

# Proposal

zur Diplomarbeit

**„DiaGen:**

Ein Framework zur  
automatischen Dialogmaskengenerierung  
für geräteunabhängige Web-Anwendungen“

von Matthias Lehmann

vergeben vom Lehrstuhl für Angewandte Telematik / e-Business  
der Universität Leipzig

Betreuer: Dipl.-Inform. Matthias Book  
erster Gutachter: Prof. Dr. Volker Gruhn

14.Mai 2004

## Inhaltsverzeichnis

Einleitung.....	3
Vorhandene Arbeiten.....	4
Device Independence.....	4
Abstrakte Dialogbeschreibung.....	5
Dialoggenerierung.....	6
Dialogsteuerung in webbasierten Anwendungen.....	6
Aufgabenstellung.....	6
Lösungsansatz.....	7
Abstrakte Dialogbeschreibungssprache.....	8
Gesamtarchitektur.....	8
Ermittlung der Geräteigenschaften.....	8
Rendern der Dialogmasken.....	9
Werkzeuge.....	9
Geplantes Vorgehen.....	9
Stand der Arbeit.....	11
Literaturliste.....	12

## Einleitung

In den letzten Jahren und Jahrzehnten haben besonders zwei technologische Entwicklungen unser alltägliches Leben revolutioniert – der Siegeszug der Personal Computer und der des Internet. Mittlerweile gehört nicht nur ein Personal Computer zur Standardausrüstung eines jeden Haushaltes, als Laptop, Palmtop, Personal Digital Assistant (PDA), Smartphone oder sonstigem Embedded Device sind Computer zu unserem ständigen Begleiter und mobilem Werkzeug für alle möglichen alltäglichen Aufgaben und Beschäftigungen geworden.

In ähnlicher Weise hat das Internet einen allgegenwärtigen Platz und eine unverzichtbare Funktion in unserem wirtschaftlichen, gesellschaftlichen und persönlichen Leben eingenommen und stellt dabei immer anspruchsvollere Dienste zur Verfügung. Dienten die Mehrzahl der Angebote im Netz zunächst lediglich der Information und bestanden daher aus statischen Seiten, so bieten immer mehr Angebote die Möglichkeit der Interaktion, wickeln Transaktionen ab und stellen Anwendungen zur Verfügung. Durch die weite Verbreitung, Zuverlässigkeit und Bekanntheit der Internettechnologien, kommen diese auch als vorrangige Technologie für Intra- und Extranets zum Einsatz und lösen dabei vielfach alte Technologien ab. So werden beispielsweise viele alte Terminalanwendungen zu hypertext-basierten Webanwendungen umgewandelt.

Durch das Zusammentreffen dieser beiden Entwicklungen tritt die Situation auf, dass Angebote im Internet von einer Vielzahl unterschiedlicher Endgeräte aufgerufen werden. Diese Endgeräte unterscheiden sich in einer großen Zahl von Eigenschaften voneinander – sie besitzen verschiedene Displaygrößen, können mit verschiedenen Datentypen und Auszeichnungssprachen umgehen, weisen verschiedene Ein- und Ausgabemöglichkeiten auf usw.

Es gilt nun, auf diese Situation geeignet zu reagieren, damit die Angebote weiterhin von allen Endgeräten aus sinnvoll nutzbar sind, nicht eine Zersplitterung des Internets in verschiedene gerätspezifische Segmente stattfindet und die Erstellung und Wartung solcher Angebote trotzdem praktikabel ist.

Dies stellt im Falle von Webanwendungen eine besondere Herausforderung dar, da die Funktionalität und Verarbeitung der Anwendung abhängig von den unterschiedlichen Eigenschaften der Endgeräte unterschiedlich ausfallen kann.

Diese Arbeit will für einen begrenzten Teil des dargestellten Problems eine Lösung finden, und zwar möchte sie die Entwicklung strikt dialogbasierter Webanwendungen – wie etwa ehemaliger Terminal-Masken-Anwendungen – ermöglichen, so dass diese gerätunabhängig und damit speziell auch mit mobilen Endgeräten nutzbar sind, indem die Dialogmasken der Anwendung gerätspezifisch aus einer abstrakten Dialogbeschreibung generiert werden.

