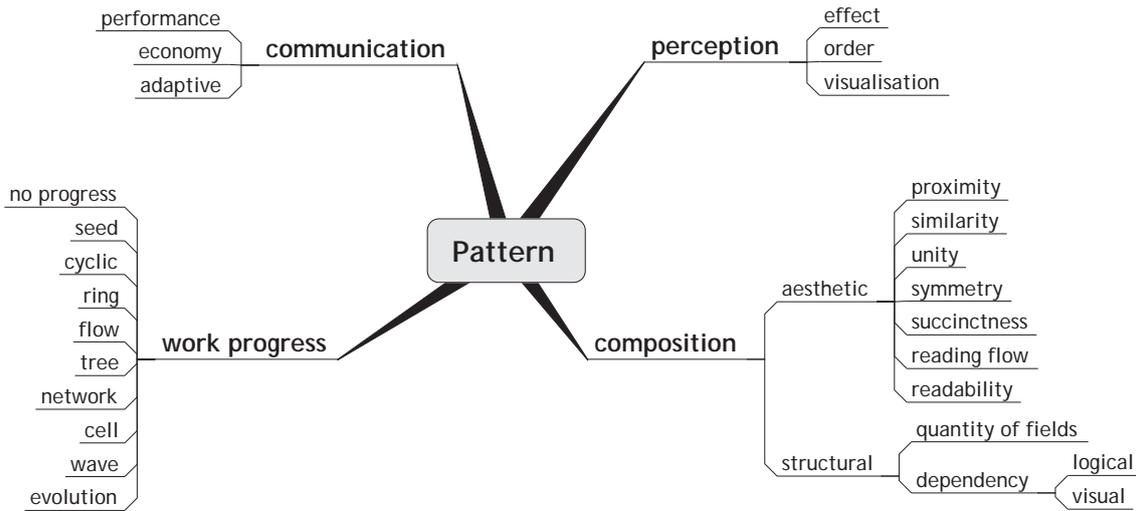


Abbildung 4.5: Pattern Klassifikation

len. Durch die objektive Wirkung ist das Entwickeln universeller Wahrnehmungs- bzw. Darstellungsgrundlagen möglich, die sich später mittels *Perception patterns* individualisieren lassen. Allgemein lässt sich zwischen Ästhetischen und die Struktur betreffenden Kompositionen unterscheiden.

Neben der unmittelbaren Komposition können Composition patterns auch als Meta-Werkzeug für die Komposition existierender Patterns verwendet werden. Typischerweise werden dazu Basis-Patterns schrittweise um weitere Patterns angereichert. Die Aufgabe ergänzender Patterns besteht dabei vor allem in der Erweiterung und Schärfung der bisherigen Ziele. Möglich ist aber auch das Unterdrücken einzelner Ziele, durch Verwendung als Anti-Pattern, die beschreiben, welche Kompositionen nicht zulässig sind. Infolgedessen sind aber auch solche Patterns innerhalb einer Komposition verwendbar, deren Ziele nicht vollständig disjunkt und zudem auch konträr sein können. Die Behandlung von derartigen Konflikten beteiligter Patterns ist für die Realisierung zwingend. Eine Lösungsoption besteht in der Schaffung von Pattern-Hierarchien, die das Gewicht jedes Patterns eindeutig bestimmen. Hilfreich ist dabei auch, eigenständig nutzbare Patterns von jenen Patterns zu unterscheiden, deren Nutzbarkeit von anderen Patterns abhängt. Darüber hinaus ist eine Strategie zur Auflösung von Konflikten erforderlich, falls konträre Pattern-Ziele gleichen Gewichts zu entscheiden sind.

Aesthetic patterns:

Das Ziel ästhetischer Kompositionen ist das Realisieren einer objektiv angemessenen Anordnung der Inhalte. Objektivität ist in diesem Zusammenhang möglich, weil nicht das konkrete Nutzerbedürfnis über die tatsächliche Darstellung entscheidet. Vielmehr handelt es sich um eine Beschreibung eines Anordnungsziels bzw. der erstrebten Form der inhaltlichen Zusammenstellung, um eine gleichartige Wahrnehmung und Wirkung

erzielen zu können, trotz möglicherweise unterschiedlicher Repräsentationen. Wenngleich keine diesbezügliche Beschränkung besteht, orientieren sich diese Patterns maßgeblich an den Gestaltgesetzen Wertheimers [77]. Unter anderem lassen sich die folgenden Kompositionsaspekte unterscheiden:

- Nähe (proximity)
- Ähnlichkeit (similarity)
- Einheitlichkeit (unity)
- Symmetrie (symmetry)
- Knappheit (succinctness)
- Lesefluss (reading flow)
- Lesbarkeit (readability)

Structural patterns:

Die strukturelle Komposition betrifft nicht nur das Festlegen der Anzahl und des Umfangs möglicher Inhalte innerhalb des Gestaltungsraums, sondern auch deren logische Anordnung entsprechend der Story. Ferner können durch Überlagerungen und Kontraste visuelle Abhängigkeiten entstehen. Infolgedessen bestimmen Composition patterns eine ausgewählte Anzahl von Präsentationsbereichen für Inhalte und definieren Regeln, wie diese zusammenzufügen sind. Dabei ist es hilfreich, bereits bestehende Lösungen zu analysieren, mit dem Ziel deren Abhängigkeiten identifizieren, bewerten und aufgreifen zu können. Auf dem Patternlevel besteht das Hauptinteresse an der Detektion allgemeiner Abhängigkeiten und Platzierungsproblemen, während Belange konkreter Inhalte Teil der Spezifikation auf dem Gridlevel sind. Folgend sollen folgende Teilbereiche unterschieden werden:

- Menge / Umfang
- Abhängigkeiten

Die Quantität legt, im Kontext der Komposition, die zur Verfügung stehenden Präsentationsbereiche für Inhalte fest. Durch derartige Patterns soll die Beladung der Container beschränkt werden, da bei WIS selbst die Gestaltungsfläche nicht immer beschränkt ist. Beispielsweise erfährt die Vertikale innerhalb eines Browsers nur selten eine Beschränkung. Grundsätzlich lässt sich die Nichteignung von Präsentationsbereichen gezielt definieren, um Fehlverwendungen von Patterns zu vermeiden, z.B. indem nicht nur die mögliche maximale Beladung definiert wird, sondern auch der Inhaltstyp, der innerhalb eines Präsentationsbereichs verwendet werden darf. Dadurch können sich diese Festlegungen direkt auf die Präsentation der Inhalte auswirken, da Art und Umfang der Inhalte vorgegebene Schranken nicht überschreiten dürfen.

Werden mehrere Präsentationsbereiche innerhalb des Gestaltungsraums geschaffen, treten mehr oder minder starke Abhängigkeiten zwischen diesen auf. Allgemein lässt sich zwischen visuellen und logischen Abhängigkeiten unterscheiden, wobei beide Varianten auf der Grundlage der Story entwickelt werden sollten, um einen Kohärenzverlust zu verhindern. Befindet sich die angestrebte, mögliche und intuitive Bedienung nicht mit der Art der Komposition der Inhalte im Einklang, wird dies die Benutzbarkeit des WIS beeinträchtigen. Visuelle Abhängigkeiten entstehen durch Ähnlichkeiten in der Darstellung und den gezielten Einsatz von Gestaltungsmitteln, sodass Formen, Farben und die Größe von Objekten über die empfundene Verbundenheit entscheiden können. Oft entstehen visuelle Abhängigkeiten bereits durch optische Nähe. Logische Abhängigkeiten betreffen die Art und Anordnung der Inhalte und sollen die Interaktion unterstützen. Patterns, die logische Abhängigkeiten forcieren, sollen die Platzierung der Inhalte intuitiv verständlich machen und ermöglichen das Entwickeln einfach wahrnehmbarer Nutzungspfade. Überdies soll sich die semantische Nähe von zusammengehörigen Inhalten und Funktionen auch visuell widerspiegeln, um den Bewegungsfluss des Nutzers optimieren und erforderliche Distanzen reduzieren zu können. Dies ist erforderlich, da neben dem eigentlichen Inhalt oft weitere Informationen zugänglich gemacht werden, die den Inhalt ergänzen können, mitunter aber auch vollkommen losgelöst von diesem sind. Folgend werden Beispiele struktureller Abhängigkeiten diskutiert.

BEISPIEL LOGISCHER ABHÄNGIGKEIT:

Exemplarisch soll eine Gestaltungsfläche Darstellungsbereiche für Navigation, Inhalt und Begleitinformationen der Anwendung vorhalten. Die generelle Darstellbarkeit lässt sich sicherstellen, sobald die Anzahl geeigneter Teilflächen innerhalb der Gestaltungsfläche mindestens der Anzahl der zu präsentierenden Inhaltstypen entspricht. Die Verwendung einer Instanz eines 3-spaltigen Grids, wie dargestellt in Abbildung 4.6, ist in diesem Falle möglich. Da die Teilflächen des gewählten Grids keine Inhaltstypeneinschränkung aufweisen, entstehen mehrere Darstellungsoptionen. Eine typische Zuordnung wird in Abbildung 4.6 gezeigt.

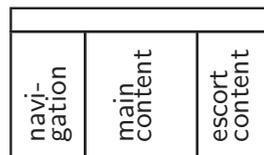


Abbildung 4.6: 3-Spalten Grid - geeignetes Zuordnungsmuster

Andere Zuordnungsmuster sind realisierbar, jedoch ist aufgrund der Semantik nicht jede Zuordnung sinnvoll, wie in Abbildung 4.7 dargestellt. Der Aufwand zur Ermittlung aller geeigneter Zuordnungsvarianten für alle Grids ist aufgrund der Unbeschränktheit der Gridvielfalt typischerweise zu hoch. Ferner beeinflusst die Leserichtung die Eignung der Zuordnung. Dennoch kann die Entscheidungsvielfalt begrenzt werden, wenn Abhängigkeiten bekannt sind, sodass mindestens ungeeignete Zuordnungen entdeckt werden. Zudem können diese Abhängigkeiten als Regeln genutzt werden und somit eine begrenzte Menge geeigneter Lösungsmöglichkeiten hervorbringen.

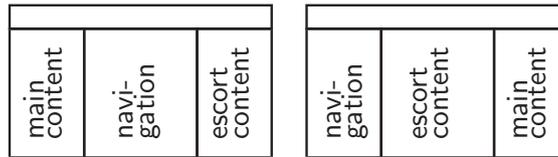


Abbildung 4.7: 3-Spalten Grid - ungeeignetes Zuordnungsmuster

Innerhalb des Beispiels lassen sich zwei Abhängigkeiten entdecken. Die erste Bindung existiert zwischen dem Inhalt und der Navigation. Da die Navigation über den präsentierten Inhalt entscheidet, sollte diese in Wahrnehmungsrichtung vor dem Inhalt wahrgenommen werden und in Benutzungsrichtung früher erreichbar sein. Aufgrund der direkten Abhängigkeit ist die örtliche Nähe der Bereiche meist zwingend für eine intuitive Benutzung. Allgemein ist dies jedoch in Abhängigkeit von den Präsentationsmöglichkeiten zu entscheiden. Befinden sich diese Bereiche nicht in unmittelbarer Nachbarschaft bzw. werden sie durch andere Bereiche visuell getrennt, kann die Kohärenzwirkung verloren gehen.

