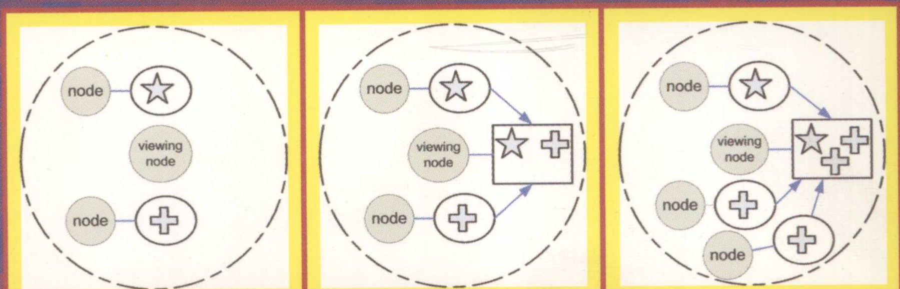
State-of-the-Art Survey

LNCS 3914

Alessandro Garcia
Ricardo Choren
Carlos Lucena
Paolo Giorgini
Tom Holvoet
Alexander Romanovsky (Eds.)

Software Engineering for Multi-Agent Systems IV

Research Issues and Practical Applications



Springer

Alessandro Garcia Ricardo Choren
Carlos Lucena Paolo Giorgini
Tom Holvoet Alexander Romanovsky (Eds.)

Software Engineering for Multi-Agent Systems IV

Research Issues and Practical Applications

Springer

Volume Editors

Alessandro Garcia
Lancaster University, Computing Department, InfoLab 21, Lancaster, LA1 4WA, UK
E-mail: a.garcia@lancaster.ac.uk

Ricardo Choren
SE/8, IME, Praça General Tibúrcio 80, Urca 22290-170, Rio de Janeiro/RJ, Brazil
E-mail: choren@de9.ime.eb.br

Carlos Lucena
Pontifical Catholic University of Rio de Janeiro, Computer Science Department
Rio de Janeiro, RJ - 22453-900, Brazil
E-mail: lucena@inf.puc-rio.br

Paolo Giorgini
University of Trento, Department of Information and Communication Technology
Via Sommarive, 14, 38050 Povo (Trento), Italy
E-mail: paolo.giorgini@dit.unitn.it

Tom Holvoet
K.U. Leuven, Department of Computer Science
Celestijnenlaan 200A, 3001 Leuven, Belgium
E-mail: Tom.Holvoet@cs.kuleuven.be

Alexander Romanovsky
University of Newcastle, School of Computing Science
Newcastle upon Tyne, NE1 7RU, UK
E-mail: alexander.romanovsky@ncl.ac.uk

Library of Congress Control Number: 2006923375

CR Subject Classification (1998): D.2, I.2.11, C.2.4, D.1.3, H.3.5

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-540-33580-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-33580-1 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11738817 06/3142 5 4 3 2 1 0

Foreword

With the integration of computing and communication into the very fabric of our social, economic, and personal existence, the manner in which we think about and build software has become the subject of intense intellectual, scientific, and engineering reexamination. New computing paradigms have been proposed and new software architectures are being examined. The study of multi-agent systems (MAS) is one important movement energized by a growing awareness that application development may need to follow radically new paths. Fundamentally, MAS denotes a new software specification and design paradigm. Moreover, when viewed in the context of large-scale deployment, it emerges as the embodiment of the quintessential concerns facing the software engineering community today. As computing and communication permeates the essential aspects of the societal infrastructure, software must become more nimble, slimmer, more natural, and more discrete. Software must integrate itself in an organic way into the activities it serves and the resources it exploits.