## **Vorhandene Arbeiten**

### **Device Independence**

Der Problemstellung der Geräteunabhängigkeit (Device Independence) wird zur Zeit einige Beachtung geschenkt. So hat das W3C eine „Device Independence Working Group“ eingerichtet, welche sich damit befaßt, Methoden zu entwickeln, welche einerseits die Eigenschaften des aufrufenden Endgerätes zur Verarbeitung zugänglich machen und andererseits Autoren bei der Erstellung von gerätunabhängigen Seiten unterstützen ([W3C-DI], Abschnitt „What is Device Independence“). Aus deren Arbeit ist bisher vor allem die Definition der Composite Capability/Preference Profiles ([CC/PP]) hervorgegangen, welche eine Beschreibung der Geräteigenschaften und Nutzerpräferenzen ermöglicht. Diese werden vom Endgerät an den Server bereitgestellt, so dass dieser die Informationen in die Verarbeitung und Angebotsadaption einbeziehen kann.

Parallel dazu hat die Wireless Application Group die User-Agent-Profiles ([UAPProf]) entwickelt, mit welchen WAP-Geräte ihr Profil, also eine formale Beschreibung ihrer Eigenschaften, bekanntgeben. Beide Spezifikationen sind so gestaltet, dass sie sich gegenseitig ergänzen.

Alternativ dazu besteht das WURFL-Projekt ([WURFL]), welches für die gleiche Problemstellung eine andere, pragmatischere, Lösung anbietet, indem die Eigenschaften vieler Endgeräte in einer XML-Datei auf dem Server gespeichert sind und über Bibliotheksaufrufe abgefragt werden können.

Ausserdem wurde im Rahmen der Entwicklung eines „Content Negotiation and Adaption Framework“ (siehe [Lemlouma01]) von Tayeb Lemlouma die Entwicklung von Universal Profiling Schemata ([UPS]) begonnen, welche ebenfalls dazu dienen, relevante Informationen bezüglich der Geräteigenschaften und Dienstleistung bereitzustellen.

In den HP Laboratories Bristol werden – mit besonderem Blick auf CC/PP und UAProf - praktische Lösungen zur Umsetzung von Device Independence untersucht und entwickelt ([Butler01a], [Butler01b], [Butler01c], [Butler02a]), und ein „Device Independent Web Application Framework“ konzipiert ([Giannetti02]). Dieses Framework will Inhalte mittels „Single Authoring“ für alle Geräteklassen nutzbar erstellen, dabei liegt der Schwerpunkt zunächst hauptsächlich auf dem gerätunabhängigem Zugriff auf Informationen. Zur Zeit der Veröffentlichung des genannten White Papers waren bereits erste Konzeptideen zur Integration von Formularen – und damit der Ermöglichung von gerätunabhängiger Interaktion und Applikationen – vorhanden, diese waren jedoch noch nicht konkret ausgearbeitet.

Viele andere Projekte und Gruppen befassen sich direkt oder indirekt mit dem Thema der Gerätunabhängigkeit, die meisten davon haben jedoch hauptsächlich den geräteunabhängigen Informationszugriff im Blick – Dokumente werden mittels besonderer Erstellungsmethoden so erstellt oder mittels späterer Verarbeitung, bspw. durch Proxy-Server, so adaptiert, dass sie von den verschiedenen Endgeräten aus verarbeitet werden können – gerätunabhängige Interaktion über Formulare bleibt dabei meist ausgeklammert.

### **Abstrakte Dialogbeschreibung**

Auch zum Thema der abstrakte Beschreibung von Dialogen und Benutzerschnittstellen gibt es einige Arbeiten. Vom XIML-Forum wird XIML – eine „eXtensible Interface Markup Language“ entwickelt (siehe [Puerta01]), dabei handelt es sich um „a universal specification for interaction data and knowledge related to a user interface“ ([Ximl.Org], „About XIML“), ebenso von der OASIS-Open [UIML] – die „User Interface Markup Language“, welche „a canonical representation of any user interface (UI) suitable for multi-platform, multi-lingual, and multi-modal UI“ ([OASIS03], Seite 9) ermöglichen will.