Die zweite Bindung im Beispiel besteht zwischen dem Inhalt und den Begleitinformationen, weshalb der Inhalt zeitlich vor den Begleitinformationen wahrgenommen werden sollte. Da es sich bei der Begleitinformation um eine optionale Ergänzung handelt, ist die Bindung weniger stark als zwischen Content und Funktionalität und kann daher eine visuelle Trennung hinnehmbar machen. Die existierenden Bindungen lassen sich zu einer Einheit verbinden, um die Kohärenzwirkung zu stärken. Bei einer späteren Verfeinerung kann auch die Leserichtung berücksichtigt werden, um die möglichen Optionen für eine konkrete Anwendung zu reduzieren. Die Leserichtung gehört jedoch nicht zu den Composition patterns, da diese vom Nutzer und der gewählten Sprache abhängig ist und Composition Patterns Anwender-unabhängige Regeln der Komposition beschreiben. Werden die Regeln auf ein 3-Spaltengrid angewandt, ergibt eine Überprüfung der Eignung lediglich zwei Ergebnisse, bei denen beide Regeln eingehalten werden können. Das erste Ergebnis entspricht der Darstellung in Abbildung 4.6, während das zweite Ergebnis, der umgekehrten Zuordnung entspricht. Die übrigen Varianten ermöglichen lediglich die Einhaltung einer Regel und weisen daher einen geringeren Grad der Eignung auf. Im Anschluss kann das Resultat mit Patterns anderer Bereiche verbunden werden. Da beispielsweise *Cognition patterns* die Wahrnehmungsfähigkeiten des Nutzers beachten, kann durch diese Patterns die Leserichtung des Nutzers festgelegt werden, in folgedessen sich eine weitere Option nicht mehr zur Anreicherung mit Inhalten eignet. Wird festgelegt, dass der Nutzer Präsentationen typischerweise von oben nach unten, von links nach rechts und von nah nach fern analysiert, stellt nur noch eine Zuordnung wie in Abbildung 4.6 eine geeignete Lösung dar.

Die genaue Kenntnis der Bindungen ist erforderlich, da diese nicht immer offensichtlich sein müssen. So ist es möglich, dass die Begleitinformation nicht den Inhalt, sondern das Verständnis der Navigation erleichtern soll, um eine Fehlnutzung zu vermeiden. Dies kann dann von Vorteil sein, wenn Navigationsschritte nicht intuitiv erfassbar sind, mit

diesen unumkehrbare Operationen verbunden sind (Online-Bestellung) oder komplexe Operationen deren Ausführung zeitintensiv ist (Quartalsbericht). In einem solchen Fall passt die Zuordnung am besten zu einer Vorlage, wenn diese der rechten Figur von Abbildung 4.7 entspricht.

Bei Analysen bzw. der Auswertung von Daten kann sich die Art der Navigation bzw. die mögliche Analyseform erst durch den eigentlichen Inhalt ergeben, einschließlich begleitender Informationen. Trifft dies zu, so wird die linksseitig dargestellte Figur der Abbildung 4.7 geeignet. Allgemein besteht somit eine starke Abhängigkeit der Darstellbarkeitseignung von der Semantik der darzustellenden Inhalte.

BEISPIEL VISUELLER ABHÄNGIGKEIT:

Neben den logischen Abhängigkeiten existieren Visuelle, die Gemeinsamkeiten und Unterschiede der darzustellenden Inhalte durch Gestaltungsmittel forcieren, um eine Zusammengehörigkeits- oder Abgrenzungswirkung zu unterstützen. Erneut sollen beispielhaft die bereits verwendeten Inhaltstypen (Navigation, Inhalt, Begleitinformation) innerhalb eines 3-Spaltenraster genutzt werden. Dabei soll die Begleitinformation die Navigation unterstützen, während sich der Inhalt aus der Benutzung der Navigation ergibt. Trotz der schwachen Bindung zwischen der Navigation und der Begleitinformation ist eine Belegung entsprechend Abbildung 4.8 geeignet und widerspricht auch nicht der beschriebenen Bindungspräferenz. Das Beispiel macht jedoch deutlich, dass die Stellung von Begleitinformationen sehr genau zu analysieren ist. In diesem Beispiel ist die Begleitinformation aus Komplexitätsgründen essentiell, um die Navigation sinnvoll nutzen und somit schließlich den gewünschten Inhalt erhalten zu können. Ist hingegen die Begleitinformation nicht erforderlich, um die Navigation zu benutzen, sollte der Bindung zwischen Inhalt und Navigation Vorrang gewährt werden.

Ursprünglich ging das 3-Spaltenraster aus dem 2-Spaltenraster hervor und sollte primär Inhalte mittels Navigation erreichbar machen. Im Unterschied zum 2-Spaltenraster sollte jedoch auch die Präsentation ergänzender, sekundärer Informationen möglich sein. Die Größe des Darstellungsbereichs jedes Inhaltstyps ist eng an seine Bedeutung gebunden. Infolgedessen wird dem Inhalt typischerweise die größte Fläche zur Verfügung gestellt und die Navigation erhält die kleinere Fläche neben dem Inhalt, da sie lediglich Mittel zum Zweck ist, um zum Inhalt zu gelangen. Für die Begleitinformation gilt, dass deren Platzbedarf wahrnehmbar kleiner sein muss als der des Inhaltstyps, den sie begleitet, beispielsweise durch Rückgriff auf das Teilungsverhältnis des *Goldenen Schnittes*. Durch eine Veränderung der Semantik der Inhaltstypen sowie der Anordnung dieser, muss auch die Zuteilung der Flächen angepasst werden. Abbildung 4.8 zeigt zwei mögliche Varianten eines 3-Spaltenraster. Dabei wird deutlich, dass innerhalb des linken Raster, der Begleitinformation deutlich zu viel Platz zur Verfügung steht, um als der Navigation untergeordnet wahrgenommen zu werden. Darüber hinaus existiert keine visuelle Dominanz des Inhalts, da der Navigationsbereich die gleiche Größe besitzt.

Das rechtsseitig dargestellte Raster der Abbildung 4.8 ist eine geeignete Lösung, welche die Semantik der Inhaltstypen berücksichtigt und deren Bedeutung visuell betont. Selbstverständlich wird die optische Dominanz nicht nur durch die Wahl einer geeigneten Größe bestimmt, sondern beispielsweise auch durch die Farbgebung. Strukturelle

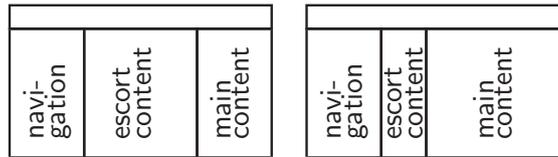


Abbildung 4.8: Visuelle Abhängigkeiten - Grid Größen

Größenunterschiede werden jedoch oft am Stärksten wahrgenommen und sind weitestgehend unabhängig von den darzustellenden Inhalten.

Nicht immer kann auf die Wirkung von Kontrasten bzgl. der Größe zurückgegriffen werden, z.B. wenn gleichwertige Bereiche entstehen sollen, die aber einer Hierarchie unterliegen. Neben einer Unterscheidung durch Gestaltungsmittel besteht die Möglichkeit des Bildens visueller Hierarchien, wie dargestellt in Abbildung 4.9. Diese Hierarchien ermöglichen es, Informationen in einer gewünschten Reihenfolge wahrzunehmen, sogar dann, wenn keine explizite Entwicklungsrichtung oder Abfolge besteht.

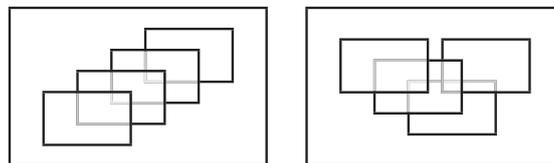


Abbildung 4.9: Visuelle Abhängigkeiten - Grid Stapel

Ferner besteht die Möglichkeit verschiedene Ansätze wie in Abbildung 4.10 zu kombinieren. So lässt sich der obige Ansatz durch die Verwendung verschiedener Größen erweitern, um die Wahrnehmung der Teilflächen als Stapel durch 3-dimensionale Aspekte zu unterstützen. Beide gezeigten Stapel enthalten eine nah-fern und oben-unten Wirkung, die sie klar verständlich machen sollte. Die Größe der Flächen der linksseitigen Lösung nimmt von oben nach unten ab. Dies verstärkt die nah-fern Beziehung und entspricht der Erfahrung und Erwartungshaltung des Nutzers bei 3-dimensionalen Abbildungen. Bei derartigen Verwendung wird die Größe nicht mehr primär mit der Bedeutsamkeit, sondern vor allem mit der Entfernung verbunden. In der rechtsseitig dargestellten Lösung steigt die Größe der Flächen mit zunehmender Entfernung an. Trotz der Kontinuität entsteht hier eine verwirrende Wirkung, da der Anstieg der Erfahrung widerspricht, dass mit der zunehmender Entfernung die Größe abnimmt. Auch wird erwartet, dass weniger Bedeutsames weiter entfernt platziert wird, infolgedessen entweder die Größe oder die Sortierung ungeeignet ist. Mit Ausnahme des Wunsches einer bewussten Provokation und Verwirrung beim Nutzer sollten daher derartige Lösungen vermieden werden.

4.3.4 Progress patterns

Als wichtiger Teil der Anwendungsentwicklung beschreiben *Progress patterns* mögliche Arten der visuellen Entwicklung. Die visuelle Entwicklung entsteht durch die Zeit und den Ort betreffende Änderungen der Darstellung von Objekten. Da das Wahrnehmen

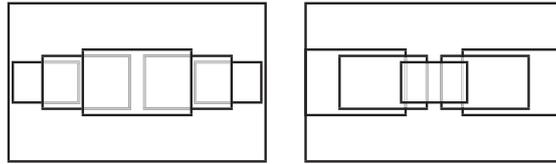


Abbildung 4.10: Visuelle Abhängigkeiten - Grid Größen & Stapel

eines Verlaufs nicht an eine Entwicklungsrichtung gebunden ist, kann sowohl die Zukunft als auch die Historie von Layout-Objekten bedeutsam sein. Auch Kompositionen von Layout-Objekten, deren Elemente in einer definierten Abfolge und an festgelegten Orten präsentiert werden sollen, lassen sich durch Progress patterns beschreiben, erfordern aber unter Umständen die Behandlung von Konflikten.

Die mögliche Illustration eines Verlaufs hängt von den jeweiligen Darstellungszielen und Darstellungsmöglichkeiten ab, nicht jedoch zwingend von der Presentation Dimension. Da sich alle Dimensionen weiterentwickeln können (Abb. 4.11), ist zunächst zu entscheiden, welche Evolution zu visualisieren ist.

| | | | |
|--------------------|------------------|--------------------------|-----------------------|
| Evolving Intention | Evolving Story | Evolving Content | Evolving Presentation |
| | Evolving Context | Evolving Functionalities | |

Abbildung 4.11: Dimensionen der Evolution

Folgend werden exemplarisch ausgewählte Dimensionen betrachtet, deren Evolution visualisiert werden soll.