Technological changes and their societal implications clearly impacted the evolution of MAS research. The starting point was the wired network that facilitated the development of distributed applications for which MAS appeared to offer a novel design strategy. The introduction of base stations and wireless communication (with devices that move along the fringe of the wired network and can disconnect unexpectedly for extended periods of time) highlighted the importance of disconnected interactions, highly decoupled computing, and migration across the wireless link. MAS proved to be particularly well suited to respond to the requirements of the new environment; for example, it seems natural for a disconnected host to rely on an agent migrated to the wired network to carry out work on its behalf. The advance of ad hoc networks offered still new opportunities and also new challenges for MAS research. Structuring applications as communities of agents that can float over a physically mobile infrastructure is clearly an intellectually exiting possibility to consider, but finding applications that match well to this environment remains a contentious point among researches and practitioners. Finally, MAS made its presence felt even in the newly emerging field of sensor networks. The flexibility of the basic MAS paradigm is indeed remarkable and all indicators point to its continuing evolution towards enabling application developers to achieve increasingly more effective utilization of the deployed computing and communications infrastructures.

The most visible MAS contributions can be attributed to successful abstraction and conceptualization efforts that demonstrate the expressive power of the various embodiments of the basic paradigm across multiple technological substrates. However, only the combination of exciting conceptual frameworks with analytical power, design methodology, and engineering practice will lead to achieving a truly high impact on the society at large. These concerns should be formative elements for the MAS research agenda. Interestingly enough, the content of this volume matches well with this perspective on the field. Papers on context awareness, coordination, and modeling continue to focus on strengthening and deepening our understanding of conceptual frameworks having both scientific and practical significance. Papers addressing requirements, architecture, and dependability capture methodological and engineering concerns. Analysis alone receives somewhat more limited coverage in this volume.

MAS research is well positioned in terms of technical coverage of the field and promises to lead to the deployment of nimble and responsive user-centered applications. However, the community needs to recognize that in today's environment the marketplace has a greater than ever voice in determining research relevance and impact. This is not due to the evolving patterns of research funding, but due to the high degree of integration of computing and communication into the workings of a modern society. Acknowledging this should lead to the emergence of a research paradigm that focuses on the creative integration of conceptual, engineering, and application concerns in the shaping of the next generation of MAS. I view this volume as an important step along this path.

December 2005

Gruia-Catalin Roman
Washington University in Saint Louis

Preface

Software is becoming present in every aspect of our lives, pushing us inevitably towards a world of autonomous distributed systems. The agent-oriented paradigm holds great promise for responding to the new realities of large-scale distributed systems. It is strongly rooted in the theories underlying multi-agent systems (MASs) and, as a result, offers appropriate software engineering abstractions and mechanisms to address issues such as context-awareness, openness, coordination, ubiquity, mobility, adaptation, and cooperation among heterogeneous and autonomous parties. Not surprisingly, multi-agent software development is one of the most rapidly growing areas of research in academia and software industry.

Nevertheless, the complexity associated with software agents and MASs is considerable, posing new challenges to software engineering community. Without adequate development techniques and methods, MASs will not be sufficiently dependable, trustworthy and extensible, thus making their wide adoption by the industry more difficult. Commercial success of MAS applications requires scalable solutions based on well-understood software engineering approaches that ensure effective deployment and enable reuse.

A large MAS is complex in many ways. When a set of agents interact over open heterogeneous environments, several problems emerge. For example, openess requires appropriate abstractions and mechanisms for supporting context-awareness. In addition, it makes their coordination and management challenging and increases the probability of exceptional situations, security holes and unexpected global effects. Moreover, as users and software engineers delegate more autonomy to their MASs and put more trust in their results, new concerns arise in real-life applications. Yet many of the existing agent-oriented solutions are far from ideal; in practice, systems are often built in an ad-hoc manner, are error-prone, not scalable, not dynamic, and not generally applicable to large-scale environments. If agent-based applications are to be successful, software engineering approaches will be needed to enable scalable deployment.