Von besonderem Interesse ist die [XForms] Spezifikation vom W3C, welche im Oktober 2003 den Status einer W3C Recommendation erreicht hat. Damit sollen Web-Formulare mit einer Vielzahl von Plattformen und Endgeräten nutzbar werden.

## **Dialoggenerierung**

Im Rahmen der Problemstellung des „ubiquitous computing“ hat sich Krzysztof Gajos in [Gajos04] mit der automatischen Generierung von Nutzerschnittstellen für unterschiedliche Endgeräte beschäftigt. Er betrachtet die Dialoggenerierung darin als Optimierungsproblem und stellt einen effektiven Algorithmus zu dessen Bewältigung vor.

## **Dialogsteuerung in webbasierten Anwendungen**

Am Lehrstuhl für Angewandte Telematik / e-Business der Universität Leipzig wird von Matthias Book das „Dialog Control Framework“ entwickelt (siehe [Book/a]). Dieses soll für Webanwendungen ein gleiches Maß von Benutzbarkeit ermöglichen, wie das der Nutzer von fensterbasierten Benutzersystemen gewöhnt ist, indem es Dialoglogik für komplexe und verschachtelte Dialogstrukturen bietet. Zur Spezifikation solcher Dialogstrukturen wird dort außerdem die XML-basierte „Dialog Flow Notation“ (siehe [Book/b]) konzipiert.

## **Aufgabenstellung**

In dieser Arbeit soll ein kleines Framework – DiaGen – entwickelt und prototypisch implementiert werden, welches die gerätunabhängige Nutzung von strikt dialogbasierten Webanwendungen ermöglichen soll.

Im allgemeinen bestehen Webanwendungen meist aus einem Mix von Inhalten, Multi-Media-Objekten und eingebetteten Formularen. Mit der Bezeichnung „strikt dialogbasierte Webanwendungen“ soll dies dahingehend eingegrenzt werden, dass eine solche Webanwendung ausschließlich aus miteinander verknüpften Dialogmasken besteht. Das bedeutet, dass es sich bei DiaGen nicht um eine allgemeine Lösung für das Problem der Gerätunabhängigkeit handelt, sondern dass hiermit lediglich eine Lösung für die besondere Situation der gerätunabhängigen Nutzung solcher strikt dialogbasierten Webanwendungen gefunden werden soll.

Die Gerätunabhängigkeit wird insofern eingegrenzt, als in dieser Arbeit lediglich visuelle hypertext-basierte Darstellungen der Dialoge betrachtet werden, wie sie auf derzeitigen mobilen Endgeräten üblich sind, andere Darstellungsarten – etwa

akkustisch mittels Sprachsynthese- oder haptisch durch Braille-Geräte – bleiben unberücksichtigt.

Weiterhin handelt es sich bei DiaGen auch nicht um ein allgemeines Dialogfluss-Steuerungs-Framework – für derartige Anforderungen sei auf das „Dialog Control Framework“ (DCF) verwiesen. DiaGen soll jedoch in geeigneter Weise mit dem DCF interagieren, indem es einerseits dessen Funktionalität nutzt und sich andererseits gut modular in das DCF einbinden lässt.

Zur Konzeption von DiaGen sollen bestehende Standards und Entwicklungen (wie etwa CC/PP, UAProf, XForms, XHTML-MP etc.) berücksichtigt und soweit sinnvoll eingesetzt werden.

Bei DiaGen geht es vor allem darum, die Dialogmasken entsprechend den Charakteristiken des aufrufenden Gerätes aus einer abstrakten Dialogbeschreibung zu generieren, es sollen jedoch auch die dabei auftretenden Randprobleme - wie die Dialogfluss-Steuerung innerhalb der Dialogsequenzen und die Validierung der Eingaben mit entsprechender Fehlerbehandlung - berücksichtigt werden.

Da die Implementierung reinen prototypischen Charakter hat, werden keine industriellen Anforderungen hinsichtlich Performance und Robustheit gestellt.

