

Human-Computer Interaction Series

Editors-in-chief

John Karat
IBM Thomas J. Watson Research Center (USA)

Jean Vanderdonckt
Université catholique de Louvain (Belgium)

Editorial Board

Gregory Abowd, Georgia Institute of Technology (USA)
Gaëlle Calvary, LIG-University of Grenoble 1 (France)
John Carroll, School of Information Sciences & Technology, Penn State University (USA)
Gilbert Cockton, University of Sunderland (UK)
Mary Czerwinski, Microsoft Research (USA)
Steven Feiner, Columbia University (USA)
Elizabeth Furtado, University of Fortaleza (Brazil)
Kristina Höök, SICS (Sweden)
Robert Jacob, Tufts University (USA)
Robin Jeffries, Google (USA)
Peter Johnson, University of Bath (UK)
Kumiyo Nakakoji, University of Tokyo (Japan)
Philippe Palanque, Université Paul Sabatier (France)
Oscar Pastor, University of Valencia (Spain)
Fabio Paternò, ISTI-CNR (Italy)
Costin Pribeanu, National Institute for Research & Development in Informatics (Romania)
Marilyn Salzman, Salzman Consulting (USA)
Chris Schmandt, Massachusetts Institute of Technology (USA)
Markus Stolze, IBM Zürich (Switzerland)
Gerd Szwillus, Universität Paderborn (Germany)
Manfred Tscheligi, University of Salzburg (Austria)
Gerrit van der Veer, University of Twente (The Netherlands)
Shumin Zhai, IBM Almaden Research Center (USA)

Human-Computer Interaction is a multidisciplinary field focused on human aspects of the development of computer technology. As computer-based technology becomes increasingly pervasive – not just in developed countries, but worldwide – the need to take a human-centered approach in the design and development of this technology becomes ever more important. For roughly 30 years now, researchers and practitioners in computational and behavioral sciences have worked to identify theory and practice that influences the direction of these technologies, and this diverse work makes up the field of human-computer interaction. Broadly speaking, it includes the study of what technology might be able to do for people and how people might interact with the technology.

In this series, we present work which advances the science and technology of developing systems which are both effective and satisfying for people in a wide variety of contexts. The human-computer interaction series will focus on theoretical perspectives (such as formal approaches drawn from a variety of behavioral sciences), practical approaches (such as the techniques for effectively integrating user needs in system development), and social issues (such as the determinants of utility, usability and acceptability).

For further volumes:

<http://www.springer.com/series/6033>

Ahmed Seffah · Jean Vanderdonckt ·
Michel C. Desmarais
Editors

Human-Centered Software Engineering

Software Engineering Models, Patterns
and Architectures for HCI

 Springer

Editors

Ahmed Seffah
Concordia University
Montreal QC H3G 1M8
Canada

Michel C. Desmarais
Ecole Polytechnique de
Montreal
Montreal QC H3C 3A7
Canada

Prof. Dr. Jean Vanderdonckt
Université catholique de Louvain
1348 Louvain-la-Neuve
Belgium

ISSN 1571-5035

ISBN 978-1-84800-906-6

e-ISBN 978-1-84800-907-3

DOI 10.1007/978-1-84800-907-3

Springer Dordrecht Heidelberg London New York

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2009926806

© Springer-Verlag London Limited 2009

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

CONTENTS

List of Figures	xi
List of Tables	xvii
Contributing Authors	xxi
1	
Human-Centered Software Engineering: Software Engineering Architectures, Patterns, and Models for Human Computer Interaction	1
<i>Ahmed Seffah, Jean Vanderdonckt, and Michel C. Desmarais</i>	
1.1 Scope	1
1.2 Specific Objectives of the CHISE Volume II	2
1.3 Overview	2
1.4 Chapter Summaries	3
References	6
Part I User Experiences, Usability Requirements, and Design	
2	
What Drives Software Development: Bridging the Gap Between Software and Usability Engineering	9
<i>Nuno J. Nunes</i>	
2.1 Introduction	9
2.2 Use Case Driven Software Development	11
2.3 Architecture Centric	14
2.4 From Essential Use Cases to the Conceptual Architecture	17
2.5 Tool Issues	20
2.6 Conclusion	23
References	24
3	
Human Activity Modeling: Toward a Pragmatic Integration of Activity Theory and Usage-Centered Design	27

Larry L. Constantine

3.1	Introduction	28
3.2	Activity Theory	29
3.3	Usage-Centered Design	31
3.4	Toward Integration	33
3.5	Human Activity Modeling	35
3.6	Design Implications	43
3.7	Process Implications	44
3.8	Application	45
3.9	Discussion	47
	References	50

4

A	User-Centered Framework for Deriving a Conceptual Design from User Experiences: Leveraging Personas and Patterns to Create Usable Designs	53
---	---	----

Homa Javahery, Alexander Deichman, Ahmed Seffah, and Mohamed Taleb

4.1	Introduction	54
4.2	A First Look at the Proposed Framework	55
4.3	Modeling User Experiences with Personas	56
4.4	Creating a Conceptual Design Using Patterns	57
4.5	An Illustrative Case Study	61
4.6	A Detailed Description of UX-process	70
4.7	Further Investigation: The P2P Mapper Tool	74
4.8	Conclusion	76
	References	79

5

	XML-Based Tools for Creating, Mapping, and Transforming Usability Engineering Requirements	83
--	--	----

