Paolo Bresciani
Paolo Giorgini
Brian Henderson-Sellers
Graham Low
Michael Winikoff (Eds.)

LNAI 3508

Agent-Oriented Information Systems II

6th International Bi-Conference Workshop, AOIS 2004
Riga, Latvia, June 2004 and New York, NY, USA, July 2004
Revised Selected Papers

Springer

Paolo Bresciani Paolo Giorgini
Brian Henderson-Sellers Graham Low
Michael Winikoff (Eds.)

Agent-Oriented Information Systems II

6th International Bi-Conference Workshop, AOIS 2004
Riga, Latvia, June 8, 2004 and New York, NY, USA, July 20, 2004
Revised Selected Papers

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Paolo Bresciani
Institute for Scientific and Technological Research (IRST), Trento, Italy
E-mail: bresciani@itc.it

Paolo Giorgini
University of Trento, Department of Information and Communication Technology
Trento, Italy
E-mail: paolo.giorgini@dit.unitn.it

Brian Henderson-Sellers
University of Technology, Sydney
Faculty of Information Technology, Sydney, Australia
E-mail: brian@it.uts.edu.au

Graham Low
University of New South Wales
School of Information Systems, Technology and Management
Sydney, Australia
E-mail: g.low@unsw.edu.au

Michael Winikoff
RMIT University, School of Computer Science and Information Technology
Melbourne, Australia
E-mail: winikoff@cs.rmit.edu.au

Library of Congress Control Number: 2004106088

CR Subject Classification (1998): I.2.11, H.4, H.3, H.5.2-3, C.2.4, I.2

ISSN 0302-9743
ISBN-10 3-540-25911-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-25911-4 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11426714 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3508

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Lecture Notes in Artificial Intelligence (LNAI)

Vol. 3508: P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, M. Winikoff (Eds.), Agent-Oriented Information Systems II. X, 227 pages. 2005.

Vol. 3501: B. Kégl, G. Lapalme (Eds.), Advances in Artificial Intelligence. XV, 458 pages. 2005.

Vol. 3492: P. Blache, E. Stabler, J. Busquets, R. Moot (Eds.), Logical Aspects of Computational Linguistics. X, 363 pages. 2005.

Vol. 3488: M.-S. Hacid, N.V. Murray, Z.W. Raś, S. Tsumoto (Eds.), Foundations of Intelligent Systems. XIII, 700 pages. 2005.

Vol. 3452: F. Baader, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XI, 562 pages. 2005.

Vol. 3419: B. Faltings, A. Petcu, F. Fages, F. Rossi (Eds.), Constraint Satisfaction and Constraint Logic Programming. X, 217 pages. 2005.

Vol. 3416: M. Böhlen, J. Gamper, W. Polasek, M.A. Wimmer (Eds.), E-Government: Towards Electronic Democracy. XIII, 311 pages. 2005.

Vol. 3415: P. Davidsson, B. Logan, K. Takadama (Eds.), Multi-Agent and Multi-Agent-Based Simulation. X, 265 pages. 2005.

Vol. 3403: B. Ganter, R. Godin (Eds.), Formal Concept Analysis. XI, 419 pages. 2005.

Vol. 3398: D.-K. Baik (Ed.), Systems Modeling and Simulation: Theory and Applications. XIV, 733 pages. 2005.

Vol. 3397: T.G. Kim (Ed.), Artificial Intelligence and Simulation. XV, 711 pages. 2005.

Vol. 3396: R.M. van Eijk, M.-P. Huget, F. Dignum (Eds.), Agent Communication. X, 261 pages. 2005.

Vol. 3394: D. Kudenko, D. Kazakov, E. Alonso (Eds.), Adaptive Agents and Multi-Agent Systems II. VIII, 313 pages. 2005.

Vol. 3392: D. Seipel, M. Hanus, U. Geske, O. Bartenstein (Eds.), Applications of Declarative Programming and Knowledge Management. X, 309 pages. 2005.

Vol. 3374: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems. X, 279 pages. 2005.

Vol. 3371: M.W. Barley, N. Kasabov (Eds.), Intelligent Agents and Multi-Agent Systems. X, 329 pages. 2005.

Vol. 3369: V.R. Benjamins, P. Casanovas, J. Breuker, A. Gangemi (Eds.), Law and the Semantic Web. XII, 249 pages. 2005.