EVOLUTION DER PRESENTATION DIMENSION

Bei der Suche und Auswahl geeigneter Progress Patterns ist es naheliegend, zunächst Aspekte einer konkreten Darstellung zu berücksichtigen. Ein Arrangement von statischen Objekten innerhalb des Gestaltungsraums erlaubt jedoch nur sehr begrenzt, evolutionäre Änderungen für den Nutzer wahrnehmbar und nachvollziehbar zu machen. Folglich eignet sich diese Herangehensweise für einfache Kompositionen festen Rahmens, während im Allgemeinen eine erweiterte Sicht erforderlich ist, welche überdies Aspekte der Wahrnehmung sowie des Kommunikationskanals berücksichtigt.

Voraussetzung und Grenze des Einsatzes visueller Entwicklungen ist die Möglichkeit der Nutzer, derartige Änderungen wahrnehmen zu können. Dabei ist es nicht erforderlich, Änderungen kontinuierlich und unmittelbar an den dargestellten Objekten wahrzunehmen bzw. verfolgen zu können. Erfahren nur wenige Objekte Änderungen, ist dies zumeist auch dann möglich, wenn die Präsentation des jeweiligen Entwicklungsstands zu diskreten Zeitpunkten größeren Abstands erfolgt (Montageanleitungsprinzip). Mit zunehmender Zahl der Änderungen sinkt die wahrnehmbare visuelle Bindung zwischen zwei Entwicklungsschritten. Existieren zu wenige Gemeinsamkeiten und Orientierungspunkte birgt dies daher die Gefahr des Kohärenzverlusts. Das Maximum möglicher Änderungen lässt sich nicht allgemein fixieren, da dies von der Anzahl der Objekte, deren Anordnung und bisherigen Gemeinsamkeiten, aber auch von den Fähigkeiten des jeweiligen

Nutzers abhängt. Existieren innerhalb einer Darstellung mehrere Entwicklungen, so ist für diese eine örtliche Trennung anzustreben, wenn sie einen ähnlichen oder gleichen Änderungsrhythmus aufweisen jedoch auch wenn sie ähnliche oder aber gegensätzliche Änderungen an gleichen Objekten bewirken. Andernfalls werden die Überschneidungen der Entwicklung zu Fehlinterpretationen führen, infolgedessen die Flexibilität der Darstellung zu deren Vermeidung zu beschränken ist.

In einigen Fällen weisen Änderungen innerhalb der diskreten Entwicklungsschritte nur wenige oder sogar keine Gemeinsamkeiten auf. Dennoch lässt sich eine wahrnehmbare visuelle Entwicklung erreichen. In diesem Fall ist jedoch ein künstlicher Rahmen erforderlich, der die Funktion der fehlenden Bindungen übernimmt und den Fortgang der Entwicklung wahrnehmbar macht. Ein typisches Beispiel für eine solche Entwicklung ist die Abwicklung des Einkaufs in Online-Shops. Um die Ware bestellen zu können, werden den Nutzern mehrere inhaltlich zusammengehörige Formulare dargeboten, deren Elemente aber nur einen sehr geringen visuellen Zusammenhang aufweisen und folglich nicht die Entwicklung wahrnehmbar machen. Um die Sicherheit der Benutzung sicherzustellen und den Verlauf vorhersehbar zu gestalten, wird in der Regel ein Rahmen geschaffen, der die Schritte des Prozesses visualisiert (z.B. benannt oder nummeriert) und zudem die Position des Akteurs innerhalb des Kaufprozesses darstellt.

Trotz der starken Nähe zum Implementationslevel besteht auch für die Präsentationsdimension die Möglichkeit einer konzeptionellen Betrachtung. Die Darstellung einer Entwicklung kann auch auf objektive Kompositionskonzepte zurückgreifen. Durch die Komposition von Objekten, wie bereits durch die Abbildungen 4.9 und 4.10 dargestellt, entsteht nicht allein eine visuelle Hierarchie, sondern zugleich der Eindruck einer Entwicklung, insbesondere bei der Verwendung ähnlicher oder gleicher Objekttypen. Durch die Gleichartigkeit wird intuitiv eine ähnliche Bedeutung zugeordnet, während Überlagerungen mit einem Stapel assoziiert werden. Entsprechend gesammelter Erfahrungen wird intuitiv angenommen, dass neue Elemente oben auf einen Stapel gelegt werden, nicht aber unter diesen oder mittendrin. Durch diese Annahme wird ein Stapel wahrgenommen, der zeitlich sortiert die Verwendungsreihenfolge seiner Elemente abbildet. Ein Progress pattern kann daher für die Präsentation vorgeben, dass Inhalte, die schrittweise präsentiert bzw. durchschritten werden können und vorsortiert sind, entsprechend der linken Figur von Abbildung 4.10 zu präsentieren sind. Oft wird dies bereits implizit umgesetzt, z.B. für Fotoalben. Hierarchien entsprechend der rechten Figur von Abbildung 4.9 erfordern einen kleinen Stapel, der inhaltlich ungeordnet sein kann, aber auch die Flexibilität aufweist, dass der Nutzer die Elemente des Stapels direkt auswählen kann - die klassische Schreibtisch Metapher.

EVOLUTION DER STORY DIMENSION

Eine intuitiv verständliche und gut bekannte Form eines Verlaufs sind Stories [65], die Bestandteil der Story Dimension sind und den Handlungsverlauf im Sinne einer Erzählung spezifizieren. Szenen und Dialogschritte erlauben in diesem Zusammenhang das Spezifizieren von Teilschritten und damit das Reduzieren der Komplexität einer Story durch Aufspalten komplexer Handlungen in Teileinheiten. Aufgrund der Erweiterung des Story-Begriffs in Kapitel 2.1.2 um Handlungsoptionen, können innerhalb einer Story alterna-

tive Handlungspfade beschrieben werden, die es dem Nutzer erlauben, aktiv den Handlungsverlauf einer Story zu beeinflussen. Durch Szenario besteht jedoch weiterhin die Möglichkeit, einen konkreten Weg durch eine Story zu beschreiben. Wegen des schrittweisen Vorgehens ist eine Story insbesondere für Entwicklungen durch Änderungen zwischen Wahrnehmungszeitpunkten prädestiniert. Dennoch schließt die Story Entwicklungen nicht aus, deren Wahrnehmung allein auf örtlichen Änderungen basiert (Abb. 4.9).

Folgend werden die möglichen Formen der Entwicklung innerhalb einer Story näher betrachtet. Hierfür sind die Kompositionstypen der Inhalte zu analysieren, da diese den Verlauf der Handlung und damit die Entwicklung maßgeblich prägen. Die Art des Verlaufs einer Story lässt sich auf dem konzeptionellen Level gut beschreiben. Der so beschriebene Verlauf entspricht typischerweise nicht der späteren Realisierung, definiert aber die nötigen Abhängigkeiten für die Implementationsschicht. Die Story und deren Businessziele können auf diese Weise allgemein gefasst sein und auf dem Implementationslevel eine Reihe von akzeptablen Lösungen ermöglichen ohne diese im Detail festlegen zu müssen.

Die hier unterschiedenen Formen der Entwicklung einer Story sind an eine Klassifikation von Brown [5] angelehnt. Es werden folgende Entwicklungstypen unterschieden:

- | | |
|---------------|-------------|
| - no progress | - network |
| - cell | - ring |
| - cyclic | - evolution |
| - flow | - tree |
| - seed | - wave |

Im Rahmen der Unterscheidung von Entwicklungstypen ist zunächst die elementarste Form zu betrachten. Infolgedessen muss die Klassifikation um den Entwicklungstyp *no progress* (Abb. 4.12a) erweitert werden, um auch Prozesse beschreiben zu können, die bewusst keiner Entwicklung unterliegen sollen. Die Existenz dieses Typs ist nicht hypothetisch, da gelegentlich keine Möglichkeit zur Interaktion als auch kein Bedarf an Änderungen hinsichtlich der Präsentation von Inhalten besteht. Ferner können sehr seltene oder geringfügige Änderungen auftreten, sodass die temporäre oder visuelle Distanz für den Nutzer zu groß ist, um eine Entwicklung wahrnehmen zu können. Auch wenn die Systembenutzungszeit geringer als die Änderungshäufigkeit und die Aufmerksamkeit des Nutzers eingeschränkt ist, kann das Wahrnehmen einer Entwicklung nicht sichergestellt werden und ist partiell gewünscht. Beispielsweise besteht bei elektronischen Werbetafeln bis auf Ausnahmen nicht der Wunsch, einen Verlauf oder einen Zusammenhang zwischen verschiedenen Werbungen wahrnehmen zu können. Vielmehr soll bewusst kein Bezug zwischen den Werbern bestehen, um Interessenkonflikte oder Fehldeutungen durch die potentiellen Kunden zu vermeiden.

Gelegentlich können auch keine Restriktionen bestehen, die den Pfad des Handelns beschränken. Dieser progress type, dargestellt in Abbildung 4.12b, soll mit *network* bezeichnet werden. Aufgrund der fehlenden Schranken des Agierens, weist dieser Entwicklungstyp die höchste Flexibilität auf und gewährleistet sehr kurze Wege zur eigentlichen Information. Durch die Zunahme der Handlungsoptionen, wird jedoch insbesondere bei umfangreichen Anwendungen die Überschaubarkeit durch die Nutzer immer geringer. Infolgedessen sollte dieser Entwicklungstyp lediglich für wenige und allgemeine Teile der Präsentation verwendet werden, um die Zugänglichkeit der Anwendung zu verbessern ohne die Bedienbarkeit zu beeinträchtigen.

Verfügt eine Story über komplexe (Teil-)Szenen, die mittels klar definierter Schnittstellen als separate Einheit zugänglich gemacht werden können, so kann hierfür der *cell* Typ, dargestellt in Abbildung 4.12c, genutzt werden. Durch die Schnittstellen erlaubt dieser Typ neben der Beschreibung komplexer Handlungen die einfache Wiederverwendung und Integration bestehender Handlungskonzepte, da zu diesem Zeitpunkt keine inhaltliche Bindung besteht. Somit lässt sich die Vorgehensweise von Businessprozessen wiederverwenden und auch standardisieren, wie z.B. das Abschließen bzw. Auslösen eines Online-Einkaufs.

Wird eine kreisförmig verbundene Folge festgelegter Schritte spezifiziert, so beschreibt diese den Entwicklungstyp *ring*, dargestellt in Abbildung 4.12d. Trotz der kreisförmigen Bindung gibt dieser Typ keine Entwicklungsrichtung vor, sodass zu jedem Zeitpunkt zwei Handlungsoptionen existieren. Ferner sind iterative Handlungen möglich. Befindet sich oder gelangt ein Akteur in einen echten Ring (ohne Ausweg), entfällt der Storyendpunkt, sodass die Handlung bis zum Abbruch unendlich wird.