Fei Huang, Jon Titus, Allan Wolinski, Kevin Schneider, and Jim A. Carter

5.1	Introduction	83
5.2	Toolset Overview	85
5.3	Using XML to Structure UE Specifications	89
5.4	Mapping Between XML-based UE and SE Specifications	92
5.5	Translating Between XML-based UE Requirements Into SE Specifications	99
5.6	Conclusion	102
	References	102

Part II Modeling and Model-Driven Engineering

6

	MultiPath Transformational Development of User Interfaces with Graph Transformations	107
--	--	-----

Quentin Limbourg and Jean Vanderdonckt

6.1	Introduction	108
6.2	Related Work	110
6.3	Expressing the UI Development Cycle with Graph Transformations	111
6.4	Development Paths	118
6.5	Conclusion	134

References	135
7	
Human-Centered Engineering with UIML	139
<i>James Helms, Robbie Schaefer, Kris Luyten, Jo Vermeulen, Marc Abrams, Adrien Coyette, and Jean Vanderdonckt</i>	
7.1 Introduction	140
7.2 UIML: An Overview	141
7.3 Tools for and Extensions of UIML	148
7.4 Improvements to UIML for Version 4.0	156
7.5 UIML-Related Standards	166
7.6 Conclusion	169
References	170
8	
Megamodeling and Metamodel-Driven Engineering for Plastic User Interfaces: Mega-UI	173
<i>Jean-Sébastien Sottet, Gaëlle Calvary, Jean-Marie Favre, and Joëlle Coutaz</i>	
8.1 Introduction	174
8.2 Plasticity: Case Study and Engineering Issues	175
8.3 Modeling, Metamodeling, and Megamodeling	182
8.4 MDE for Plasticity	190
8.5 Conclusion and Perspectives	196
References	197
9	
Cause and Effect in User Interface Development	201
<i>Ebba Thora Hvannberg</i>	
9.1 Introduction	201
9.2 Research Study	205
9.3 Eliciting Needs and Context	209
9.4 Design	210
9.5 Evaluation in Context	214
9.6 Foundation and Context of an Evaluation Model	215
9.7 Conclusion	218
References	219
Part III Interactive Systems Architectures	
10	
From User Interface Usability to the Overall Usability of Interactive Systems: Adding Usability in System Architecture	225
<i>Mohamed Taleb, Ahmed Seffah, and Daniel Engleberg</i>	
10.1 Introduction	226
10.2 Background and Related Work	227
10.3 Identifying and Categorizing Typical Scenarios	228
10.4 Patterns as Solutions to the Problems Documented as Scenarios	230
10.5 Modeling Cause-Effect Relationships Between Software Elements and Usability	237
10.6 Conclusion and Future Investigations	242

References	243
11	
Toward a Refined Paradigm for Architecting Usable Systems	245
<i>Tamer Rafla, Michel C. Desmarais, and Pierre N. Robillard</i>	
11.1 Introduction	245
11.2 An Overview of Previous Work	246
11.3 Usability at the Requirements Definition Stage	248
11.4 Usability-Centered Software Development Process	251
11.5 Conclusion	253
References	254
12	
Trace-Based Usability Evaluation Using Aspect-Oriented Programming and Agent-Based Software Architecture	257
<i>Jean-Claude Tarby, Houcine Ezzedine, and Christophe Kolski</i>	
12.1 Introduction	257
12.2 First Approach for Early Usability Evaluation: Injection of the Mechanism of Traces by Aspect-Oriented Programming	258
12.3 Second Approach: Interactive Agent-Based Architecture and Evaluation Module	263
12.4 Towards an Assistance System for the Evaluation of Agent-Based Interactive Systems	266
12.5 Comparison Between the two Approaches	268
12.6 Conclusion	273
References	274
13	
Achieving Usability of Adaptable Software: The AMF-Based Approach	277
<i>Franck Tarpin-Bernard, Kinan Samaan, Bertrand David</i>	
13.1 Introduction	277
13.2 State-of-the-art	278
13.3 AMF and Its Relationships With Other Models	281
13.4 A Method for Designing Adaptable Applications	289
13.5 Future Developments and Conclusion	294
References	295
Part IV Reengineering, Reverse Engineering, and Refactoring	
14	
The GAINS Design Process:	301
How to do Structured Design of User Interfaces in any Software Environment	
<i>Martha J. Lindeman</i>	
14.1 The Costs of Changing User Interfaces	302
14.2 Overview of the GAINS Process	304
14.3 Overview of XP's Planning Levels	305
14.4 Evaluations of Usability	313
14.5 Difficulties With Two XP Assumptions	314
14.6 Conclusions	314
References	315

15		
Legacy Systems Interaction Reengineering		317
<i>Mohammad El-Ramly, Eleni Stroulia, and Hani Samir</i>		
15.1	Introduction	318
15.2	Motivation for Interaction Engineers	318
15.3	Generic Methodology	319
15.4	Applications of Interaction Reengineering	323
15.5	From Websites to Web Services	325
15.6	Advantages and Limitations	331
	References	332
16		
Reverse Engineering for Usability Evaluation		335
<i>Atif M. Memon</i>		
16.1	Introduction	335
16.2	GUI Model	338
16.3	Design of the GUI Ripper	345
16.4	Implementation	347
16.5	Empirical Evaluation	349
16.6	Related Work	351
16.7	Conclusions and Future Work	352
	References	353
17		
Task Models and System Models as a Bridge between HCI and SE		357
<i>David Navarre, Philippe Palanque, and Marco Winckler</i>		
17.1	Introduction	358
17.2	Related Work	359
17.3	Why a Task Model is Not Enough	359
17.4	A Classical System Model	361
17.5	The Improved System Model	362
17.6	Scenarios as a Bridge Between Tasks and System Models	363
17.7	A Case Study	365
17.8	The Integration of the Models: CTT-ICO	375
17.9	Conclusions	382
	References	384
Authors Index		387
Subject Index		395