Vol. 3366: I. Rahwan, P. Moraitis, C. Reed (Eds.), Argumentation in Multi-Agent Systems. XII, 263 pages. 2005.

Vol. 3359: G. Grieser, Y. Tanaka (Eds.), Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets. XIV, 257 pages. 2005.

Vol. 3346: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), Programming Multi-Agent Systems. XIV, 249 pages. 2005.

Vol. 3345: Y. Cai (Ed.), Ambient Intelligence for Scientific Discovery. XII, 311 pages. 2005.

Vol. 3343: C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, T. Barkowsky (Eds.), Spatial Cognition IV. XIII, 519 pages. 2005.

Vol. 3339: G.I. Webb, X. Yu (Eds.), AI 2004: Advances in Artificial Intelligence. XXII, 1272 pages. 2004.

Vol. 3336: D. Karagiannis, U. Reimer (Eds.), Practical Aspects of Knowledge Management. X, 523 pages. 2004.

Vol. 3327: Y. Shi, W. Xu, Z. Chen (Eds.), Data Mining and Knowledge Management. XIII, 263 pages. 2005.

Vol. 3315: C. Lemaître, C.A. Reyes, J.A. González (Eds.), Advances in Artificial Intelligence – IBERAMIA 2004. XX, 987 pages. 2004.

Vol. 3303: J.A. López, E. Benfenati, W. Dubitzky (Eds.), Knowledge Exploration in Life Science Informatics. X, 249 pages. 2004.

Vol. 3276: D. Nardi, M. Riedmiller, C. Sammut, J. Santos-Victor (Eds.), RoboCup 2004: Robot Soccer World Cup VIII. XVIII, 678 pages. 2005.

Vol. 3275: P. Perner (Ed.), Advances in Data Mining. VIII, 173 pages. 2004.

Vol. 3265: R.E. Frederking, K.B. Taylor (Eds.), Machine Translation: From Real Users to Research. XI, 392 pages. 2004.

Vol. 3264: G. Paliouras, Y. Sakakibara (Eds.), Grammatical Inference: Algorithms and Applications. XI, 291 pages. 2004.

Vol. 3259: J. Dix, J. Leite (Eds.), Computational Logic in Multi-Agent Systems. XII, 251 pages. 2004.

Vol. 3257: E. Motta, N.R. Shadbolt, A. Stutt, N. Gibbins (Eds.), Engineering Knowledge in the Age of the Semantic Web. XVII, 517 pages. 2004.

Vol. 3249: B. Buchberger, J.A. Campbell (Eds.), Artificial Intelligence and Symbolic Computation. X, 285 pages. 2004.

Vol. 3248: K.-Y. Su, J. Tsujii, J.-H. Lee, O.Y. Kwong (Eds.), Natural Language Processing – IJCNLP 2004. XVIII, 817 pages. 2005.

Vol. 3245: E. Suzuki, S. Arikawa (Eds.), Discovery Science. XIV, 430 pages. 2004.

Vol. 3244: S. Ben-David, J. Case, A. Maruoka (Eds.), Algorithmic Learning Theory. XIV, 505 pages. 2004.

Vol. 3238: S. Biundo, T. Frühwirth, G. Palm (Eds.), KI 2004: Advances in Artificial Intelligence. XI, 467 pages. 2004.

Vol. 3230: J.L. Vicedo, P. Martínez-Barco, R. Muñoz, M. Saiz Noeda (Eds.), Advances in Natural Language Processing. XII, 488 pages. 2004.

Vol. 3229: J.J. Alferes, J. Leite (Eds.), Logics in Artificial Intelligence. XIV, 744 pages. 2004.

Vol. 3228: M.G. Hinchey, J.L. Rash, W.F. Truszkowski, C.A. Rouff (Eds.), Formal Approaches to Agent-Based Systems. VIII, 290 pages. 2004.

Vol. 3215: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. LVII, 906 pages. 2004.

Vol. 3214: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. LVIII, 1302 pages. 2004.

Vol. 3213: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LVIII, 1280 pages. 2004.

Vol. 3209: B. Berendt, A. Hotho, D. Mladenic, M. van Someren, M. Spiliopoulou, G. Stumme (Eds.), Web Mining: From Web to Semantic Web. IX, 201 pages. 2004.