The main motivation for producing this book is the Software Engineering for Large-Scale Multi-Agent Systems (SELMAS) workshop series, which focuses on bringing together software engineering practitioners and researchers to discuss the several issues arising when MASs are used to engineer complex systems. SELMAS 2005[1] was the fourth edition of the workshop, organized in association with the 27th International Conference on Software Engineering (ICSE), held in Saint Louis, USA, in May 2005. The theme of this particular workshop edition was "Software Everywhere—Context-Aware Agents." To produce the book based on this workshop edition, we decided to extend the workshop coverage, and to

[1] Garcia, A. el al.: Software Engineering for Large-Scale Multi-Agent Systems - SELMAS 2005 (Workshop Report). ACM Software Engineering Notes, Vol. 30, N. 4, July 2005.

invite several of the workshop participants to write chapters for books based on their original position papers, as well as several other leading researchers in the area to prepare additional chapters. Thus, this book is a continuation of a series of three previous ones[2–4].

This book brings together a collection of 15 papers addressing a wide range of issues in software engineering of MASs, reflecting the importance of agent properties in today's software systems. The papers in this book describe recent developments in specific issues and practical experience. At the end of each chapter, the reader will find a list of relevant references for further reading. The papers in this book are grouped into five sections: Context-Awareness, Coordination, Dependability, Modelling, and Requirements and Software Architecture. The first section is especially dedicated to the SELMAS 2005 theme. The other sections contain high-quality contributions on other important complementary concerns in MAS development. We believe that this carefully prepared volume, describing the most recent developments in the field of software engineering for MASs, will be of particular value to all readers interested in these key topics.

With a comprehensive selection of case studies and software engineering solutions for MASs applications, this book provides a valuable resource for a wide audience of readers. The main target readers for this book are researchers and practitioners who want to keep up with the progress of software engineering of MASs, individuals keen to understand the interplay between agents and other software development paradigms, and those interested in experimental results from MAS applications. Software engineers involved in particular aspects of MASs as a part of their work may find it interesting to learn about application of software engineering approaches in building real systems. Some chapters in the book discuss the transitions involving different MAS development phases, such as requirements, architecture specifications, and implementation. One key contribution of this volume is the description of the latest approaches to reasoning about complex MASs. We are confident that this book will be of considerable use to the software engineering community by providing many original and distinct views on such an important interdisciplinary topic, and by contributing to a better understanding and cross-fertilization among individuals in this research area. Our thanks go to all our authors, whose work made this book possible. Many of them also helped during the reviewing process. We would also like to express our gratitude to the members of the Evaluation Committee who were generous with their time and effort when reviewing the submitted papers. We specially thank all people involved—authors, workshop participants, and

[2] Garcia, A., Lucena, C., Castro, J., Zambonelli, F., Omicini, A. (eds.): Software Engineering for Large-Scale Multi-Agent Systems. LNCS, vol. 2603, Springer-Verlag, April 2003.

[3] Lucena, C., Garcia, A., Romanovsky, A., Castro, J., Alencar, P. (eds.): Software Engineering for Multi-Agent Systems II. LNCS, vol. 2940, Springer-Verlag, February 2004.

[4] Choren, R., Garcia, A., Lucena, C., Romanovsky, A. (eds.): Software Engineering for Multi-Agent Systems III. LNCS, vol. 3390, Springer-Verlag, February 2005.

reviewers—for making this book series and the SELMAS workshop editions a high-quality scientific joint project.

December 2005

Alessandro Garcia (Lancaster University, UK)
Ricardo Choren (IME-Rio, Brazil)
Carlos Lucena (PUC-Rio, Brazil)
Paolo Giorgini (University of Trento, Italy)
Tom Holvoet (K. U. Leuven, Belgium)
Alexander Romanovsky (University of Newcastle, UK)