List of Figures

2.1	Multiple use case descriptions for the withdraw money use case in the TaskSketch tool	13
2.2	Application example of the UP architecture versus a revised proposal for interactive systems: top — transcription of a solution provided in Conallen (1999); bottom — the new solution based on Nunes and Cunha (2000)	15
2.3	From task flows to the system architecture: an example for a simplified bank system	18
2.4	Semi-automatic traceability support in TaskSketch	19
2.5	Tool usage based on a survey of 370 practitioners (Campos and Nunes, 2007a)	21
2.6	Workstyle transition frequency and cost (Campos and Nunes, 2007)	22
2.7	Frequency distribution of the several views in TaskSketch (recorded through automatic logging)	22
3.1	The structure of human activity (adapted from Engeström, 1999)	30
3.2	Hierarchical nature of activity (adapted from Engeström et al., 1999).	30
3.3	Example of an Activity Map in a retail sales context	39
3.4	Example of Participation Map for retail selling situation	41
3.5	Example of partial Activity-Task Map for retail selling	43
3.6	Logical overview of usage-centered design with activity modeling	46
3.7	System-centered Participation Map for collaborative usability inspections	47
3.8	Performance Map for the UInspect collaborative usability inspections tool	48
4.1	Current problem of deriving a design from user experiences	55
4.2	The proposed framework	56
4.3	Example of persona	59
4.4	A Site Map page implemented using Tree Hyperbolic, a sophisticated visualization pattern	60
4.5	A Home Page Design combining several patterns	61
		xi

4.6	Questionnaire results of Novice and Expert users	64
4.7	Pattern Skeleton of NCBI home page	67
4.8	Pattern-Oriented Design of NCBI home page	68
4.9	The UX-P Design Process	71
4.10	Website structure using three basic information patterns	73
4.11	Clustering in P2P Mapper	75
4.12	Overview of P2P Mapper Tool	77
5.1	The structure of the USERLab toolset and its databases	86
5.2	Creating a requirement template record with the MAT	91
5.3	The interactions of the Methodology Mapping Tool	93
5.4	Basic interactions of the Methodology Mapping Tool	100
5.5	An integration and translation tool	100
6.1	Overall framework of our methodology	114
6.2	Transformations between viewpoints	117
6.3	A transformation system in our methodology	118
6.4	Framework for model transformations	118
6.5	Transformation paths, step and substep	119
6.6	Forward transformational development of UIs	120
6.7	Creation of abstract individual components derived from task model leaves	121
6.8	Creation of abstract containers derived from task model structure	121
6.9	Creation of a facet for an abstract individual component derived from task action type	121
6.10	A sequentialization of abstract individual component derived from task temporal relationships	122
6.11	A placement of abstract container derived from task temporal relationships	122
6.12	A creation of windows derived from containment relationships at the abstract level	123
6.13	A generation of window structure derived from containment relationship at the abstract level	124
6.14	Creation of an editable text component (i.e., an input field) derived from facets type of abstract components	124
6.15	A placement of graphical individual components derived from spatiotemporal relationships at the abstract level	125
6.16	A window navigation definition derived from spatiotemporal relationships at the abstract level	125
6.17	Reverse transformational development of UIs	126
6.18	Creation of a facet at the abstract level derived from a type analysis of graphical individual components	127
6.19	Definition of task action types derived from an analysis of facets at the abstract level	128
6.20	Context adaptation at different levels of our framework	128

6.21	Flattening of a task tree structure	129
6.22	Transforming all temporal relationship to concurrent	129
6.23	A merging of facets of abstract individual components	130
6.24	Erasing abstract individual components with no facets left	130
6.25	Initializing of the adaptation process by creating graphical component to adapt into	131
6.26	Creation of tabbed item and transfer of the content of the adapted window	131
6.27	Deletion of unnecessary containers	132
6.28	Merging of a non-editable text component (e.g., a label) and an editable text component (e.g., an input field) into one single editable text component	132
6.29	Squeezing of a layout structure to display vertically	133
7.1	The Meta-Interface Model	145
7.2	The UIML specifications for the “Hello World!” example	146
7.3	Graphical overview of abstraction by using vocabularies	147
7.4	Listing of mapping of a date part on types of concrete widgets	148
7.5	A general screen-shot of the sketching tool	151
7.6	Various alternative representations of the same widget (here, a slider)	151
7.7	Example interface sketch	152
7.8	An example of a UI for a PDA and its corresponding UIML code	152
7.9	UIML code of the simple UI state machine example	158
7.10	Room reservation form and corresponding state machine	159
7.11	Part of the UIML code for the room reservation	159
7.12	Fully functional music player interface rendered on desktop computer from UIML document	162
7.13	Fully functional music player interface rendered on PDA from UIML document	163
8.1	Four functionally equivalent UIs that differ from the set of usability criteria	177
8.2	A hand made mock-up similar to Figure 8.1d, but that improves the Compatibility criterion thanks to a clickable map of the house	178
8.3	A distributed version of HHCS: selecting a room is done on the PDA the temperature is specified on the PC	178
8.4	A CTT task model and an UML domain model of the case study	181
8.5	Self-description is three-fold. It covers the design rationale (i.e., the design process transforming requirements into a resulting interactive system (IS)), the resulting IS itself (i.e., a net of mappings linking together different perspectives in a consistent way), and the evolution process for supporting the switch to another interactive system.	182
8.6	The pyramid of models and the pyramid of actors	186
8.7	An example of megamodel charting the basic concepts and relations in MDE	190