## **Lösungsansatz**

Die Grundidee bei DiaGen besteht darin, die Dialogmasken für die unterschiedlichen Endgeräte dynamisch aus einer abstrakten Dialogbeschreibung zu generieren.

Beim Aufruf eines Dialogs durch ein Endgerät, müssen zunächst dessen Eigenschaften – vor allem seine Displaygröße – ermittelt werden. Mit diesen Informationen wird die zum angeforderten Dialog gehörende abstrakte Dialogbeschreibung in eine Sequenz von Dialogmasken geteilt, welche der Displaygröße des aufrufenden Endgerätes entsprechen. Diese Dialogmasken müssen nun entsprechend der vom aufrufenden Endgerät verwendeten Markupsprache und ggf. unter Berücksichtigung weiterer Eigenschaften

gerendert werden (wobei das Teilen in Dialogmasken und deren Rendern praktisch wahrscheinlich nicht in getrennten Arbeitsschritten durchgeführt werden kann, da zwischen beiden viele Abhängigkeiten bestehen). Die Dialogmasken werden nacheinander an das Endgerät gesendet, wobei dort vom Nutzer Eingaben getätigt werden. Diese Eingaben sollten vom DiaGen-Server validiert werden. Wurden unzulässige Werte eingegeben, muß die betreffende Dialogmaske erneut an das Endgerät gesendet werden.

Anhand dieser kurzen Ablaufbeschreibung werden folgende Teilprobleme deutlich:

- eine abstrakte Dialogbeschreibungssprache muss ausgewählt und problemspezifisch erweitert werden
- eine Gesamtarchitektur für den DiaGen-Server muss konzipiert werden
- es müssen Mechanismen und Algorithmen entwickelt werden, welche
  - die Eigenschaften des Endgerätes ermitteln und für die weitere Verarbeitung bereitstellen
  - die abstrakte Dialogbeschreibung entsprechend den Eigenschaften des Endgerätes in eine Sequenz von Dialogmasken zerlegen und diese rendern
  - den Fluß zwischen den Dialogmasken steuern
  - Eingaben validieren und Dialogmasken zur Fehlerbehandlung erstellen

### **Abstrakte Dialogbeschreibungssprache**

Die abstrakte Dialogbeschreibungssprache für DiaGen wird wahrscheinlich auf [XForms] basieren. Als W3C Recommendation ist XForms bekannt und bereits vielfach eingesetzt. Allerdings sollen auch andere Beschreibungssprachen wie [UIML] und XIML (siehe [Puerta01]) auf ihre Eignung untersucht werden. Sollten sie für diesen Zweck besser geeignet sein, ist einer dieser Sprachen natürlich der Vorzug zu geben. In jedem Fall wird die zugrundegelegte Beschreibungssprache jedoch problemspezifisch erweitert werden müssen; da es sich jeweils um XML-basierte Sprachen handelt, ist das über XML-Namensräume sauber möglich.

### **Gesamtarchitektur**

Die Gesamtarchitektur soll so gestaltet werden, dass sie eine gute Integration in das DCF (siehe [Book/a]) bietet und dessen Funktionalität soweit wie möglich nutzt.

## **Ermittlung der Geräteigenschaften**

Für die Ermittlung und Bereitstellung der Eigenschaften des aufrufenden Gerätes ist geplant, auf die aktuellen Standards [CC/PP] und [UAProf] zurückzugreifen. Diese Repräsentieren die Arbeitsergebnisse anerkannter und kompetenter Gremien (des W3C und der Wireless Application Group), von ihnen ist in Zukunft eine zunehmende Unterstützung durch Technik und Wirtschaft zu erwarten, außerdem existiert ein Softwarepaket „DELI“ (siehe [Butler01a]), welches auf CC/PP und UAProf basiert und somit einen Großteil der benötigten Funktionalität bereits zur Verfügung stellt. Eventuell sind auch alternative Lösungen mittels [WURFL] oder [UPS] in Erwägung zu ziehen, falls sich dieser Lösungsansatz als unpraktikabel erweisen sollte.

