

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. Ein 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.

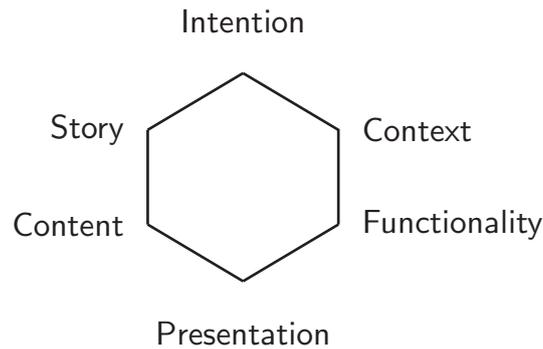


Abbildung 2.1: 6 Dimensionen der WIS Entwicklung

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