8.8	The megamodel coverage	191
8.9	An excerpt of the megamodel for HCI. The coverage is limited to UI. Neither the functional core nor the context of use are considered	192
8.10	A mapping metamodel for general purpose. The composition between Mapping and Metamodel is due to EMF (Eclipse Modeling Framework)	193
8.11	Examples of mappings in HHCS. Mappings tell the usability properties they satisfy. Here properties are expressed in Bastien-Scapin's framework	194
8.12	Extra-UIs for observing and controlling the tasks distribution on platforms	195
8.13	Extra- and Meta-UIs	196
8.14	Beyond UIs: Mega-UIs	197
9.1	Cause in Design and Effect in Problem Domain	203
9.2	Development model	216
10.1	The roles of the MVC architecture components	228
10.2	Traditional "twin towers" model of usability and other software quality factors	238
10.3	Revised model of usability, including possible types of cross-relationships with architecture (bold links)	240
10.4	Most probable types of cross-relationships between usability and architecture (bold links)	241
11.1	Three usability layers of Folmer's framework	248
11.2	The requirements elicitation workflow	253
11.3	Extra activity in "understanding users' needs" workflow detail	254
12.1	AOP basic principles compared to traditional object oriented-approach	259
12.2	Injection of mechanism of traces by aspects	261
12.3	Global agent-oriented architecture for interactive systems	264
12.4	Principle of coupling between agent-based architecture of the interactive system and its evaluation (Trabelsi, 2006)	265
12.5	Architecture of the evaluation assistance system	266
12.6	Association with each presentation agent of an informer agent	267
12.7	Evaluation assistance system	269
13.1	Arch model	280
13.2	The main AMF administrators	282
13.3	A simple interaction described with AMF	283
13.4	The AMF and the Arch models links	283
13.5	AMF meta-model	284
13.6	Links between AMF and the applicative classes (concrete facets)	284
13.7	The AMF editor (Architecture edition)	285
13.8	The AMF editor (Generated control visualization)	286
13.9	Example of relationships between tasks and ports in the interaction model	288

13.10	Example of concretization of an abstract task and the associated abstract port	289
13.11	A pattern for the abstract task “ Select and Move an element ”	290
13.12	Task and domain models after the first design step (music player sample)	291
14.1	Users’ goals and roles for the bulletin-board example	305
15.1	A general methodology for interaction reengineering	322
15.2	The CelLEST UI reengineering process	325
15.3	Interaction reengineering of websites to web services (Jiang and Stroulia, 2004)	326
15.4	Interaction reengineering of form-based CUIs to Web services	329
15.5	System architecture for form-based CUI reengineering to Web services (Canfora et al., 2006)	330
16.1	(a) Open window, (b) its Partial State	340
16.2	Examples of GUI forests	340
16.3	GUI Forest (Tree) for MS WordPad	342
16.4	Partial event-flow graph of MS WordPad	344
16.5	Visiting each node of a forest	345
16.6	Traversing and extracting the GUI of an application	347
17.1	Petri net representation of the concrete task for playing the game of 15	361
17.2	Presentation part of a basic system	362
17.3	Behavior of the basic system for the game of 15	362
17.4	Examples for the improved system model	363
17.5	A screen-shot of the radar screen with planes (left-hand side, one of the planes 1123 is assumed)	367
17.6	The abstract task model of case study using CTT notation	367
17.7	The concrete and detailed task model of the case study	368
17.8	CTTE for extracting scenarios	370
17.9	Tools available for designers in PetShop environment	372
17.10	IDL description of the class “MefistoPlaneManager”	372
17.11	ObCS description of the class “MefistoPlaneManager”	373
17.12	Rendering function of the class “MefistoPlaneManager”	374
17.13	The activation function of the class “MefistoPlaneManager”	374
17.14	IDL description of the class “MefistoPlane”	374
17.15	ObCS of the class “MefistoPlane”	375
17.16	The presentation part of the class “MefistoPlane”	375
17.17	The rendering function of the class “MefistoPlane”	376
17.18	IDL description of the class “MefistoMenu”	376
17.19	ObCS of the class “MefistoMenu”	376
17.20	The presentation part of the class “MefistoMenu”	377
17.21	Rendering function of the class “MefistoMenu”	377
17.22	The activation function of the class “MefistoMenu”	377

17.23	The framework for CTTE – PetShop integration	378
17.24	Association of interactive tasks and user services	380
17.25	The scenario player	381
17.26	Execution of the scenario of the system	381
17.27	Interaction between scenario and current execution: plane IDs are selected at runtime	382
17.28	Interaction between scenario and current execution: values for frequency are selected at runtime	383
17.29	End of the execution of the scenario on the system model	383