Organization

Evaluation and Program Committee

N. Alechina (University of Nottingham, UK)
E. Alonso (City University London, UK)
R. Ashri (University of Southampton, UK)
B. Bauer (Universität Augsburg, Germany)
C. Bernon (IRIT, France)
M. Brian Blake (Georgetown University, USA)
P. Bresciani (IRST, Italy)
J-P. Briot (Université Paris VI, France)
G. Cabri (Università di Modena e Reggio Emilia, Italy)
J. Castro (UFPE, Brazil)
M. Cossentino (ICAR-CNR, Italy)
S. Cost (University of Maryland Baltimore County, USA)
J. Debenham (University of Technology - Sydney, Australia)
J. Dix (University of Manchester, UK)
C. A. Fernández (Universidad Politécnica de Madrid, Spain)
M. Fredriksson (Blekinge Institute of Tech., Sweden)
C. Ghidini (ITC-irst, Italy)
M-P. Gleizes (IRIT, France)
Z. Guessoum (Université Paris VI, France)
B. Henderson-Sellers (U. of Technology - Sydney, Australia)
K. Henricksen (University of Queensland, Australia)
J. Indulska (University of Queensland, Australia)
C. Jonker (Vrije Universiteit Amsterdam, The Netherlands)
M. Kolp (Université Catholique de Louvain, Belgium)
J. C. Maldonado (USP - So Carlos, Brazil)
V. Mascardi (Università di Genova, Italy)
H. Mouratidis (University of East London, UK)
J. Odell (Agentis Software, USA)
E. Oliveira (Universidade do Porto, Portugal)
A. Omicini (Università di Bologna, Italy)
S. Ossowski (Universidad Rey Juan Carlos, Spain)
L. Penserini (ITC-irst, Italy)
G. Rossi (Universidad Nacional de La Plata, Argentina)
A. G. Serrano (Universidad Politécnica de Madrid, Spain)
J. M. Serrano (Universidad Rey Juan Carlos, Spain)
O. Shehory (IBM Research Center in Haifa, Israel)
J. Shepherdson (British Telecommunications plc., UK)
A. von Staa (PUC-Rio, Brazil)
M. Stal (Siemens AG, Germany)

W. Truszkowski (NASA, USA)
M. Weiss (Carleton University, Canada)
D. Weyns (Katholieke Universiteit Leuven, Belgium)
M. Winikoff (RMIT University, Australia)
F. Zambonelli (Università di Modena e Reggio Emilia, Italy)
A. Zisman (City University London, UK)

Lecture Notes in Computer Science 3914

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Lecture Notes in Computer Science

For information about Vols. 1–3839

please contact your bookseller or Springer

Vol. 3946: T.R. Roth-Berghofer, S. Schulz, D.B. Leake (Eds.), Modeling and Retrieval of Context. XI, 149 pages. 2006. (Sublibrary LNAI).

Vol. 3939: C. Priami, L. Cardelli, S. Emmott (Eds.), Transactions on Computational Systems Biology IV. VII, 141 pages. 2006. (Sublibrary LNBI).

Vol. 3936: M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, A. Yavlinsky (Eds.), Advances in Information Retrieval. XI, 584 pages. 2006.

Vol. 3934: J.A. Clark, R.F. Paige, F.A. C. Polack, P.J. Brooke (Eds.), Security in Pervasive Computing. X, 243 pages. 2006.

Vol. 3933: F. Bonchi, J.-F. Boulicaut (Eds.), Knowledge Discovery in Inductive Databases. VIII, 251 pages. 2006.

Vol. 3931: B. Apolloni, M. Marinaro, G. Nicosia, R. Tagliaferri (Eds.), Neural Nets. XIII, 370 pages. 2006.

Vol. 3930: D.S. Yeung, Z.-Q. Liu, X.-Z. Wang, H. Yan (Eds.), Advances in Machine Learning and Cybernetics. XXI, 1110 pages. 2006. (Sublibrary LNAI).

Vol. 3928: J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.), Smart Card Research and Advanced Applications. XI, 359 pages. 2006.

Vol. 3927: J. Hespanha, A. Tiwari (Eds.), Hybrid Systems: Computation and Control. XII, 584 pages. 2006.

Vol. 3925: A. Valmari (Ed.), Model Checking Software. X, 307 pages. 2006.

Vol. 3924: P. Sestoft (Ed.), Programming Languages and Systems. XII, 343 pages. 2006.

Vol. 3923: A. Mycroft, A. Zeller (Eds.), Compiler Construction. XIII, 277 pages. 2006.

Vol. 3922: L. Baresi, R. Heckel (Eds.), Fundamental Approaches to Software Engineering. XIII, 427 pages. 2006.