Beschreibt ein Ring eine gerichtete, kreisförmig verbundene Folge von Schritten, entfallen Handlungsalternativen, sodass der Nutzer höchstens den Zeitpunkt des Voranschreitens in der Handlung beeinflussen kann, nicht jedoch die Handlungsrichtung. Iterative Handlungen sind in diesem Fall nur noch zyklisch möglich, weshalb dieser Entwicklungstyp als *cyclic*, dargestellt in Abbildung 4.12e, bezeichnet werden soll. Die Unendlichkeit der Handlung bleibt von der zusätzlichen Einschränkung unberührt. Zyklische Handlungen werden in der Regel dann verwendet, wenn Wiederholungen realisiert werden sollen bei denen die Abfolge der Handlungsschritte nicht verändert werden kann oder soll. Dies kann beispielsweise von Vorteil sein, um vorkonfektionierte virtuelle Museumsrundgänge zu spezifizieren [42].

Sind zyklische Handlungen von der Anzahl der Iterationen abhängig und sollen Variationen in Form einer Erweiterung oder Reduktion auftreten, beispielsweise durch Rekursion, so kann der Handlungstyp *evolution*, dargestellt in Abbildung 4.12f, verwendet werden. Beim Feststellen der wiederholten Teilnahme an einem virtuellen Museumsrundgang können durch diesen Entwicklungstyp zusätzliche, weiterführende Informationen stärker in den Fokus des Nutzers gerückt werden. Bei einer solchen Form eines Zyklus ist nicht zwingend durch den Nutzer wahrnehmbar, dass es sich um einen unendlichen Handlungsprozess handelt.

Besteht eine feste Folge von Prozessschritten mit klarem Beginn und Endpunkt, so beschreibt dies den Entwicklungstyp *flow*, dargestellt in Abbildung 4.12g. Existieren

trotz eines festen Verlaufs zu Beginn mehrere Handlungsalternativen, z.B. als Folge einer Individualisierung, so folgt die Handlung dem Entwicklungstyp *seed*, dargestellt in Abbildung 4.12h. Ferner kann dieser Entwicklungstyp benötigt werden, um Kollaborationen zu beschreiben. Eine Verwendung und Integration der Resultate dieses Prozesses ist dadurch möglich, dass das Ergebnis jedes Kollaborationspartners (Endpunkt jeder Prozessfolge) durch einen Handlungstyp wie dem *flow*-Typ verbunden werden. Dabei kann die Reihenfolge und Art der Bindung der Resultate das Gesamtergebnis beeinflussen.

Ist eine Handlung an Dekompositionsentscheidungen des Nutzers gebunden, so wird der Handlungstyp *tree* erforderlich, dargestellt in Abbildung 4.12i. Eine mögliche Anwendung dieses Handlungstyps ist der Bereich des Information Retrieval. Aufgrund der Komplexität und den Anforderungen lexikalischer Dekompositionen, ist der Verwendungsbereich jedoch beschränkt.

Können Duplikate gebildet werden und ist zugleich ein Handlungsverlauf an Dekompositionsentscheidungen gebunden, entsteht ein übergeordneter Entwicklungstyp, der als *wave* Typ bezeichnet werden soll, dargestellt in Abbildung 4.12j.

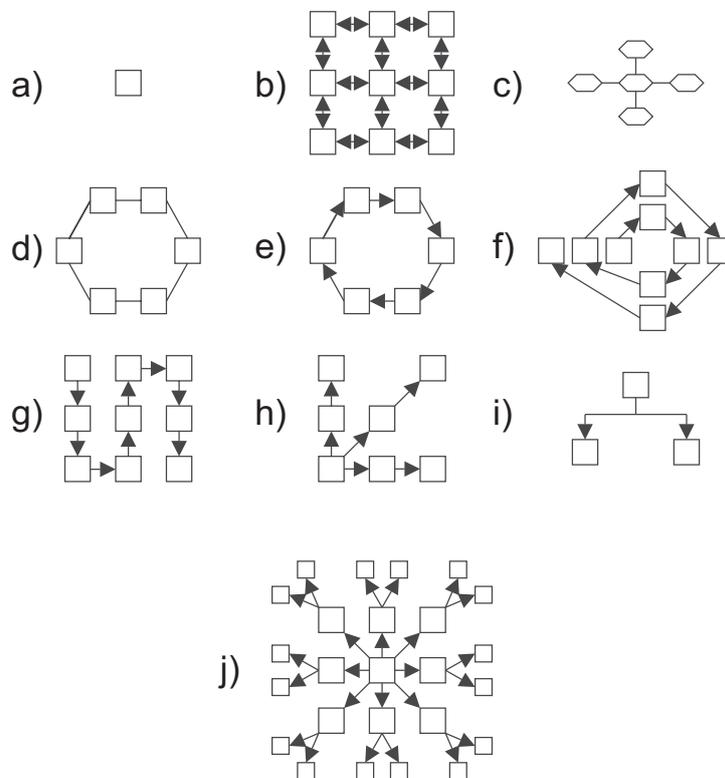


Abbildung 4.12: Typen des Handlungsverlaufs

Abbildung 4.12 unterscheidet mehrere Handlungstypen deren Verwendung von den Zielen der Anwendung und im Detail von der Story abhängt. Wie bereits partiell durch den *wave*-Typ gezeigt, ist die Kombination und Integration von Handlungstypen möglich. Durch die Kombination eines Handlungstypenpaars müssen jedoch nicht zwingend Vorteile entstehen und auch die Integration unterliegt Beschränkungen.

Die Grundlage für die Entwicklung der Handlungstypen sind allgemeine Einflussfaktoren, die in Kombination über die Ziele und Möglichkeiten eines jeden Handlungstyps entscheiden. Die Kombination von Handlungstypen erweitert die Liste der Einflussfaktoren kummulativ und ist die Ursache für den begrenzten Zugewinn bei der Kombination einzelner Handlungstypen. In Tabelle 4.1 wird für jeden der oben beschriebenen Handlungstypen die Liste der Einflussfaktoren aufgeführt. Dies soll zugleich verdeutlichen, dass weder die Liste der Handlungstypen noch die der Einflussfaktoren Anspruch auf Vollständigkeit erhebt und damit je nach Bedarf erweiterbar ist. In Abhängigkeit von der individuellen Sicht auf das System kann sich auch die generelle Klassifikation der Einflussfaktoren unterscheiden.

| | no progress | seed | cyclic | ring | flow | tree | network | cell | wave | evolution |
|-------------|-------------|------|--------|------|------|------|---------|------|------|-----------|
| directed | - | + | + | - | + | + | - | - | + | + |
| decomposed | - | - | - | - | - | + | - | + | + | - |
| iterative | - | - | + | + | - | - | - | - | - | + |
| evolutional | - | + | - | - | + | + | - | - | + | + |
| duplication | - | + | - | - | - | - | - | - | + | - |
| interaction | - | + | + | + | + | + | + | + | + | + |

Tabelle 4.1: Handlungstypen: Einflussfaktoren

In Tabelle 4.1 lassen sich auf einfache Weise Gemeinsamkeiten zwischen den Handlungstypen feststellen. Die genauen Unterschiede und Bindungen sind jedoch nur schwer erkennbar, weshalb diese durch Abbildung 4.13 nochmals gesondert herausgestellt werden sollen. Diese Abbildung stellt dar, aus welchen Handlungstypen andere Handlungstypen durch Verfeinerung der Spezifikation hervorgehen können. Die Verfeinerung wird in diesem Fall durch die Erweiterung um Einflussfaktoren erzielt. Den Ursprung für die Ableitung aller Handlungstypen stellt *no progress* dar, da dieser wegen des fehlenden Entwicklungswunsches durch keine der genannten Einflussfaktoren gekennzeichnet ist. Durch schrittweise Verfeinerung können alle beschriebenen Handlungstypen abgeleitet werden. Es gilt jedoch zu beachten, dass Abbildung 4.13 nicht alle möglichen Transitionen darstellt, sondern nur ausgewählte, um die Ableitbarkeit der beschriebenen Handlungstypen visualisieren zu können. Weitere Transitionen sind möglich, z.B. die grau gefärbte Transition von dem Handlungstyp *seed* hin zum *wave* Typ.

4.3.5 Evolution pattern

Die bisher eingeführten Patterns erlauben es, anhand einer homogenen Suite von Typen der Visualisierung, Inhalte geeignet darzustellen und den Verlauf der Handlung zu visualisieren. Dies wird dem Bedarf kurz- bis mittelfristiger Änderungen gerecht, berücksichtigt aber keine globalen Änderungen. Neben der Visualisierung von Inhalten

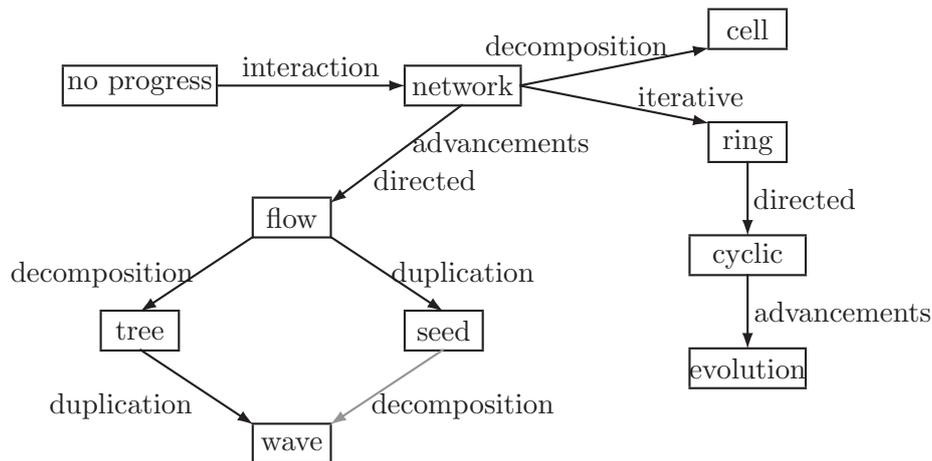


Abbildung 4.13: Handlungstypen: Beziehungen

und Handlungen im Speziellen gilt es auch die Evolution des Layout der Anwendung im Allgemeinen zu betrachten.

Die Entwicklung des Layout von Webanwendungen wird überwiegend als ein abgeschlossener Prozess verstanden. Pflege und Wartung werden daher durch eine teilweise oder vollständige Wiederholung des Entwicklungsprozesses erreicht, entweder bei Bedarf oder einem festen Rhythmus folgend. Ziel der Wiederholung ist es, eine veränderte Wahrnehmung oder neue Wünsche der Nutzer zu berücksichtigen, dem Sinken der Attraktivität des Auftritts entgegenzuwirken oder aber neue Funktionalität sowie Content-Typen zu integrieren.

Da das Layout bislang ausschließlich auf dem niedrigsten Abstraktionsniveau entwickelt wird, dem Implementationslevel, ist der Prozessaufwand einer Entwicklung und Weiterentwicklung stark vom Umfang des Webauftritts abhängig. Templates lösen diese Abhängigkeit zumeist nur unzureichend, sobald eine Vielzahl von Content-Typen existiert und diese flexibel, jedoch in Abhängigkeit von Anderen, nutzbar sein sollen. Daher kann sich der Änderungsaufwand mit steigendem Umfang der Anwendung auch überproportional erhöhen. Dies resultiert zumeist in selteneren Änderungen.