List of Tables

3.1	Extended usage-centered design notation for activity modeling	36
3.2	Relationships between activities	38
4.1	Persona elements	58
4.2	Pattern Selection based on persona	66
4.3	UX-P Process steps and tool support	74
5.1	MAT-created XML for a PUF Task Record's Identification Information section	92
5.2	MAT-created XML for a PUF Task Record's Linkage Information Section	93
5.3	PUF UML tags and their relationship to UML	94
5.4	Some high-level mapping of entities from PUF to UML	95
5.5	High-level mapping of PUF tasks to UML actors	96
5.6	Direct attribute-to-attribute mapping of PUF to UML	96
5.7	Attribute-to-attribute part mapping of PUF to UML	97
5.8	Attribute to new attribute mapping of PUF to UML	98
6.1	A comparison of terms used in MDA and our methodology	116
7.2	Definition of LiquidApps TM components	149
9.1	Methods used in case studies	206
9.2	Requirements for the design of an evaluation model	217
9.3	Requirements for a design model	217
10.1	Example of design patterns	233
10.2	A partial vision of the consolidated ISO 9126 measurement framework	239
10.3	Examples of relationships between invisible software entities and usability factors	242
11.1	Canceling commands (Bass and John, 2003)	247
11.2	Providing feedback property (Folmer et al., 2003)	251
11.3	Summary of observations	252
12.1	Comparison between the two approaches	270
12.1	Comparison between the two approaches (continued)	271

12.1	Comparison between the two approaches (continued)	272
14.1	List of users' goals and roles identified in the Statements of Purpose	310
14.2	First-draft set of primary goals for all users of the bulletin-board system	310
16.1	Event types in some GUIs	343
16.2	Time for ripping windows and Java applications	350
16.3	Structural attributes (GUI forest)	351
16.4	Usability attributes for event-flow	352

To my mentors, professors and friends Dr Rabia
Khaled and Pr Bertrand Tim David For their
support, advice and guidance.
Ahmed

To my wife Teodora and our daughter Marina.
Jean

To those aging loved ones,
may we comfort them as well as
they nurtured us.
Michel

Contributing Authors

Marc Abrams serves as Harmonia's President and CTO. In the past, Dr. Abrams has been with the former U.S. Army Concepts Analysis Agency, a post-doc in the Distributed Operating Systems group in Stanford's Computer Science Department, a Visiting Scientist in the network protocol group at IBM's Zurich Research Laboratory in Switzerland, and an Associate Professor with Tenure in Computer Science at Virginia Tech in Blacksburg, Virginia. He has been the Principal Investigator for almost \$20M in research and development projects with IBM, MDA, NAVAIR, NAVSEA, NSF, ONR, OSD, Raytheon, and SAIC. He received his Ph.D. from the University of Maryland at College Park in Computer Science. Before Harmonia, Dr. Abrams was a tenured associate professor at Virginia Tech, where his research on Human-Computer Interfaces led to the creation of UIML and later the co-founding of Harmonia. UIML forms the basis for Harmonia's LiquidApps® product. At Virginia Tech, he also co-founded the Center for Human Computer Interaction, and worked with faculty in HCI in fields ranging from cognitive psychology to human factors on scenario-driven HCI design.

Gaëlle Calvary has been an assistant professor at UJF since 1999, and a member of the HCI research group headed by Joëlle Coutaz. Her research area is UI plasticity. She aims at providing models, methods and tools for supporting the engineering of plastic UIs capable of overcoming multiple, dynamic, and unforeseeable contexts of use. Her approach combines MDE and AI techniques.

Jim A. Carter is a Professor and Director of the Usability Engineering Research Lab (USERLab) of the Computer Science Department, University of Saskatchewan. His research investigates methods and methodologies for improving usability and accessibility and for incorporating these concerns within traditional software engineering methods and methodologies. Jim is involved as a Canadian expert in the development of ergonomic, software engineering, user interface, and accessibility standards within ISO and ISO/IEC JTC1.

Larry L. Constantine is Chief Scientist Constantine & Lockwood, Ltd, as well as an ACM Distinguished Engineer and one of the pioneers of software design whose current work in usage-centered design continues a long tradition of technical innovation and professional leadership. In a career spanning four decades, he has contributed numerous concepts and techniques forming the foundations of modern practice in software engineering and applications development. Constantine has keynoted numerous major international conferences and has taught in 20 countries around the world. He is a prolific writer and a widely read columnist. Among his publications in both the computer sciences and human sciences are over 150 articles and papers (see bibliography) plus 17 books, including the award-winning *Software for Use* (Addison-Wesley, 1999), written with Lucy Lockwood; *The Peopleware Papers* (Prentice Hall, 2001); and the software engineering classic, *Structured Design* (Prentice Hall, 1979), written with Ed Yourdon. His books and papers have been translated into Chinese, Danish, Dutch, German, Italian, Japanese, Portuguese, Russian, and Spanish. Constantine is a professor in the Department of Mathematics and Engineering at the University of Madeira, Funchal (Portugal) where he is also Director of LabUSE: the Laboratory for Usage-centered Software Engineering. He also served on the faculty of the University of Technology, Sydney (Australia), where he was Professor of Information Technology. He is a graduate of the Massachusetts Institute of Technology, Sloan School of Management.