Vol. 3921: L. Aceto, A. Ingólfsdóttir (Eds.), Foundations of Software Science and Computation Structures. XV, 447 pages. 2006.

Vol. 3920: H. Hermanns, J. Palsberg (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. XIV, 506 pages. 2006.

Vol. 3918: W.K. Ng, M. Kitsuregawa, J. Li, K. Chang (Eds.), Advances in Knowledge Discovery and Data Mining. XXIV, 879 pages. 2006. (Sublibrary LNAI).

Vol. 3917: H. Chen, F.Y. Wang, C.C. Yang, D. Zeng, M. Chau, K. Chang (Eds.), Intelligence and Security Informatics. XII, 186 pages. 2006.

Vol. 3916: J. Li, Q. Yang, A.-H. Tan (Eds.), Data Mining for Biomedical Applications. VIII, 155 pages. 2006. (Sublibrary LNBI).

Vol. 3915: R. Nayak, M.J. Zaki (Eds.), Knowledge Discovery from XML Documents. VIII, 105 pages. 2006.

Vol. 3914: A. Garcia, R. Choren, C. Lucena, P. Giorgini, T. Holvoet, A. Romanovsky (Eds.), Software Engineering for Multi-Agent Systems IV. XIV, 255 pages. 2006.

Vol. 3910: S.A. Brueckner, G.D.M. Serugendo, D. Hales, F. Zambonelli (Eds.), Engineering Self-Organising Systems. XII, 245 pages. 2006. (Sublibrary LNAI).

Vol. 3909: A. Apostolico, C. Guerra, S. Istrail, P. Pevzner, M. Waterman (Eds.), Research in Computational Molecular Biology. XVII, 612 pages. 2006. (Sublibrary LNBI).

Vol. 3907: F. Rothlauf, J. Branke, S. Cagnoni, E. Costa, C. Cotta, R. Drechsler, E. Lutton, P. Machado, J.H. Moore, J. Romero, G.D. Smith, G. Squillero, H. Takagi (Eds.), Applications of Evolutionary Computing. XXIV, 813 pages. 2006.

Vol. 3906: J. Gottlieb, G.R. Raidl (Eds.), Evolutionary Computation in Combinatorial Optimization. XI, 293 pages. 2006.

Vol. 3905: P. Collet, M. Tomassini, M. Ebner, S. Gustafson, A. Ekárt (Eds.), Genetic Programming. XI, 361 pages. 2006.

Vol. 3904: M. Baldoni, U. Endriss, A. Omicini, P. Torroni (Eds.), Declarative Agent Languages and Technologies III. XII, 245 pages. 2006. (Sublibrary LNAI).

Vol. 3903: K. Chen, R. Deng, X. Lai, J. Zhou (Eds.), Information Security Practice and Experience. XIV, 392 pages. 2006.

Vol. 3901: P.M. Hill (Ed.), Logic Based Program Synthesis and Transformation. X, 179 pages. 2006.

Vol. 3899: S. Frintrop, VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search. XIV, 216 pages. 2006. (Sublibrary LNAI).

Vol. 3898: K. Tuyls, P.J. 't Hoen, K. Verbeeck, S. Sen (Eds.), Learning and Adaption in Multi-Agent Systems. X, 217 pages. 2006. (Sublibrary LNAI).

Vol. 3897: B. Preneel, S. Tavares (Eds.), Selected Areas in Cryptography. XI, 371 pages. 2006.

Vol. 3896: Y. Ioannidis, M.H. Scholl, J.W. Schmidt, F. Matthes, M. Hatzopoulos, K. Boehm, A. Kemper, T. Grust, C. Boehm (Eds.), Advances in Database Technology - EDBT 2006. XIV, 1208 pages. 2006.

Vol. 3895: O. Goldreich, A.L. Rosenberg, A.L. Selman (Eds.), Theoretical Computer Science. XII, 399 pages. 2006.

Vol. 3894: W. Grass, B. Sick, K. Waldschmidt (Eds.), Architecture of Computing Systems - ARCS 2006. XII, 496 pages. 2006.