## **Rendern der Dialogmasken**

Die hauptsächliche Herausforderung beim Rendern der Dialogmasken besteht darin, aus der Dialogbeschreibung derart eine Sequenz von Dialogmasken zu erstellen und diese Dialogmasken zu layouten, dass diese möglichst gut für die Displaygröße des aufrufenden Gerätes passen. Krzysztof Gajos stellt in [Gajos04] einen Algorithmus zur automatischen Generierung von Benutzerschnittstellen im Kontext des ubiquitous computing vor – dabei handelt es sich um eine ähnliche Problemstellung, so dass der Algorithmus vielleicht für diese Arbeit in adaptierter Form Verwendung finden kann.

## **Werkzeuge**

Da sowohl das DCF als auch DELI in Java implementiert sind, wird auch für die Implementation von DiaGen Java zum Einsatz kommen, um eine reibungslose Integration zu gewährleisten.

Aufgrund der Qualitäten von Python im Bereich des Rapid Prototyping kann – soweit sinnvoll – Jython eingesetzt werden. Dabei handelt es sich um eine Implementierung von Python auf der Java Virtual Machine, welche vollen Zugriff auf die Java-API ermöglicht und somit die Vorzüge beider Sprachen – Python und Java – miteinander vereint.

Python ist eine interpretierte Sprache, dadurch entfallen zeitraubende Kompilierungszyklen bei der Entwicklung. Ausserdem zeichnet sich Python durch seinen kompakten Code aus – Quelltexte in Python sind meist 3 bis 5 mal kürzer

als ihr Java-Äquivalent –, und dieser weist ausserdem ein hohes Maß an Lesbarkeit und Verständlichkeit auf. Diese Eigenschaften wirken sich maßgeblich auf die Effizienz der Entwicklung und die spätere Wartbarkeit aus.

## Geplantes Vorgehen

Für die Bearbeitung dieser Aufgabenstellung ist folgender Zeitplan vorgesehen (in Klammern ist dabei zu jedem Arbeitsschritt eine grobe Zeiteinschätzung in Wochen angegeben):

- Thema festlegen
- Literaturrecherche (1W)
- thematische Einarbeitung (2W)
- Proposal schreiben (1W)
- **Arbeit anmelden (Meilenstein M1)**
- generelle Literatur bearbeiten (1W)
- vorhandene Systeme untersuchen (2W)
- Auswahl und Anpassung der abstrakten Dialogspezifikationsprache (1W)
- Entwurf der Gesamtarchitektur und der Teilsysteme (3W)
- **Beginn Entwicklungsphase (Meilenstein M2)**
- Setup Entwicklungsumgebung und Einarbeitung (1W)
- Entwicklung des Eigenschaften-Ermittlungs-Teilsystems (1W)
- Entwicklung des Dialog-Generierungs-Teilsystems (3W)
- Entwicklung des Eingabe-Verifikations-Teilsystems (1W)
- Entwicklung des Gesamtsystems (2W)
- Integration in DCF (1W)
- **Abschluß Entwicklungsphase (Meilenstein M3)**
- Demo-Anwendung erstellen (2W)
- Vortrag im Diplomantenseminar (1W)
- schriftliche Arbeit überarbeiten (2W)
- **Abgabe der Arbeit (Meilenstein M4)**

Parallel zu allen Arbeitsschritten wird jeweils die entsprechende Literatur bearbeitet und die schriftliche Arbeit angefertigt.



## **Stand der Arbeit**

Mit der Fertigstellung dieses Proposals ist M1 erreicht. Damit ist die Literaturrecherche weitestgehend abgeschlossen und eine erste Einarbeitung in die Problemstellung erfolgt. Dazu wurden erste Arbeiten der gefundenen Literatur gelesen und initiale Lösungsansätze aufgestellt.

Als nächstes muss – wie aus dem Zeitplan zu ersehen ist – die generelle Literatur, welche sich nicht konkret mit einem der Teilprobleme befaßt, weiter bearbeitet werden und die vorhandenen Software-Systeme, welche mit DiaGen interagieren, untersucht werden – dabei handelt es sich also vor allem um das DCF und DELI. Anhand des durch diese Schritte gewonnenen Wissens kann dann die konkrete Konzeption des Gesamtsystems erfolgen.