Joëlle Coutaz has been professor at UJF since 1973 and the founder (in 1990) and head of the HCI research group (Ingénierie de l'Interaction Homme-Machine) at Laboratoire d'Informatique de Grenoble. In April 2007, she was elected to the SIGCHI Academy for "leadership in the profession in Computer Human Interaction." In June 2007, she received the Honorary Degree of Doctor of Science from the University of Glasgow.

Adrien Coyette recently earned his Ph.D. in HCI at Université catholique de Louvain. His thesis work won him the Brian Schackel 2007 award for the most outstanding contribution with international impact in HCI. He is currently a member of the research staff at UCL Unité de systèmes d'information in Louvain School of Management.

Bertrand David is a Full Professor of Computer Science at the Ecole Centrale de Lyon (ECL) in Lyon working in the area of human-computer interaction, cooperative systems and wearable computer use in pervasive environments. He was cofounder and director for 8 years of a multidisciplinary research lab ICTT, working on cooperative systems design and evaluation. He is coeditor in chief of RIHM: Francophone Journal of Human-Machine Interaction.

Alexander Deichman is a graduate student with the Human-Centered Software Engineering Group, including the Usability and Empirical Studies Lab, in the Department of Computer Science and Software Engineering at Concordia University. He holds a

Bachelor of Engineering (Computer) from Concordia University, and is now pursuing his studies in Human-Computer Interaction. His research interests lie in designing novel tools that can support the UI design process. He is currently working on a tool called P2P Mapper, which helps designers in building pattern-oriented designs based on user specifications. Alex has been involved in a number of usability projects, mostly for web-based and biomedical applications.

Michel C. Desmarais has been an associate professor of Computer Engineering at École Polytechnique de Montréal, Canada, since 2002. He was the head of the HCI group at the Computer Research Institute of Montreal from 1990 to 1998, and was manager of software engineering teams in a private software company for 4 years afterwards. Besides user-centered engineering, his research interests are in artificial Intelligence and user modeling. http://www.professeurs.polymtl.ca/michel.desmarais/desmarais_michel_c.html

Mohammad El-Ramly completed his Ph.D. studies at Alberta University (Canada). He is a lecturer in Computer Science at the University of Leicester (UK). His research activities include software systems evolution, maintenance, reverse engineering, and reengineering. He is also investigating the applications of data mining to software data.

Daniel Engleberg earned a master's degree in cognitive sciences from the University of Michigan and bachelor of science from McGill University. He is a senior user experiences and usability professional with Nuange Technologies, a Montreal-based IT company. Daniel is active with the Usability Professional Association and has organized several workshops on HCI and Software Engineering Integration.

Houcine Ezzedine is a maître de Conférences at the Université de Valenciennes. He earned his Ph.D. in 1985 and his “Habilitation” in 2002 from the Université de Valenciennes et du Hainaut-Cambrésis. He is the author of over 60 scientific publications and an active member of the Laboratoire d'automatique de Mécanique et d'Informatique industrielles et Humaines et au sein du groupe de recherche RAIHM : Raisonnement Automatique et Interaction Homme-Machine.

Jean-Marie Favre is a Software Explorer and a Language Archaeologist practicing XFORS. He serves as an Assistant Professor at UJF. His research work aims at (1) understanding phenomenon that arises during the long-term evolution of very large scale software products, and (2) to improve software engineering techniques accordingly.

James Helms currently serves as a Software Engineer focused on usability and UI development for Digital Receiver Technology, Inc. Before that, he was responsible for ensuring that Harmonia, Inc. maintained a rigorous usability engineering ethic. He is

the Chair of the OASIS UIML standardization committee and served on the original LiquidUI® product development team. His research background includes the development of a new usability engineering process model (the Wheel), in-depth analysis of usability issues associated with deploying training documents and interfaces to PDAs, and the field evaluation of a distributed, multiuser virtual science laboratory. Mr. Helms received his master's degree in Computer Science and Applications from Virginia Tech in 2001.

Fei Huang is an M.Sc. student majoring in usability engineering. She also has expertise in databases and in accessibility engineering. Fei is a member of the USERLab of the Computer Science Department, University of Saskatchewan.

Ebba Thora Hvannberg is a professor of computer science. She has a B.S. in computer science from the University of Iceland, and an M.S. and a Ph.D. from Rensselaer Polytechnic Institute, New York. Her research interests include human-computer interaction and software engineering.

Homa Javahery is a researcher and project manager with the Human-Centered Software Engineering Group, including the Usability and Empirical Studies Lab, in the Department of Computer Science and Software Engineering at Concordia University. She holds a Ph.D. in Computer Science (Human-Computer Interaction) from Concordia University, and a Bachelor of Science degree from McGill University. She is combining different design approaches from human sciences and engineering disciplines to develop novel design frameworks which effectively address the user's needs and experiences. She has managed a number of large-scale usability studies and participated in collaborative projects at the INRIA Research Institute in France and the Daimler-Chrysler Research Institute in Germany. Her main areas of expertise are biomedical applications and multiple user interfaces.

Christophe Kolski has been a professor in computer science at the Université de Valenciennes et du Hainaut-Cambrésis since 1995. He earned his Ph.D. in 1989 and his "habilitation" in 1995. He is the author of over 200 publications and a member of the Laboratoire d'automatique de Mécanique et d'Informatique industrielles et Humaines et au sein du groupe de recherche RAIHM : Raisonnement Automatique et Interaction Homme-Machine.