Vol. 3891: J.S. Sichman, L. Antunes (Eds.), Multi-Agent-Based Simulation VI. X, 191 pages. 2006. (Sublibrary LNAI).

Vol. 3890: S.G. Thompson, R. Ghanea-Hercock (Eds.), Defence Applications of Multi-Agent Systems. XII, 141 pages. 2006. (Sublibrary LNAI).

Vol. 3889: J. Rosca, D. Erdogmus, J.C. Príncipe, S. Haykin (Eds.), Independent Component Analysis and Blind Signal Separation. XXI, 980 pages. 2006.

Vol. 3888: D. Draheim, G. Weber (Eds.), Trends in Enterprise Application Architecture. IX, 145 pages. 2006.

Vol. 3887: J.R. Correa, A. Hevia, M. Kiwi (Eds.), LATIN 2006: Theoretical Informatics. XVI, 814 pages. 2006.

Vol. 3886: E.G. Bremer, J. Hakenberg, E.-H.(S.) Han, D. Berrar, W. Dubitzky (Eds.), Knowledge Discovery in Life Science Literature. XIV, 147 pages. 2006. (Sublibrary LNBI).

Vol. 3885: V. Torra, Y. Narukawa, A. Valls, J. Domingo-Ferrer (Eds.), Modeling Decisions for Artificial Intelligence. XII, 374 pages. 2006. (Sublibrary LNAI).

Vol. 3884: B. Durand, W. Thomas (Eds.), STACS 2006. XIV, 714 pages. 2006.

Vol. 3882: M.L. Lee, K.L. Tan, V. Wuwongse (Eds.), Database Systems for Advanced Applications. XXI, 923 pages. 2006.

Vol. 3881: S. Gibet, N. Courty, J.-F. Kamp (Eds.), Gesture in Human-Computer Interaction and Simulation. XIII, 344 pages. 2006. (Sublibrary LNAI).

Vol. 3880: A. Rashid, M. Aksit (Eds.), Transactions on Aspect-Oriented Software Development I. IX, 335 pages. 2006.

Vol. 3879: T. Erlebach, G. Persinao (Eds.), Approximation and Online Algorithms. X, 349 pages. 2006.

Vol. 3878: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing. XVII, 589 pages. 2006.

Vol. 3877: M. Detyniecki, J.M. Jose, A. Nürnberger, C. J. '. van Rijsbergen (Eds.), Adaptive Multimedia Retrieval: User, Context, and Feedback. XI, 279 pages. 2006.

Vol. 3876: S. Halevi, T. Rabin (Eds.), Theory of Cryptography. XI, 617 pages. 2006.

Vol. 3875: S. Ur, E. Bin, Y. Wolfsthal (Eds.), Hardware and Software, Verification and Testing. X, 265 pages. 2006.

Vol. 3874: R. Missaoui, J. Schmidt (Eds.), Formal Concept Analysis. X, 309 pages. 2006. (Sublibrary LNAI).

Vol. 3873: L. Maicher, J. Park (Eds.), Charting the Topic Maps Research and Applications Landscape. VIII, 281 pages. 2006. (Sublibrary LNAI).

Vol. 3872: H. Bunke, A. L. Spitz (Eds.), Document Analysis Systems VII. XIII, 630 pages. 2006.

Vol. 3870: S. Spaccapietra, P. Atzeni, W.W. Chu, T. Catarci, K.P. Sycara (Eds.), Journal on Data Semantics V. XIII, 237 pages. 2006.

Vol. 3869: S. Renals, S. Bengio (Eds.), Machine Learning for Multimodal Interaction. XIII, 490 pages. 2006.

Vol. 3868: K. Römer, H. Karl, F. Mattern (Eds.), Wireless Sensor Networks. XI, 342 pages. 2006.

Vol. 3866: T. Dimitrakos, F. Martinelli, P.Y.A. Ryan, S. Schneider (Eds.), Formal Aspects in Security and Trust. X, 259 pages. 2006.

Vol. 3865: W. Shen, K.-M. Chao, Z. Lin, J.-P.A. Barthès, A. James (Eds.), Computer Supported Cooperative Work in Design II. XII, 659 pages. 2006.