## Literaturliste

- [Butler02a] Mark Butler, Fabio Giannetti, Roger Gimson, Tony Wiley: „Device Independence and the Web“, Information Infrastructure Laboratory, HP Laboratories Bristol; HPL-2002-249  
IEEE Internet Computing - September/October 2002 - pp 81-86  
© Copyright IEEE 2002
- [Butler01b] Mark H. Butler : „Current Technologies for Device Independence“, Publishing Systems and Solutions Laboratory, HP Laboratories Bristol; HPL-2001-83  
© Copyright Hewlett-Packard Company 2001
- [Giannetti02] Fabio Giannetti: „Device Independence Web Application Framework (DIWAF)“, Information Infrastructure Laboratory, HP Laboratories Bristol; HPL-2002-264  
W3C Device Independence Authoring Techniques Workshop, St Leon, Germany, 25-6 September, 2002  
© Copyright Hewlett-Packard Company 2002
- [Lemlouma02b] Tayeb Lemlouma and Nabil Laya: „Device Independent Principles for Adapted Content Delivery“, OPERA Project Zirst, Research Report <http://opera.inrialpes.fr/people/Tayeb.Lemlouma/Papers/DIReport1.pdf>
- [Wiecha90] Charles Wiecha and Stephen Boies: „Generating User Interfaces: Principles and Use of its Style Rules“, IBM T.J. Watson Research Center; 1990  
© ACM 089791-410-4/90/0010/0021
- [Gajos04] Krzysztof Gajos, Daniel S. Weld: „SUPPLE: Automatically Generating User Interfaces“, University of Washington; IUI 04, January 13-16, 2004, Madeira, Funchal, Portugal. 2004  
© ACM 1581138156/ 04/0001
- [Garvey03] Patrick Garvey (Talaris Corporation), Bill French (University of California, Berkeley - School of Information Management and Systems): „Generating User Interfaces from Composite Schemas“, Talaris Corporation <http://www.idealliance.org/xmlusa/03/call/xmlpapers/03-03-04.994/03-03-04.html>
- [Bullerdick01] Christian Bullerdick: „Diplomarbeit DAGGer - Draft Application GUI Generator Ein Werkzeug für die automatisierte Erstellung graphischer Benutzeroberflächen anhand von schematischen Dialogvorlagen und Datenmodellbeschreibungen“, Diplomarbeit am Lehrstuhl 10, Software-Technologie des Fachbereichs Informatik der Universität Dortmund, März 2001
- [Kolari02] Juha Kolari, Timo Laakko, Eija Kaasinen, Matti Aaltonen, Tapio Hiltunen, Eija-Liisa Kasesniemi, Minna Kulju, Raisa Suihkonen: „Net in Pocket? Personal mobile access to web services“, VTT Information Technology, VTT Publications 464, ESPO 2002
- [Butler02b] Mark H. Butler: „Some Questions and Answers on CC/PP and UAProf“, Information Infrastructure Laboratory, HP Laboratories Bristol; HPL-2002-73  
© Copyright Hewlett-Packard Company 2002