Um auch den Anforderungen umfangreicher Anwendungen besser gerecht zu werden, ist das Layout stärker vom Umfang der Anwendung zu entkoppeln. Möglich ist dies durch die Verlagerung der Spezifikation auf ein höheres Abstraktionsniveau. Bereits auf dem konzeptionellen Level lässt sich die Darstellung von spezifischen Anforderungen der Endgeräte entkoppeln und die Verwendbarkeit von Content-Typen flexibel planen. Auch Farbschemata lassen sich unabhängig entwickeln, indem die Content-Typ-Suite die grundlegenden Anforderungen für geeignete Farbschemata abstrakt definiert, z.B. die nötige Zahl der Farben oder deren zulässige Farbdifferenzen.

Evolution patterns, die die allgemeine Layoutentwicklung betreffen, können mit Hilfe der konzeptionellen Basis die Weiterentwicklung des Layout planen. Hierzu definieren sie Anforderungen und Grenzen von Weiterentwicklungen. Ziel solcher Evolution patterns kann es sein, die mögliche Zahl der Änderungen pro Zeiteinheit zu begrenzen, um die si-

chere Orientierung der Nutzer innerhalb der Anwendung nicht zu gefährden. Übersteigt die Zahl oder der Umfang der Änderungen eine zu definierende Grenze, ist ein Teil der Änderungen zu einem späteren Zeitpunkt anzuwenden. Voraussetzung hierfür ist, dass Änderungsmaßnahmen existierende Abhängigkeiten von anderen Änderungen kennen. Unabhängig von der tatsächlichen Fertigstellung von Änderungsvorhaben, lässt sich auf diese Weise ein permanenter Änderungsprozess der kleinen Schritte nachbilden. Die Schrittgröße sollte sich dabei die Aktivität der Nutzer berücksichtigen, damit für diese keine abrupten Wechsel wahrnehmbar sind und sie sich fortgesetzt in der Anwendung orientieren können.

5 Adaption

Erst mit der Möglichkeit der Adaption von Inhalten entsteht der Bedarf eines content- und benutzungsgesteuerten generischen Layout. Im Bereich der HCI unterscheiden Anke und Sundaram bei der Adaption zwischen *adaptable* und *adaptive* Methoden [3]. Die *adaptable* Methoden erlauben es dem Nutzer, die Inhalte und deren Darstellung selbständig an die eigenen Bedürfnisse anzupassen. Dabei soll sich die Darstellung, die anhand der expliziten Angaben abgeleitet wurde, nicht selbsttätig ändern können. Als *adaptive* bezeichnen die Autoren jene Methoden, welche die Darstellung implizit verändern. Dieses reaktive Verhalten wird unter anderem durch Beobachtung und Berücksichtigung des Nutzerverhaltens möglich.

Auf dem Implementationsniveau erschwert der *adaptable* Ansatz mit einer steigenden Zahl von Nutzern und zunehmender Individualisierung die aktive Weiterentwicklung von Inhalten und deren Präsentation, da weder die gewählten Faktoren noch die Darstellung automatisiert verändert werden dürfen, jedoch der Versuch des Änderns oder Erweiterns von Layout-Komponenten Konflikte mit bestehenden Konfigurationen der Nutzer hervorrufen kann. Im Zuge der Evolution des Systems führt dies schließlich zur Stagnation der Entwicklung der Gestaltung, wenn die unbeschränkte Gültigkeit getätigter Profilanfragen zugesichert werden soll. Andernfalls müsste im Konfliktfall betroffenen Nutzern eine eigenständige Layout-Instanz zugestanden werden, was wegen des damit verbundenen hohen Wartungsaufwands nicht praktikabel ist. Das Verlagern der Spezifikation der Gestaltung auf ein höheres Abstraktionsniveau ermöglicht für den *adaptable* Ansatz das Loslösen von einer konkreten Layout-Instanz und daher einen besseren Umgang mit Erweiterungen. Konflikte lassen sich aber auch dann nicht ausschließen, wenn Layout-Anforderungen der Nutzer weiterhin auf dem Implementationsniveau fixiert werden. Erst wenn auch diese Angaben abstrahiert werden, können die bestehenden Probleme für Erweiterungen als auch Änderungen gelöst werden.

Die als *adaptive* bezeichneten Methoden sind insbesondere für die Entwicklung eines benutzungsgesteuerten Layout hilfreich, da sie es ermöglichen, die Darstellung an die jeweilige Situation anzupassen. Auf dem Implementationsniveau lassen sich die Methoden verwenden, um die Darstellung in ausgewählten und klar definierten Situationen anzupassen. Eine weitergehende Verwendung scheitert aber typischerweise am damit verbundenen hohen Aufwand für deren Erstellung, Nutzung und Pflege. Durch Verlagerung auf ein höheres Abstraktionsniveau können die Methoden verwendet werden, um die Darstellung anhand eines spezifischen Handlungsverlaufs zu entscheiden. Dies begünstigt auch den Umgang mit der Evolution des Gesamtsystems. Soweit die Content-Objekte geeignet erstellt werden, kann auch der erforderliche Umfang der Darstellung nicht nur für eine Gesamthandlung, sondern dynamisch entsprechend des jeweils ermittelten Bedarfs für einen konkreten Inhalt entschieden werden.

5.1 Adaptionfähigkeiten aktueller CMS

Die überwiegende Mehrheit aktueller CMS bietet nur geringfügige Möglichkeiten der Adaption. Selbst bei Vorhandensein machen die Anwendungen nur selten Gebrauch von den gebotenen Möglichkeiten. Hauptursache ist der hohe Pflegeaufwand derzeitiger Lösungen, der durch die Bindung an das Implementationsniveau verursacht wird. Folgend werden mehrere Adaptionstypen unter Berücksichtigung derzeitiger Möglichkeiten betrachtet.

Equipmentadaption:

In begrenztem Umfang unterstützt inzwischen eine Reihe von Entwicklungsumgebungen beim Anpassen der Darstellung an die Fähigkeiten des Equipments. Oft ist dies dennoch mit erheblichem, zusätzlichem Entwicklungsaufwand, Kompromissen hinsichtlich der Flexibilität und Funktionalität sowie Beschränkungen auf Plattformen und Technologien verbunden.

Für viele aktuelle Web-Anwendungen ist die Nutzbarkeit mit unterschiedlichen Gerätekategorien zwingend. Dieser Anforderung versucht man häufig dahingehend Rechnung zu tragen, dass die Gestaltung explizit für die jeweiligen Kategorien entwickelt wird. Da in diesem Fall keine Unterscheidung anhand der Fähigkeiten der Geräte erfolgt, führt dies zur Entwicklung separater Layout-Instanzen mit einhergehender Redundanz oder Beschränkungen auf Gemeinsamkeiten der Gerätekategorien. Darüber hinaus entstehen auch innerhalb einer Kategorie Beschränkungen auf Gemeinsamkeiten, die insbesondere bei hybriden Geräten unbefriedigende Kompromisse bedeuten.

Nutzeradaption:

Die Möglichkeiten der Adaption der Gestaltung an die Bedürfnisse der Nutzer sind bei aktuellen CMS nur rudimentär vorhanden und möglich. Für die Nutzer besteht noch am häufigsten die Möglichkeit, aus vorgegebenen Farbschemata wählen zu dürfen. Eine freie Farbwahl existiert in aller Regel nicht, wenngleich sie grundsätzlich realisierbar ist. Nicht weniger selten können Nutzer aus alternativen Templates wählen. Diese sichern typischerweise gleichwertige Platzverhältnisse zu, um Konflikte bei der Darstellung von Inhalten vermeiden zu können und schränken damit die Möglichkeiten deren Entwicklung ein. Eine echte Adaption und weitergehende Möglichkeiten der Individualisierung scheitern zumeist am damit verbundenen erforderlichen Aufwand einer Realisierung auf dem Implementationsniveau. Eine echte Berücksichtigung der Fähigkeiten und auch der Einschränkungen der Nutzer ist daher derzeit nicht möglich.

Contentadaption:

Die Adaption der Darstellung von Inhalten ist mit derzeitigen CMS fast ausschließlich explizit realisierbar. Zumeist tritt diese Anforderung im Zusammenhang mit der Equipment-Adaption in Erscheinung und sieht sich daher auch mit ähnlichen Problemen konfrontiert. Aufgrund der ebenfalls beschränkten Möglichkeiten, die Bedürfnisse

der Nutzer geeignet aufnehmen zu können, ist die Adaption der Inhalte auch in diesem Fall nur stark eingeschränkt realisierbar.

Da Inhalte oftmals innerhalb einer Baumstruktur zugänglich gemacht werden und nur selten im Rahmen einer Handlung, ist der Gesamtbedarf der Contentadaption nur bedingt erkennbar. Die Spezifikation einer Handlung kann sich derzeit nicht direkt auf die Darstellung von Inhalten auswirken, sondern dient höchstens als unabhängige Orientierung für das spätere Verbinden von Inhalten und das Festlegen erforderlicher Detaillierungsgrade. Gerade die fehlende Unterstützung für Stories erschwert unter anderem die Entwicklung von Lernportalen, die einen großen Bedarf an flexiblen Einstiegen in eine Handlung besitzen und Inhalte auch in Abhängigkeit vom gewählten Zugang darzustellen wünschen.

Benutzungsadaption:

Die Adaption der Darstellung an die Bedürfnisse, die sich während der Benutzung ergeben, ist mit derzeitigen CMS lediglich stark eingeschränkt realisierbar. Aufgrund der Beschränkungen und Hürden der Content- sowie der Nutzeradaption lässt sich die Benutzungsadaption zumeist nur für einfache Entscheidungen realisieren, z.B. ob Inhalte angezeigt werden sollen oder nicht. Außerhalb von Lernportalen wird diese Form der Adaption daher nur selten genutzt.

Der Bereich der Benutzungsadaption umfasst auch Möglichkeiten, die Darstellung der Inhalte, beispielsweise in ihrem Umfang, an den Wissensstand und den gewählten Zugang des Nutzers anpassen zu können. Darüber hinaus lassen sich auch weitere Schritte der Handlung in Abhängigkeit von dem bisher beschrittenen Weg nicht nur ermitteln, sondern entsprechend ihrer Bedeutung angemessen visualisieren. Mit vertretbarem Aufwand und ohne hinreichende Unterstützung von Stories sind diese Formen der Benutzungsadaption aber bislang nicht realisierbar.

5.2 Adaption der Gestaltung

Die Entscheidung der Darstellung von Inhalten wird typischerweise auf dem Implementationsniveau getroffen, z.B. innerhalb der Templatespezifikation einer Web-Anwendung. Die Darstellungsparameter und damit die Aufgaben des Präsentationssystems werden dabei oft anhand von Beispielsituationen bestimmt und wirken nicht selten global. Die Eignung einer Darstellungsform muss jedoch weder allgemeingültig sein, noch lässt sich die Eignung auf diese Weise adäquat prüfen. Ferner steigt immer stärker der Bedarf, Inhalte externer Quellen integrieren und z.B. Mashups bilden zu können [43]. Weder auf die Struktur und Weiterentwicklung solcher Inhalte hat man typischerweise direkten Einfluss, noch ist üblicherweise der Zeitpunkt und Umfang von geplanten Änderungen bekannt. Entsprechend häufig entstehen Situationen, in denen die Darstellung ungeeignet ist und Inhalte unvollständig oder nicht mehr präsentiert werden können, da weder auf abweichende oder neue Darstellungssituationen, noch auf sich ändernden Informationsumfang oder Erweiterungen des Inhaltstyps selbständig reagiert werden kann.