Vol. 3863: M. Kohlhase (Ed.), Mathematical Knowledge Management. XI, 405 pages. 2006. (Sublibrary LNAI).

Vol. 3862: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), Programming Multi-Agent Systems. XIV, 267 pages. 2006. (Sublibrary LNAI).

Vol. 3861: J. Dix, S.J. Hegner (Eds.), Foundations of Information and Knowledge Systems. X, 331 pages. 2006.

Vol. 3860: D. Pointcheval (Ed.), Topics in Cryptology – CT-RSA 2006. XI, 365 pages. 2006.

Vol. 3858: A. Valdes, D. Zamboni (Eds.), Recent Advances in Intrusion Detection. X, 351 pages. 2006.

Vol. 3857: M.P.C. Fossorier, H. Imai, S. Lin, A. Poli (Eds.), Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. XI, 350 pages. 2006.

Vol. 3855: E. A. Emerson, K.S. Namjoshi (Eds.), Verification, Model Checking, and Abstract Interpretation. XI, 443 pages. 2005.

Vol. 3854: I. Stavrakakis, M. Smirnov (Eds.), Autonomic Communication. XIII, 303 pages. 2006.

Vol. 3853: A.J. Ijspeert, T. Masuzawa, S. Kusumoto (Eds.), Biologically Inspired Approaches to Advanced Information Technology. XIV, 388 pages. 2006.

Vol. 3852: P.J. Narayanan, S.K. Nayar, H.-Y. Shum (Eds.), Computer Vision – ACCV 2006, Part II. XXXI, 977 pages. 2006.

Vol. 3851: P.J. Narayanan, S.K. Nayar, H.-Y. Shum (Eds.), Computer Vision – ACCV 2006, Part I. XXXI, 973 pages. 2006.

Vol. 3850: R. Freund, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 371 pages. 2006.

Vol. 3849: I. Bloch, A. Petrosino, A.G.B. Tettamanzi (Eds.), Fuzzy Logic and Applications. XIV, 438 pages. 2006. (Sublibrary LNAI).

Vol. 3848: J.-F. Boulicaut, L. De Raedt, H. Mannila (Eds.), Constraint-Based Mining and Inductive Databases. X, 401 pages. 2006. (Sublibrary LNAI).

Vol. 3847: K.P. Jantke, A. Lunzer, N. Spyratos, Y. Tanaka (Eds.), Federation over the Web. X, 215 pages. 2006. (Sublibrary LNAI).

Vol. 3846: H. J. van den Herik, Y. Björnsson, N.S. Netanyahu (Eds.), Computers and Games. XIV, 333 pages. 2006.

Vol. 3845: J. Farré, I. Litovsky, S. Schmitz (Eds.), Implementation and Application of Automata. XIII, 360 pages. 2006.

Vol. 3844: J.-M. Bruel (Ed.), Satellite Events at the MoDELS 2005 Conference. XIII, 360 pages. 2006.

Vol. 3843: P. Healy, N.S. Nikolov (Eds.), Graph Drawing. XVII, 536 pages. 2006.

Vol. 3842: H.T. Shen, J. Li, M. Li, J. Ni, W. Wang (Eds.), Advanced Web and Network Technologies, and Applications. XXVII, 1057 pages. 2006.

Vol. 3841: X. Zhou, J. Li, H.T. Shen, M. Kitsuregawa, Y. Zhang (Eds.), Frontiers of WWW Research and Development - APWeb 2006. XXIV, 1223 pages. 2006.

Vol. 3840: M. Li, B. Boehm, L.J. Osterweil (Eds.), Unifying the Software Process Spectrum. XVI, 522 pages. 2006.