- [Butler01c] Mark H. Butler : „Implementing Content Negotiation using CC/PP and WAP UAProf“, Information Infrastructure Laboratory, HP Laboratories Bristol; HPL-2001-190  
© Copyright Hewlett-Packard Company 2001
- [Butler01a] Mark H. Butler: „DELI: A DELivery context LIBrary for CC/PP and UAProf“, External Technical Report HPL-2001-260, 2001
- [Book/a] Matthias Book, Volker Gruhn: „A Dialog Control Framework for Hypertext-Based Applications“, Dept. of Computer Science, University of Leipzig, Germany
- [Book/b] Matthias Book, Volker Gruhn: „A Dialog Flow Notation for Web-based Applications“, Dept. of Computer Science, University of Leipzig, Germany
- [Schilit02] Bill N. Schilit (Intel Research), Jonathan Trevor. David M. Hilbert (FX Palo Alto Laboratory), Tzu Khiau Koh (Xerox Singapore): „Web Interaction Using Very Small Internet Devices“  
© 2002 IEEE 0018-9162/02/\$17.00
- [Jones03] Matt Jones, Gary Marsden Norliza Mohd-Nasir, Kevin Boone, George Buchanan: „Improving Web Interaction on Small Displays“, Interaction Design Centre, School of Computing Science, Middlesex University, UK, 2003  
<http://www8.org/w8-papers/1b-multimedia/improving/improving.html>
- [Lemlouma01] Tayeb Lemlouma and Nabil Layaïda: „NAC: A Basic Core for the Adaptation and Negotiation of Multimedia Services“, OPERA Project, INRIA Rhône Alpes, 2001  
<http://opera.inrialpes.fr/people/Tayeb.Lemlouma/Papers/ANegoP.pdf>
- [Lemlouma02a] Tayeb Lemlouma and Nabil Layaïda: „Content Adaptation and Generation Principles for Heterogeneous Clients“ OPERA Project, INRIA Rhône Alpes  
Position Paper for the W3C Workshop on Device Independent Authoring Techniques
- [Grundy02] John Grundy, Biao Yang: „An environment for developing adaptive, multi-device user interfaces“, Department of Electrical and Electronic Engineering and Department of Computer Science University of Auckland, New Zealand  
Fourth Australasian User Interface Conference (AUIC2003),  
© Copyright 2002, Australian Computer Society, Inc.
- [Wang03] Jingtao Wang, Jennifer Mankoff: „Theoretical and Architectural Support for Input Device Adaptation“, Computer Science Division, UC Berkeley  
CUU 03, November 10-11, 2003, Vancouver, British Columbia, Canada.  
© Copyright 2003 ACM 1-58113-701-X/03/0011
- [Dees04] Walter Dees: „Handling Device Diversity through Multi-Level Stylesheets“, Philips Research Laboratories  
IUI 04, Jan. 13 16, 2004, Madeira, Funchal, Portugal.  
ACM 1-58113-815-6/04/0001.
- [Florins04] Murielle Florins, Jean Vanderdonckt: „Graceful Degradation of User Interfaces as a Design Method for Multiplatform Systems“, IAG - School of Management - Université catholique de Louvain

IUI 04, January 13 16, 2004, Madeira, Funchal, Portugal.  
© Copyright 2004 ACM 1-58113-815-6/04/0001

- [Fiala04] Zoltan Fiala, Michael Hinz (Dresden University of Technology), Geert-Jan Houben, Flavius Frasinca (Technische Universiteit Eindhoven): „Design and Implementation of Componentbased Adaptive Web Presentations“  
SAC 04, March 1417, 2004, Nicosia, Cyprus  
© Copyright 2004 ACM 1581138121/ 03/04
- [Gaedke98] Gaedke et al., „Web Content Delivery to Heterogeneous Platforms“,  
Lecture Notes in Computer Science (LNCS), Springer Verlag, vol. 1552, 1998
- [Suryanarayana02] Lalitha Suryanarayana (SBC Technologies Resources, USA),  
Johan Hjelm (Ericsson Research, Sweden): „Profiles for the Situated Web“  
WWW 2002, May 7-11, 2002, Honolulu, Hawaii, USA.  
ACM 1-58113-449-5/02/0005.
- [Han00] Richard Han, Veronique Perret, Mahmoud Naghshineh: „WebSplitter: A Unified XML Framework for Multi-Device Collaborative Web Browsing“, IBM Thomas J. Watson Research Center, USA  
CSCW 00, December 2-6, 2000, Philadelphia, PA.  
© Copyright 2000 ACM 1-58113-222-0/00/0012
- [Trewin03] Shari Trewin (IBM T.J. Watson Research Center, USA), Gottfried Zimmermann, Gregg Vanderheiden (Trace Center, Univ. Wisconsin-Madison, USA): „Abstract User Interface Representations: How Well do they Support Universal Access?“  
CUU 03, November 10-11, 2003, Vancouver, British Columbia, Canada.  
© Copyright 2003 ACM 1-58113-701-X/03/0011
- [Rees02] Michael J. Rees: „Evolving the Browser Towards a Standard User Interface Architecture“, School of Information Technology, Bond University, Australia  
Third Australasian User Interfaces Conference (AUIC2002), Melbourne, Australia. Conferences in Research and Practice in Information Technology, Vol. 7. John Grundy stages  
© Copyright 2001, Australian Computer Society Inc.
- [Buchanan01] George Buchanan, Sarah Farrant, Matt Jones, Harold Thimbleby (Middlesex University, UK), Gary Marsden (Dept of Computer Science, University of Cape Town, South Africa), Michael Pazzani (AdaptiveInfo, USA): „Improving Mobile Internet Usability“  
WWW10, May 1-5, 2001, Hong Kong, China.  
ACM 1-58113-348-0/01/0005.
- [Ball00] Ball, T., Colby, C., Danielsen, P., Jagadeesan, L.J., Jagadeesan, R., Läufer, K., Mataga, P., Rehor, K.: „Sisl: Several Interfaces, Single Logic.“,  
International Journal of Speech Technology 3, 2 (June 2000), 91-106. Kluwer Academic Publishers, 2000
- [W3C04a] W3C: „Authoring Techniques for Device Independence“  
<http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/>  
W3C Working Group Note 18 February 2004
- [W3C03a] W3C: „Device Independence Principles“  
<http://www.w3.org/TR/2003/NOTE-di-princ-20030901/>  
W3C Working Group Note 01 September 2003