Quentin Limbourg earned a Ph.D. in Management Sciences from the Management School (IAG) of the University of Louvain (UCL). His dissertation topic was "Multi-path Development of User Interfaces." Currently he is an IT consultant at the R&D unit of SmalS-MvM, a Scientific collaborator at the Management School (IAG) of the University of Louvain (UCL), as well as a temporary invited lecturer (only for 2005-2006) at the Computer Science Department of the Université Libre de Bruxelles. His is investigating the application of software engineering to user interface development

including Model-Driven Architecture, Conceptual modeling and Languages for user interface specification (e.g., UsiXML).

Martha J. Lindeman is president of Agile Interactions, Inc. in Columbus, Ohio. She has over 20 years of experience in the design and evaluation of how people process information in interactions with other people and technology.

Kris Luyten is working in the area of Human-Computer Interaction, focusing on advanced techniques and technologies that support the design, development, and deployment of traditional and non traditional user interfaces. These include context-sensitive, collaborative, and embedded/ubiquitous user interfaces. He is Assistant Professor at Hasselt University in Belgium and Adjunct group-leader HCI research group, Expertise Centre for Digital Media. He has authored and coauthored over 80 scientific papers and has participated in 25 conference and workshops program committees.

Atif M. Memon is an Associate Professor in the Department of Computer Science, University of Maryland. He received his B.S. in Computer Science in 1991 from the University of Karachi, his M.S. in Computer Science in 1995 from the King Fahd University of Petroleum and Minerals, and his Ph.D. in Computer Science from the University of Pittsburgh. He was awarded a Gold Medal during his undergraduate education, and received fellowships from the Andrew Mellon Foundation for his Ph.D. research. His research interests include program testing, software engineering, artificial intelligence, plan generation, reverse engineering, and program structures. He is a member of the ACM and the IEEE Computer Society and serves on the editorial boards of the *Journal of Software Testing, Verification, and Reliability*, *The Open Software Engineering Journal*, and the *Canadian Journal of Pure and Applied Sciences*. He has served on numerous National Science Foundation panels. He is currently serving on a National Academy of Sciences panel as an expert in the area of Computer Science and Information Technology, for the Pakistan-U.S. Science and Technology Cooperative Program, sponsored by the United States Agency for International Development (USAID).

David Navarre is a Lecturer in Computer Science Human Computer Interaction and Software Engineering at the University Toulouse 1. The main area of his research activities is defining methods and techniques dedicated to real-time safety critical interactive systems.

Nuno J. Nunes is an Associate Professor in the Mathematics and Engineering Department of the University of Madeira where he lectures in the fields of software engineering, programming languages and human-computer interaction. He holds a degree in Software and Computer Engineering (LEIC) from the Instituto Superior Tecnico (1994) of the Technical University of Lisbon, an MPhil in Software Engineering from the University of Madeira (1997), and a Ph.D. (Doctor of Engineering Degree in Sys-

tems and Computer Science) in Software Engineering from the University of Madeira (2001). In 1993 and 1994 he worked at INESC in the areas of telematic systems and services and advanced computer architecture. Since 1995 he shifted his research interests to software engineering and human-computer interaction. He was involved in the organization of several international conferences (IUI/CADUI2004, UML2004 and CAISE2005) and workshops (Wisdom99, Tupis00, DSV-IS2003). He is also the leading author of the Wisdom (Whitewater Interactive System Development with Object Models) lightweight software engineering method, a UML method specifically tailored for small software developing companies. Currently his research interests include User-centered Development, User Interface Design, Object-Oriented Methods, Agile Software Development, and Organizational Engineering. He is a member of the ACM, SIGCHI, SIGSOFT, and the IEEE Computer Society.

Philippe Palanque is full professor of computer science, University of Toulouse III (France). He is the Head of Master 2 Professional in HCI at Toulouse University. His interests cover a broad range of topics at the intersection of HCI and software engineering including critical safety systems.

Tamer Rafla holds a master's degree in software engineering from École Polytechnique de Montréal with his research nominated for the Best Thesis Award. His research interests include HCI, requirements engineering, software architecture and software process. He is currently working as a business consultant for a leading provider of SAP consulting services. He is a member of the Usability Professionals Association, the Worldwide Institute of Software Architects, and the Québec Order of Engineers.

Pierre-N. Robillard is a full professor of software engineering at École Polytechnique de Montréal. He leads the Software Engineering Research Laboratory whose research interests include software process, software cognition, software quality assurance and software applications to bio-informatics. He contributed to over 120 research papers, conference proceedings and three books on software engineering related topics. The latest book is *Software Engineering Process with the UPEDU*, Addison-Wesley, 2003. He is a licensed professional engineer in the province of Quebec and in Canada, and a member of the IEEE, ACM, EACE, and CIPS.

Kinan Samaan received his master's degree in "computer science for the society service." In 2002 he was awarded a Ph.D. in the area of adaptation of human-computer interface and software engineering from the Ecole Centrale de Lyon (France). He is currently the executive manager of Mamoun International Corporation—Aleppo branch, an international institute of higher education.