¥359.00元

Table of Contents

Context-Awareness and Coordination

Modeling

Requirements and Software Architecture

Dependability

Policy-Driven Configuration and Management of Agent Based Distributed Systems*

Anand R. Tripathi, Devdatta Kulkarni, and Tanvir Ahmed

Department of Computer Science,
University of Minnesota, Minneapolis, MN 55455, USA
{tripathi, dkulk, tahmed}@cs.umn.edu

Abstract. In this paper, we demonstrate a policy-driven approach for building and managing large scale agent based systems. We identify different classes of policies for agent based component integration. We also identify the system services and mechanisms that are required for policy-driven integration of components and their management. Policies are derived from the application level requirements and are used in dynamic configuration of agent based systems. Through case studies of two applications we demonstrate the utility of the policy-driven component integration approach in distributed agent systems.

1 Introduction

Building and managing large distributed component systems is becoming an increasingly challenging task. Continuous intervention by system administrators is generally limited in large-scale distributed environments. System support is needed for reconfiguration and reorganization when systems evolve with the addition of new components.

Component based approaches for structuring distributed systems have sufficiently matured and are widely used today [1]. New challenges in using the component technology are concerned with building systems using dynamic integration of active components such as agents [2]. Policy-driven approaches have been used previously for managing distributed systems [3]. In this paper we propose a policy-driven approach for building *distributed agent systems.* We identify the policy classes and the essential services required in building distributed agent systems.

Policies are derived from an application's functional and non-functional requirements. The functional requirements of an application may require certain components to be present on an agent or certain agents to coordinate towards some common goals. The non-functional requirements of an application may state that all the agents and components should be monitored for failures. In the policy-driven integration approach, the agent composition and the inter-agent interactions are driven by the policies which are derived from the high level requirements. The policies act as a glue in creating a dynamic configuration of

* This work was supported by National Science Foundation grants 0087514 and 0411961.

A. Garcia et al. (Eds.): SELMAS 2005, LNCS 3914, pp. 1–16, 2006.

the system in order to satisfy the functional and non-functional requirements. Since the application requirements are realized through a set of policies hence it is important to monitor events indicating policy violations (*policy-events*) and perform *policy-actions* to ensure that the requirements are not violated. Such an event and an action pair forms a *rule*. These rules are derived from the policies and form the basic mechanism for building policy-driven systems.

An agent is an active object encapsulating other components and is considered as a *first-class component*. It serves as an execution environment with security privileges; representing some principal in the environment. Moreover, an agent may be capable of migrating in the network. In this paper we use the term *component* to refer to objects that are contained in an agent; these objects may be active or passive. This agent model is supported through the Ajanta mobile agent programming framework [4, 5].

In policy-driven distributed agent systems, there are intra-agent policies for component integration, inter-agent policies for agent-to-agent interactions, and policies to ensure system robustness. Based on these policies, rules are derived for dynamic integration of agents and components in the system.

Agent based distributed computing models provide an ideal foundation for policy-driven component integration. Agents are autonomous entities that can encapsulate and enforce local policies. Autonomous agents are capable of learning and adapting to the new or modified global policies that dictate the interactions among distributed agents. Mechanisms of self-configuration, self-monitoring and recovery can be built using agent's capabilities.

In this paper we identify policy classes and present essential services and mechanisms for building distributed agent systems. We use two case study examples to elaborate on this policy classification. One of these systems is targeted towards network monitoring using mobile agents [6] and the other is targeted towards building secure distributed collaborative applications from their high-level specifications [7, 8, 9]. Both these systems are based on the Ajanta mobile agent programming framework.

These two applications lie at the opposite end of the spectrum of building policy-driven distributed agent systems. Network monitoring represents the class of systems that are open and evolving. Such systems do not have any limit on the number of agents present in the domain. There can be spontaneous arrival or departure of agents in the domain. Network monitoring system is targeted towards large domains where hosts, agents, or components may get added or removed over a period of time. Network monitoring policies are related to the configuration, monitoring, and failure-recovery of agents and components in the domain. On the other hand the specification driven secure distributed collaboration framework is used for synthesizing collaborative application by integrating a set of agents and components having specific collaboration and security requirements. Only a fixed set of agents can participate in the collaboration. External agents cannot join the collaboration spontaneously after the collaboration environment has been instantiated. In this way the secure distributed collaboration system represents a closed agent system.