Mit der Trennung der Inhalte vom Layout wurde zwar das Anpassen der Darstellung von Inhalten an veränderte Situationen oder mehrere Umgebungen grundsätzlich möglich, die Trennung erfolgt aber bisher auf dem Implementationsniveau, infolgedessen die Darstellung der Elemente pro Umgebung zu entscheiden ist. Insbesondere wegen des damit verbundenen Wartungsaufwands können meist nur wenige Umgebungen angemessen unterstützt werden.

Die bestehenden Probleme lassen sich vermeiden, wenn die Trennung von Inhalt und Darstellung nicht auf dem Implementationsniveau erfolgt, sondern die Darstellungsanforderungen der Inhalte von den Möglichkeiten der Darstellung getrennt werden. Durch diesen Schritt wird überprüfbar, ob Inhalte mit den verfügbaren Darstellungscontainern dargestellt werden können. Dies umfasst auch Inhalte, für die kein separater Darstellungscontainer existiert, jedoch ein Solcher aus bestehenden Containern und unter Berücksichtigung der Anforderungen gebildet werden kann. Insbesondere den Problemen der Evolution eines Systems und der Integration externer Inhalte lässt sich dadurch begegnen und auch die Konsequenzen geplanter Änderungen werden automatisiert testbar.

Platzbedarf von Inhalten

Soll die Darstellbarkeit von Inhalten hinsichtlich ihres Platzbedarfs geprüft werden, lassen sich *Grids* nutzen, die den Präsentationsbereich in Teilbereiche aufteilen, innerhalb derer die Inhalte darzustellen sind. Eine geeignete Aufteilung des Präsentationsbereichs lässt sich anhand der Inhalte auch vollständig automatisiert finden, wenn die Art und Darstellungsvielfalt von Inhaltsobjekten gering ist und sich zudem klare Regeln für jede Präsentationsituation definieren lassen. Eine geringe Vielfalt ist jedoch nur in sehr wenigen Anwendungsbereichen akzeptabel. Wenn die Gestaltung lediglich eine untergeordnete Rolle einnimmt, weil der Nutzwert infolge einer gewünschten Ordnung der Inhalte maximiert werden soll, lassen sich geeignete Aufteilungen auch automatisch generieren, wie z.B. bei Telefonbüchern oder YellowPages [18]. Für viele Anwendungen stellen aber der in diesem Fall auftretende Mangel an Abwechslungsreichtum und das fehlende Gestaltungspotential eine zu große Einschränkung dar.

Interaktionsbedarf von Inhalten

Neben dem Platzbedarf gehören auch Anforderungen an die Interaktion zum Bereich der Content-Adaption. Die Notwendigkeit ergibt sich oft bereits im Zusammenhang mit einfachen Web-Formularen. Obwohl beispielsweise die Möglichkeiten von Formularelementen mit der Einführung von HTML5 [74] erweitert wurden, sind die Prüfmöglichkeiten weiterhin auf einfache Anforderungen beschränkt. Entsprechend können viele Anforderungen bislang nur durch Kombination mit JavaScript [21] oder durch Zerlegung in weniger komplexe Formulare umgesetzt werden.

Durch die Trennung der Anforderungen von den Möglichkeiten lässt sich auch die Hürde der Spezifikation komplexer, interaktiver Inhalte nehmen, indem diese den Interaktions- und auch Prüfbedarf festlegen, nicht aber die visuelle Umsetzung entscheiden [31]. Die Darstellungscontainer sind jedoch dahingehend zu erweitern, dass sie festlegen, welche Interaktionsanforderungen in welcher Weise umgesetzt werden können und sollen.

Im Ergebnis erlaubt dies nicht nur komplexere Formulare, sondern auch das automatische Wiederverwenden gefundener Lösungen und das Prüfen der Darstellbarkeit. Ferner ist ein einfacher Wechsel der Umsetzung von Interaktionsanforderungen an zentraler Stelle möglich, sobald alternative Darstellungsmöglichkeiten genutzt werden sollen.

Elemente, die innerhalb eines Formulars zu präsentieren sind, können mit Parametern wie den Folgenden angereichert werden:

- Auswahl einer Option - (Selection)
- Abwahl einer Option - (Deselection)
- Vorauswahl einer Option - (Preselection)
- Unveränderliche Option - (Unchangeable option)
- Verpflichtende Auswahl einer Option - (Force selection)
- Verpflichtende Abwahl einer Option - (Force deselection)

Sobald mehrere Elemente verwendet werden, ist auch zu entscheiden, ob diese Gruppen zuzuordnen sind. Im Unterschied zu den bisherigen Möglichkeiten der Darstellung muss die Zugehörigkeit der Elemente im Allgemeinen nicht auf eine einzige Gruppe beschränkt werden. Für jede Gruppe von Elementen sind ebenfalls obige Parameter nutzbar. Ergänzend ist aber zu entscheiden, für wie viele oder welche Elemente der Gruppe die Regeln anzuwenden sind.

Für alle Elemente und Gruppen können komplexe Abhängigkeiten existieren, die es erlauben, den Status von Elementen sowie die Interaktionsmöglichkeiten des Akteurs direkt oder indirekt zu beeinflussen. Um die gewünschten Möglichkeiten der Interaktion und Manipulation von Elementen später entscheiden zu können, sind daher auch deren Abhängigkeiten von anderen Elementen und Gruppen zu berücksichtigen. Umgekehrt kann es auch erforderlich sein, dass Manipulationen an einem Element oder einer Gruppe Änderungen an anderen Elementen oder Gruppen bewirken können. Für Gruppen sind darüber hinaus auch allgemeine Anforderungen realisierbar, um z.B. das Minimum oder Maximum gewählter Elemente festlegen zu können.

Die Anforderungen der Elemente und Gruppen werden schließlich den Möglichkeiten des Darstellungscontainers gegenübergestellt, um zu ermitteln welche Darstellungsvarianten realisierbar sind. Der Blick auf derzeitige Möglichkeiten offenbart, dass bereits für einfache Elemente wie den *Radio*-Buttons keine trivialen Darstellungs-/Interaktionsregeln existieren. Beispielsweise ist in diesem Fall die Abwahl von Elementen nicht explizit möglich, sondern nur durch Auswahl eines anderen Elements der Gruppe. In Kombination mit der Vorauswahl bewirkt dies automatisch einen Entscheidungszwang. Da zudem eine Mehrfachauswahl nicht möglich ist, kann die Vorauswahl kann nur für ein einziges Element erfolgen.

Content-gesteuerte Adaption

Das vorrangige Ziel der Adaption besteht zunächst darin, die grundsätzliche Darstellbarkeit von Inhalten sicherzustellen. Dazu sind eine Reihe von Faktoren zu berücksichtigen, z.B. die Darstellungsfähigkeiten eines Geräts, die Fähigkeiten und Präferenzen des Benutzers, die Interaktionsmöglichkeiten oder die Nutzungssituation innerhalb derer Angaben zu tätigen sind. Dies verhindert Darstellungsfehler, vermeidet aber weder ungeeignete Darstellungen noch ungeeignete Kombinationen von Darstellungselementen.

Sollen auch ästhetische oder spezielle Anforderungen der Anwendung erfüllt werden, sind abhängige Darstellungsanforderungen festzulegen, die für die Darstellung nicht verpflichtend aber von Vorteil sind. Ferner ist festzulegen, welche Anforderungen in welchen Situationen nicht miteinander harmonieren und entsprechend wann der gemeinsame Einsatz zu vermeiden ist. Da sowohl die Attraktivität der Anwendung aufrecht zu erhalten als auch der Verlust der Orientierung zu verhindern ist, gilt es Monotonie in der Darstellung gleichermaßen zu vermeiden, wie das subjektive Wahrnehmen von Unbeständigkeit durch den Akteur. Das Festlegen solcher Regeln kann mit Hilfe der eingeführten Patterns erfolgen.

Durch das Festlegen der weitergehenden Anforderungen lässt sich neben der Darstellbarkeit auch prüfen, ob eine geeignete Darstellung verfügbar ist und erlaubt in der Folge Bewertungen der Darstellungskomponenten und der gesamten Darstellung hinsichtlich ihrer Eignung. Darüber hinaus entsteht das Potential, gesamte Handlungen hinsichtlich harmonisierender Darstellungskomponenten und -regeln zu bewerten.

Benutzungsgesteuerte Adaption

Das Ziel des benutzungsgesteuerten Layout ist es, die Darstellung während der Benutzung dynamisch an die jeweiligen Bedürfnisse des Akteurs und der Handlung anzupassen. Damit dies rechtzeitig möglich ist, muss ein solch reaktives Verfahren Faktoren der Interaktion aber auch des Handlungsverlaufs beobachten zu können.

Die Art der Interaktion mit dem System lässt sich üblicherweise recht gut ermitteln, z.B. durch fehlende Cursor-Bewegungen. Das Berücksichtigen der Art der Interaktion ist für die Darstellung bedeutsam, da die damit verbundene Funktionalität nicht Equipment-übergreifend verfügbar sein muss. Beispielsweise lassen sich Hover-Effekte mit einem Touch-Device höchstens emulieren und sind damit bei der Darstellung zu vermeiden. Touch-Devices erfordern auch größere Interaktionsbereiche wg. der geringeren möglichen Präzision bei der Benutzung.

Neben der Interaktion kann auch die Adaption an die Nutzungsumgebung erfolgen. Beispielsweise kann eine angepasste Darstellung Probleme instabiler Verbindungen reduzieren. In einem ersten Schritt kann auf die Bereitstellung von Begleitinformationen großen Umfangs verzichtet bzw. diese nur noch auf Abruf verfügbar gemacht werden. Darüber hinaus lassen sich umfangreiche Inhaltsobjekte aufteilen, infolgedessen diese innerhalb mehrerer Gestaltungen ausgespielt werden können und damit der Umfang je abgerufener Gestaltung sinkt. Ist nicht nur die Verbindung instabil, sondern auch die Bandbreite gering, kann die Nutzbarkeit zumeist durch vollständigen Verzicht auf optionale Gestaltungselemente oder die Ausgestaltung von Elementen erhalten werden.

Die Navigation eines Akteurs kann in verschiedener Hinsicht Anforderungen an die Gestaltung bewirken. In Verbindung mit einem Touch-Device lässt sich gelegentlich eine kurze Verweildauer auf einer gewählten Webseite gefolgt von einem Rückschritt zur vorhergehenden Seite und dem erneuten Einstieg in eine visuell benachbarte Seite beobachten. Bei gehäuften Auftreten in vergleichbaren Situationen deutet dies typischerweise darauf hin, dass die Darstellung von Navigationselementen zu klein und daher eine treffsichere Navigation schwierig ist.

Die Analyse der Historie der Benutzung kann dabei unterstützen, die Eignung von Handlungen zu prüfen. Hierfür kann der aktuelle Verlauf der Handlung mit dem geplanten Verlauf der Handlung sowie weiteren geeigneten Handlungen verglichen werden. Erforderlichenfalls kann damit auch auf die bessere Eignung alternativer Handlungen hingewiesen werden.