- [W3C04b] W3C: „Authoring Challenges for Device Independence“  
<http://www.w3.org/TR/2003/NOTE-acdi-20030901/>  
W3C Working Group Note 1 September 2003
- [CC/PP] W3C: „Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0“  
<http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/>  
W3C Recommendation 15 January 2004
- [UAProf] Wireless Application Group: „WAG UAPROF Version 1.0-Nov-1999 User Agent Profile Specification“  
© Wireless Application Protocol Forum, Ltd. 1999.
- [XHTML-MP] Wireless Application Group: „XHTML Mobile Profile Version 29-Oct-2001 Wireless Application Protocol“  
WAP-277-XHTMLMP-20011029-a, Version 29-Oct-2001  
© 2001, Wireless Application Protocol Forum, Ltd.
- [WML] Wireless Application Group: „WAP WML Version 16-Jun-1999 Wireless Application Protocol Wireless Markup Language Specification Version 1.1“  
<http://www.wapforum.org/what/technical/SPEC-WML-19990616.pdf>  
© Copyright Wireless Application Protocol Forum, Ltd.
- [XForms] W3C: „XForms 1.0“  
<http://www.w3.org/TR/2003/REC-xforms-20031014/>  
W3C Recommendation 14 October 2003
- [Puerta01] Angel Puerta, Jacob Eisenstein: „XIML: A Universal Language for User Interfaces“, RedWhale Software  
<http://www.ximl.org/About/BasicDocs.asp>  
© 2001 RedWhale Software.
- [UIML] OASIS Open: „User Interface Markup Language (UIML) Specification“, Working Draft 3.1, 11 March 2004  
<http://www.oasis-open.org/committees/download.php/5937/uiml-core-3.1-draft-01-20040311.pdf>  
© Copyright OASIS Open 2004
- [OASIS03] OASIS Open: „The Relationship of the UIML 3.0 Spec. to Other Standards/Working Groups“, January 28, 2003  
<http://lists.oasis-open.org/archives/uiml/200306/doc00000.doc>
- [Seshadri99] Govind Seshadri: „Understanding JavaServer Pages Model 2 architecture - Exploring the MVC design pattern“  
<http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc.html>
- [W3C-DI] W3C Device Independence Working Group  
<http://www.w3.org/2001/di/#wg>
- [Ximl.Org] XIML Forum  
<http://www.ximl.org>
- [WURFL] Wireless Universal Resource File  
<http://wurfl.sourceforge.net/>
- [UPS] Universal Profiling Schemata  
<http://opera.inrialpes.fr/people/Tayeb.Lemlouma/NegotiationSchema/>