Robbie Schaefer received his doctoral degree at Paderborn University in 2007, where he has been working as a research assistant since 2001. There, he worked on sev-

eral European projects with a focus on the challenges of user interface development for ambient intelligence. During this time he has authored about 30 publications in this field. His dissertation is titled "Model-Based Development of Multimodal and Multi-Device User Interfaces in Context-Aware Environments." He is active in the standardization of the User Interface Markup Language and a member of the OASIS UIML technical committee. Currently he works for Mettenmeier, a company which focuses on geographical information systems and asset management for gas, water, and electricity suppliers.

Kevin Schneider is Professor, Director of the Software Engineering Lab and Department Head of the Computer Science Department, University of Saskatchewan. His research investigates models, notations, and techniques to support collaborative software evolution. He is an elected member of the IFIP working group 2.7/13.4 on user interface engineering and the Prairie representative for the Canadian Association of Computer Science.

Ahmed Seffah is currently professor of Information and Software Technology at Ecole Hôtelière Lausanne. He was previously an associate professor in the Department of Computer Science and Software Engineering. From 2000 to 2008 he has been the Concordia research chair on human-centered software engineering, a term he coined. His research interests are at the intersection of human-computer interaction, psychology, and software engineering, with an emphasis on usability and quality in use metrics and measurement, human experiences modeling as well as patterns as a vehicle for capturing and incorporating empirically valid best human-centric users and developers' experiences into software engineering processes. He is the cofounder of the Usability and Empirical Studies Lab which provides an advanced facility to support research and development in the field of human-centered software. Dr. Seffah is the vice chair of the IFIP working group on user-centered systems design methodologies and the cochair of the first working conference on Human-Centered Software Engineering.

Jean-Sébastien Sottet is doing a Ph.D. in Computer Science under the supervision of Gaëlle Calvary and Jean-Marie Favre. He is exploring Model Driven Engineering for plastic User Interfaces (UIs), i.e., UIs capable of adapting to their context of use ("User, Platform, Environment") while preserving usability.

Eleni Stroulia is an associate professor, Department of Computer Science (University of Alberta, Canada). Her recent research focuses on two major themes: (a) software understanding and reengineering and (b) reflective agent-based architectures for service-oriented Web-based applications.

Mohamed Taleb is a Ph.D. candidate at Concordia University and Ecole de Technologie Supérieure de Montreal. He has more than 15 years' experience as a senior soft-

ware engineer and HCI expert. His main interests include pattern driven, model based and architecture-centric development approaches for interactive systems engineering.

Jean-Claude Tarby is a maître de Conférences at the Université des Sciences et Technologies de Lille. He earned his Ph.D. in computer science at the Université de Toulouse. He is a member of Equipe NOCE at Laboratoire TRIGONE.

Franck Tarpin-Bernard is an Assistant Professor of Computer Science at the National Institute of Applied Sciences (INSA) in Lyon working in the area of human-computer interaction adaptation. He completed his Ph.D. in 1997 in the area of computer supported-collaborative work and software engineering. In 2000, he cofounded Scientific Brain Training, the current world-leader company of cognitive training software.

Jon Titus is a Database Analyst for the Potash Corporation and a member of the USERLab of the Computer Science Department, University of Saskatchewan.

Jean Vanderdonckt is a Full Professor in Computer Science at Université catholique de Louvain (Belgium), Louvain School of Management (IAG-LSM) where he leads the Belgian Laboratory of Computer-Human Interaction (BCHI, <http://www.isys.ucl.ac.be/bchi>). This laboratory is conducting research, development, and consulting services in the domain of user interface engineering, a domain that is located midway between software engineering, human-computer interaction, and usability engineering. Jean Vanderdonckt is the founder and the coordinator of the UsiXML Consortium (www.usixml.org) that structures activities towards the definition and the usage of UsiXML (User Interface eXtensible Markup Language) as a common User Interface Description Language. He is the coordinator of HCI activities within the Similar network of excellence (www.similar.cc, The European research taskforce creating human-machine interfaces SIMILAR to human-human communication). He is also a member of the European COST n°294 Action MAUSE (www.cost294.org) on usability engineering. He is a senior member of IEEE, ACM, and SIGCHI. He is a professor within the Faculty of Economical, Social and Political Sciences (ESPO), School of Management (IAG), Université Catholique de Louvain La Neuve (Belgium). He also leads the BCHP Laboratory. He has coauthored and edited several books and published over 100 publication in the fields of software engineering and HCI.

Jo Vermeulen is a researcher at the Expertise Centre for Digital Media (EDM), a research institute of Hasselt University. He obtained his M.Sc. in computer science from Hasselt University in 2005 and joined EDM. Jo has been working with the User Interface Markup Language (UIML) since 2004, when he did an internship under the supervision of Kris Luyten to improve the open source Uiml.net renderer. For his M.Sc. thesis, he extended UIML with a declarative, platform-independent layout specification

based on spatial constraints. He has been participating in the UIML Technical Committee since 2005. His research interests lie at the intersection of human-computer interaction and ubiquitous computing. He is currently pursuing a Ph.D. in this topic.

Marco Winckler completed his Ph.D. studies in 2004, focusing on the navigation modeling of complex Web applications. Since then, he has been a senior lecturer at University Toulouse III. His current research mingles Human-Computer Interaction methods and Software Engineering methods applied to the development of Web-based interactive systems.

Allan Wolinski is a Web Developer for Point2 and a member of the USERLab of the Computer Science Department, University of Saskatchewan.