6 Resümee

Anhand der überwältigenden Menge existierender Web-Anwendungen und -portale, lässt sich allgemein beobachten, dass deren Darstellung weiterhin auf dem Implementationslevel entwickelt wird. Die Ursache liegt weniger in fehlenden, technologischen Möglichkeiten der Realisierbarkeit auf höheren Abstraktionsschichten, sondern vielmehr in der Vielzahl verfügbarer Systeme, die den Entwurf auf dem Implementationslevel unterstützen. Diese Systeme erlauben inzwischen eine schnelle Umsetzung und unterstützen in stark begrenztem Umfang auch die Entwicklung für mehrere Plattformen.

Sowohl Adaption als auch Personalisierung sind jedoch auf diesem niedrigen Abstraktionslevel nur mit großem Aufwand oder in kleinem Umfang realisierbar. Das Anpassen der Darstellung an die Bedürfnisse der Nutzer wird daher in aller Regel nicht umgesetzt oder geht nicht über eine minimale Anzahl von Layout- oder Farbvarianten hinaus. Auch die zu erwartende Evolution einer Anwendung kann auf diese Weise nicht frühzeitig berücksichtigt werden.

Ziel der Arbeit war es, mit Hilfe der Screenography das Web-Engineering mit bestehenden HCI-Ansätzen zu verbinden. Unter Berücksichtigung von Erkenntnissen aus dem Bereich der Kunst wurde die Entwicklung der Gestaltung betrachtet und die Erfordernisse und Möglichkeiten diskutiert, um bestehende Beschränkungen aufzulösen, die Flexibilität zu erhalten und auch Adaptivität zuzulassen.

Zusammenfassend lässt sich feststellen, dass die Screenography den Weg weist, um Anwendungen auf einem höheren Abstraktionsniveau entwickeln zu können. Die Entwicklungsdimensionen helfen in diesem Zusammenhang nicht nur bei der Klassifikation der Anwendung sowie dem Auffinden und Festschreiben von Abhängigkeiten, sondern erlauben auch eine Fokussierung der Anwendungsentwicklung, mit dem Ziel, den Entwicklungsaufwand und die damit verbundene Komplexität beeinflussen zu können.

Da der Aufwand einer vollständigen Spezifikation für kleine Projekte typischerweise zu hoch ist, kann dem Aufbau von Pattern-Bibliotheken eine zentrale Bedeutung beigegeben werden. Die Bibliotheken ermöglichen das Wiederverwenden von Anforderungen aber auch das Anpassen an die eigenen Bedürfnisse.

Die Screenography erlaubt das vollständige Loslösen der Entwicklung der Gestaltung von der konkreten Implementierung, indem die Anforderungen der Anwendung und der Inhalte an die Darstellung unabhängig vom Layout festgelegt werden. Demgegenüber legen Layout-Elemente fest, welche Darstellungsmöglichkeiten sie besitzen und machen damit entscheidbar, welche Inhaltstypen präsentierbar sind. Das Layout wird dadurch zu einer Sammlung geeigneter Elemente aus denen die Anwendung wählen darf. Durch das späte Entscheiden der plattform-spezifischen Darstellung wird die Entwicklung als auch die Weiterentwicklung nicht beschränkt und erlaubt es, Plattform- und in Teilen Technologie-unabhängig zu agieren.

Glossar

Akteur

Ein Akteur ist ein aktiv handelnder Nutzer innerhalb des Story-Raums, dessen Handlungsoptionen durch Stories begrenzt werden. Die Handlung eines Akteurs kann durch ein System mittels Entscheidungskriterien und -logiken emuliert werden.

Dramaturgie

Die Dramaturgie hat die Aufgabe ein hinreichend großes Interesse beim Akteur für die Handlung hervorzurufen. Möglich ist dies durch geeignete Wahl eines konkreten Handlungsverlaufs, der auch Unterbrechungen zulassen kann.

Gestaltungsraum

Ein Gestaltungsraum ist ein begrenzter Raum, innerhalb dessen Grenzen sich Gestaltungselemente platzieren und durch Gestaltungsmittel charakterisieren lassen. Eine Anwendung nutzt typischerweise mehrere Gestaltungsräume, deren Erreichbarkeit durch Bindungen zwischen diesen sichergestellt wird. Für einen Gestaltungsraum kann allgemein die verfügbare Funktionalität, mögliche Interaktivität sowie die Anforderungen an Contentobjekte, deren Zusammenwirken und Darstellung definiert werden. Die tatsächliche Visualisierung von Informationen innerhalb eines Gestaltungsraums ist von einer konkreten Handlung abhängig und folgt dem zugehörigen Ausgestaltungs- und Interaktionskonzept.

Grid

Grids stellen ein Ordnungssystem dar, das die Darstellungsfläche in Teilbereiche unterteilt. Traditionell geschieht dies durch Verwendung horizontaler und vertikaler Linien, wodurch eine allgemeine Vorlage mit Halte- und Orientierungspunkten entsteht, an denen sich die späteren Inhalte in einfacher Weise ausrichten lassen.

Raumarrangement

Ein Raumarrangement bezeichnet eine Menge verbundener Gestaltungsräume. Das Raumarrangement ist ein gerichteter Graph, da auch die Bindungen zwischen den Räumen richtungsabhängig sind.

Story (Film)

In diesem Kontext beschreibt die Story eine festgelegte Handlung innerhalb eines Story-Raums. Eine Story greift dabei auf die Datenbasis zu, die ihr durch den Story-Raum zur Verfügung gestellt wird und innerhalb dessen sie definiert wurde. Sie umfasst Szenen, die den Verlauf der Handlung in logische Abschnitte gliedern.

Story (Screen)

Die Story im Kontext von Bildschirmen beschreibt eine spezifizierte Handlung innerhalb eines Story-Raums, ergänzt um die Möglichkeit der Spezifikation alternativer Handlungspfade. Der Verlauf einer Handlung ist durch die Wahl von Handlungsoptionen an die Entscheidungen des jeweiligen Akteurs gebunden.

Story-Raum

Der Story-Raum bildet den Rahmen zur Entwicklung von Stories. Die Stories eines Story-Raums sollen eine gemeinsame Handlungsintention aufweisen, die die allgemeinen Ziele und Aufgaben der zu spezifizierenden Stories beschreibt. Die Stories stellen dem Akteur verschiedene Handlungen und Handlungspfade zur Verfügung, um den unterschiedlichen Bedürfnissen verschiedener Nutzerprofile besser gerecht werden zu können.

Ein Story-Raum bestimmt und beschränkt die Sicht auf die (globale) Datenbasis und damit die Möglichkeiten zur Spezifikation einer Story. Innerhalb einer Anwendung lassen sich mehrere Story-Räume definieren, um disjunkte Teilhandlungen separieren und die lokale Komplexität senken zu können. Dies erleichtert auch die Bindung der Spezifikationsmöglichkeiten an Nutzer oder Gruppenrechte.

Szene

Eine Szene ist eine logische Teileinheit einer Story, die ihrerseits aus weiteren (Teil-)Szenen bestehen kann. Eine Szene beschreibt einen Zeitpunkt oder eine Zeitspanne. Jeder Szene lassen sich Contentobjekte zuordnen, die für die Eingabe, Verarbeitung oder Ausgabe von Inhalten innerhalb der Szene benötigt werden. Contentobjekte einer Elternszene sind auch innerhalb ihrer Kindszenen zugänglich. Jede Szene kann Eingabewerte und -typen definieren, die sie erwartet und welche ihr übermittelt werden können, aber auch festlegen, welche Ausgaben sie produzieren kann und darf.

Szenographie

Die Szenographie hat ihre Wurzeln im Bereich von Theater, Film und Fernsehen und umfasst die kunstvolle Inszenierung von Gestaltungsräumen innerhalb derer konkrete Darstellungen oder Szenen stattfinden. Die tatsächliche Ausgestaltung solcher Räume hängt unter anderem von der Art der Handlung, der beabsichtigten Botschaft sowie der gewünschten Dramaturgie ab..

Index

Akteur, 10

Content, 17

Context, 14

Dramaturgie, 38

Erschließung, 45
 horizontal, 46
 vertikal, 47

Functionality, 17

Gestaltungselement, 39

Gestaltungsmittel, 39

Gestaltungsraum, 40

Grids, 75

Intention, 9

Patterns, 91

Presentation, 18

Raumarrangement, 42, 47

Rundgang
 einfach, 72
 individuell, 73

Screenography, 36

Story, 9

Story-Raum, 10

Suche
 proaktiv, 70
 reaktiv, 71

Szene, 11

Szenographie, 38

Templates, 89

Literaturverzeichnis

- [1] Christopher Alexander. *A Pattern Language: Towns - Buildings - Construction*. Oxford University Press, 1977.
- [2] Margita Altus. *Decision Support for Conceptual Database Design Based on the Evidence Theory: An Intelligent Dialogue Interface for Conceptual Database Design*. Kovac, 2000.
- [3] Jürgen Anke and David Sundaram. Personalization techniques and their application. In Mehdi Khosrow-Pour, editor, *Encyclopedia of E-Commerce, E-Government, and Mobile Commerce*, pages 919–925. Idea Group Reference, 2006.
- [4] D. Benyon, T. Green, and D. Bental. *Conceptual modeling for user interface development*. Springer, London, 1998.
- [5] Laura Brown. *Integration Models: Templates for Business Transformation - The Authoritative Solution*. Sams Publishing, 2000.
- [6] John M. Carroll, editor. *Designing Interaction: Psychology at the Human-Computer Interface*. Cambridge University Press, Cambridge, England, 1991.
- [7] S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai, and M. Matera. *Designing Data-Intensive Web Applications*. Morgan Kaufmann, San Francisco, 2003.
- [8] Noam Chomsky. A review of b. f. skinner’s verbal behavior. *Language*, 35(1):26–58, 1959.
- [9] Noam Chomsky. *Cartesian Linguistics: A Chapter in the History of Rationalist Thought*. Harper & Row, 1966.
- [10] A. Choungourian. Color preferences and cultural variation. In *Perceptual and Motor Skills*, volume 26, pages 1203 – 1206. Southern Universities Press, 1968.
- [11] J. Conallen. *Building Web Applications with UML*. Addison-Wesley, Boston, 2003.
- [12] G. Scott Danford, Beth Tauke, State University of New York at Buffalo. Center for Inclusive Design & Environmental Access, New York (N.Y.). Mayor’s Office for People with Disabilities, New York (N.Y.). Dept. of Design, Construction, American Institute of Architects. New York Chapter, and New York (N.Y.). Office of the Mayor. *Universal Design New York*. City of New York Office of the Mayor, New York, 2001.

- [13] O. De Troyer and C. Leune. WSDM: A user-centered design method for web sites. In *Computer Networks and ISDN Systems – Proceedings of the 7th International WWW Conference*, pages 85–94. Elsevier, 1998.
- [14] Manfred Fuhrmann, editor. *Aristoteles: Poetik, c. 335 BC*. Reclam, 1994.
- [15] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
- [16] J. W. Goethe. *Farbenlehre*. Cotta, Stuttgart, 1810.
- [17] David S. Gorfein and William G. Stone. Encoding interval in short-term memory. *Journal of Verbal Learning and Verbal Behavior*, 6:520 – 522, 1967.
- [18] Winfried H. Graf. Constraint-based graphical layout of multimodal presentations. In *Readings in intelligent user interfaces*, pages 263–285. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
- [19] Sung H. Han and Sang W. Hong. A systematic approach for coupling user satisfaction with product design. *Ergonomics*, 46(13-14):1441–1461, 2003.
- [20] G.-J. Houben, P. Barna, F. Frasinca, and R. Vdovjak. HERA: Development of semantic web information systems. In *Third International Conference on Web Engineering – ICWE 2003*, volume 2722 of *LNCS*, pages 529–538. Springer-Verlag, 2003.
- [21] Ecma International. Standard ecma-262 - ecmascript 2016 language specification. <http://www.ecma-international.org/publications/standards/Ecma-262.htm>, 2016.
- [22] J. Itten. *Kunst der Farbe*. O. Maier, 1961.
- [23] S. Johnson. *Interface culture*. Harper, San Francisco, 1997.
- [24] Gaetano Kanizsa. Margini quasi-percettivi in campi con stimolazione omogenea. *Rivista di Psicologia*, 49(1):7–30, 1955.
- [25] Boris Herbert Kleint. *Bildlehre*. Schwabe & Co., Basel, 1929.
- [26] Kurt Koffka. *Die Grundlagen der psychischen Entwicklung*. A.W. Zickfeldt, Osterwieck / Harz, 1921.
- [27] Wolfgang Köhler. *Gestalt Psychology*. H. Liveright, New York, 1929.
- [28] John Krauskopf. Effect of retinal image stabilization on the appearance of heterochromatic targets. *J. Opt. Soc. Am.*, 53(6):741–741, Jun 1963.

- [29] D. Lowe, B. Henderson-Sellers, and A. Gu. Web extensions to UML: Using the MVC triad. In Stefano Spaccapietra, Salvatore T. March, and Yahiko Kambayashi, editors, *Conceptual Modeling – ER 2002*, volume 2503 of *LNCS*, pages 105–119. Springer-Verlag, 2002.
- [30] Kris Luyten, Karin Coninx, Marc Abrams, and Harmonia Inc. Integrating uiml, task and dialogs with layout patterns for multi-device user interface design. In *The 11th International Conference on Human-Computer Interaction, Las Vegas*, pages 22–27, 2005.
- [31] Hui Ma, René Noack, Faizal Riaz-ud-Din, Klaus-Dieter Schewe, and Bernhard Thalheim. Capturing forms in web information systems. In *4th International Conference on Innovations in Information Technology (IIT)*, pages 198–202, Nov 2007.
- [32] George Armitage Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, 1956.
- [33] T. Moritz. Skriptum - Grundlagen der Gestaltung visueller Medien. Media Design course, Computer Science Institute, Cottbus University of Technology, 1997.
- [34] T. Moritz. *Visuelle Gestaltungsraster interaktiver Informationssysteme als integrativer Bestandteil des immersiven Bildraumes*. PhD thesis, HFF Berlin-Babelsberg, 2007.
- [35] T. Moritz, R. Noack, K.-D. Schewe, and B. Thalheim. Principles of Screenography. CAiSE Forum, 2007. 4 pages.
- [36] T. Moritz, K.-D. Schewe, and B. Thalheim. Strategic modelling of web information systems. *International Journal on Web Information Systems*, 1(4):77–94, 2005.
- [37] Thomas Moritz, René Noack, Klaus-Dieter Schewe, and Bernhard Thalheim. Intention-driven screenography. In Heinrich C. Mayr and Dimitris Karagiannis, editors, *ISTA*, volume 107 of *LNI*, pages 128–139. GI, 2007.
- [38] Josef Müller-Brockmann. *Gestaltungsprobleme des Grafikers*. Niggli, Sulgen/Zurich, 1961.
- [39] Josef Müller-Brockmann. *Grid systems in graphic design*. Niggli, Sulgen/Zurich, 1996.
- [40] Ulric Neisser. *Cognitive Psychology*. Prentice Hall, 1967.
- [41] René Noack. Generische Gestaltung von Web-Oberflächen. Master’s thesis, Brandenburgische Technische Universität Cottbus, Institut für Informatik, 2005.
- [42] René Noack. Patterns for museums. In Andreas Bienert, Gerd Stanke, James Hemsley, and Vito Cappellini, editors, *Electronic Media and Visual Arts - EVA 2008*. Gesellschaft zur Förderung angewandter Informatik e.V., 2008.

- [43] René Noack. Rules and patterns for website orchestration. In Jianhua Yang, Athula Ginige, Heinrich C. Mayr, and Ralf-Detlef Kutsche, editors, *UNISCON*, volume 20 of *Lecture Notes in Business Information Processing*, pages 268–279. Springer, 2009.
- [44] René Noack and Bernhard Thalheim. Patterns for screenography. In Roland Kaschek, Christian Kop, Claudia Steinberger, and Günther Fliedl, editors, *UNISCON*, volume 5 of *Lecture Notes in Business Information Processing*, pages 484–495. Springer, 2008.
- [45] D. Norman. *The design of everyday things*. Basic Books, New York, NY, 1988.
- [46] Theresa A. O’Connell and Elizabeth D. Murphy. The usability engineering behind user-centered processes for web site development lifecycles. In P. Zaphiris, editor, *Human Computer Interaction Research in Web Design and Evaluation*, pages 1–21. Information Science Publishing, 2007.
- [47] Fabio Paterno. *Model-Based Design and Evaluation of Interactive Applications (Applied Computing)*. Springer, 1999.
- [48] Simon Polovina and Will Pearson. Communication + dynamic interface = better user experience. In Claude Ghaoui, editor, *Encyclopedia of Human Computer Interaction*, pages 85–91. Idea Group Reference, 2006.
- [49] Ravi Sandhu, David F. Ferraiolo, and D. Richard Kuhn. Role-based access controls. American National Standard for Information Technology - ANSI INCITS 359-2004, 2004.
- [50] Joseph M. Scandura and William G. Roughead Jr. Conceptual organizers in short-term memory. *Journal of Verbal Learning and Verbal Behavior*, 6(4):679 – 682, 1967.
- [51] K.-D. Schewe and B. Thalheim. The co-design approach to web information systems development. *International Journal on Web Information Systems*, 1(1):5–14, 2005.
- [52] K.-D. Schewe and B. Thalheim. Conceptual modelling of web information systems. *Data and Knowledge Engineering*, 54(2):147–188, 2005.
- [53] K.-D. Schewe and B. Thalheim. Usage-based storyboarding for web information systems. Technical Report 0613, Christian Albrechts University Kiel, Germany, 2006.
- [54] K.-D. Schewe and B. Thalheim. User models: A contribution to pragmatics of web information systems design. In K. Aberer, Z. Peng, and E. Rundensteiner, editors, *Web Information Systems – Proceedings WISE 2006*, volume 4255 of *LNCS*, pages 512–523. Springer-Verlag, 2006.
- [55] Klaus-Dieter Schewe and Bernhard Thalheim. Conceptual modelling of web information systems. *Data & Knowledge Engineering*, 54(2):147–188, 2005.

- [56] Klaus-Dieter Schewe and Bernhard Thalheim. Life cases: A kernel element for web information systems engineering. In Joaquim Filipe and José A. Moinhos Cordeiro, editors, *WEBIST (Selected Papers)*, volume 8 of *Lecture Notes in Business Information Processing*, pages 139–156. Springer, 2007.
- [57] Klaus-Dieter Schewe and Bernhard Thalheim. Pragmatics of storyboarding for web information systems: usage analysis. *IJWGS*, 3(2):128–169, 2007.
- [58] Friedrich Schumann. Beiträge zur Analyse der Gesichtswahrnehmungen. Erste Abhandlung: Einige Beobachtungen über die Zusammenfassung von Gesichtseindrücken zu Einheiten. *Zeitschrift für Psychologie und Physiologie der Sinnesorgane*, 23:1–23, 1900.
- [59] D. Schwabe and G. Rossi. An object oriented approach to web-based application design. *TAPOS*, 4(4):207–225, 1998.
- [60] B. Shneiderman and C. Plaisant. *Designing the User Interface: Strategies for effective human-computer interaction*. Addison-Wesley, x, 2004.
- [61] David Skopec. *Digital Layout for the Internet and other Media*. Ava Publishing SA, 2003.
- [62] Kenia Sousa, Albert Schilling, and Elizabeth Furtado. Integrating usability, semiotic, and software engineering. In Aristides Dasso and Ana Funes, editors, *Verification, Validation and Testing in Software Engineering*, pages 47–70. Idea Group Publishing, 2007.
- [63] Lothar Spillmann. Gehirn und Gestalt. *Kognitionswissenschaft*, 9(3):122–143, 2001.
- [64] Srinath Srinivasa. *A calculus of fixpoints for characterizing interactive behavior of information systems*. PhD thesis, Brandenburg University of Technology Cottbus, Faculty of Mathematics, Natural Sciences and Computer Science, 2001.
- [65] B. Thalheim. Co-design of structuring, functionality, distribution, and interactivity of large information systems. Technical Report 0315, Cottbus University of Technology, Computer Science Institute, 2003.
- [66] B. Thalheim. Model suites. In *2nd International Workshop on Knowledge Cluster Systems*, pages 20–40. IOS Press, 2008.
- [67] Bernhard Thalheim. *Entity-Relationship Modeling - Foundations of Database Technology*. Springer, 2000.
- [68] Bernhard Thalheim and Antje Düsterhöft. Sitelang: Conceptual modeling of internet sites. In Hideko S. Kunii, Sushil Jajodia, and Arne Sølvberg, editors, *ER*, volume 2224 of *Lecture Notes in Computer Science*, pages 179–192. Springer, 2001.

- [69] Jan Tschichold. *Die neue Typografie*. Verlag des Bildungsverbandes der Deutschen Buchdrucker, 1928.
- [70] TYPO3-Association. Typo3-The Enterprise Open Source CMS. <https://typo3.org/>, 2016.
- [71] E. Vale. *The technique of screen and television writing*. Simon and Schuster, New York, 1982.
- [72] D. K. van Duyne, J. A. Landay, and J. I. Hong. *The Design of Sites: patterns, principles and processes for crafting a customer-centered Web experience*. Addison Wesley, 2002.
- [73] Martijn van Welie and Gerrit C. van der Veer. Pattern languages in interaction design. In Matthias Rauterberg, Marino Menozzi, and Janet Wesson, editors, *INTERACT*. IOS Press, 2003.
- [74] W3C. Html5. <https://www.w3.org/TR/html5/>, 2014.
- [75] Qing Wang and René Noack. Intelligent author identification. In Juan Trujillo, Gillian Dobbie, Hannu Kangassalo, Sven Hartmann, Markus Kirchberg, Matti Rossi, Iris Reinhartz-Berger, Esteban Zimányi, and Flavius Frasinca, editors, *ER Workshops*, volume 6413 of *Lecture Notes in Computer Science*, pages 96–106. Springer, 2010.
- [76] Max Wertheimer. Experimentelle Studien über das Sehen von Bewegung. *Zeitschrift für Psychologie*, 61:161–265, 1912.
- [77] Max Wertheimer. Untersuchungen zur Lehre von der Gestalt. II. *Psychologische Forschung*, 4(1):301–350, 1923.