Vol. 3206: P. Sojka, I. Kopecek, K. Pala (Eds.), Text, Speech and Dialogue. XIII, 667 pages. 2004.

Vol. 3202: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Knowledge Discovery in Databases: PKDD 2004. XIX, 560 pages. 2004.

Vol. 3201: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Machine Learning: ECML 2004. XVIII, 580 pages. 2004.

Vol. 3194: R. Camacho, R. King, A. Srinivasan (Eds.), Inductive Logic Programming. XI, 361 pages. 2004.

Vol. 3192: C. Bussler, D. Fensel (Eds.), Artificial Intelligence: Methodology, Systems, and Applications. XIII, 522 pages. 2004.

Vol. 3191: M. Klusch, S. Ossowski, V. Kashyap, R. Unland (Eds.), Cooperative Information Agents VIII. XI, 303 pages. 2004.

Vol. 3187: G. Lindemann, J. Denzinger, I.J. Timm, R. Unland (Eds.), Multiagent System Technologies. XIII, 341 pages. 2004.

Vol. 3176: O. Bousquet, U. von Luxburg, G. Rätsch (Eds.), Advanced Lectures on Machine Learning. IX, 241 pages. 2004.

Vol. 3171: A.L.C. Bazzan, S. Labidi (Eds.), Advances in Artificial Intelligence – SBIA 2004. XVII, 548 pages. 2004.

Vol. 3159: U. Visser, Intelligent Information Integration for the Semantic Web. XIV, 150 pages. 2004.

Vol. 3157: C. Zhang, H. W. Guesgen, W.K. Yeap (Eds.), PRICAI 2004: Trends in Artificial Intelligence. XX, 1023 pages. 2004.

Vol. 3155: P. Funk, P.A. González Calero (Eds.), Advances in Case-Based Reasoning. XIII, 822 pages. 2004.

Vol. 3139: F. Iida, R. Pfeifer, L. Steels, Y. Kuniyoshi (Eds.), Embodied Artificial Intelligence. IX, 331 pages. 2004.

Vol. 3131: V. Torra, Y. Narukawa (Eds.), Modeling Decisions for Artificial Intelligence. XI, 327 pages. 2004.

Vol. 3127: K.E. Wolff, H.D. Pfeiffer, H.S. Delugach (Eds.), Conceptual Structures at Work. XI, 403 pages. 2004.

Vol. 3123: A. Belz, R. Evans, P. Piwek (Eds.), Natural Language Generation. X, 219 pages. 2004.

Vol. 3120: J. Shawe-Taylor, Y. Singer (Eds.), Learning Theory. X, 648 pages. 2004.

Vol. 3097: D. Basin, M. Rusinowitch (Eds.), Automated Reasoning. XII, 493 pages. 2004.

Vol. 3071: A. Omicini, P. Petta, J. Pitt (Eds.), Engineering Societies in the Agents World. XIII, 409 pages. 2004.

Vol. 3070: L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L.A. Zadeh (Eds.), Artificial Intelligence and Soft Computing - ICAISC 2004. XXV, 1208 pages. 2004.

Vol. 3068: E. André, L. Dybkjær, W. Minker, P. Heisterkamp (Eds.), Affective Dialogue Systems. XII, 324 pages. 2004.

Vol. 3067: M. Dastani, J. Dix, A. El Fallah-Seghrouchni (Eds.), Programming Multi-Agent Systems. X, 221 pages. 2004.

Vol. 3066: S. Tsumoto, R. Słowiński, J. Komorowski, J.W. Grzymała-Busse (Eds.), Rough Sets and Current Trends in Computing. XX, 853 pages. 2004.

Vol. 3065: A. Lomuscio, D. Nute (Eds.), Deontic Logic in Computer Science. X, 275 pages. 2004.

Vol. 3060: A.Y. Tawfik, S.D. Goodwin (Eds.), Advances in Artificial Intelligence. XIII, 582 pages. 2004.

Vol. 3056: H. Dai, R. Srikant, C. Zhang (Eds.), Advances in Knowledge Discovery and Data Mining. XIX, 713 pages. 2004.

Vol. 3055: H. Christiansen, M.-S. Hacid, T. Andreasen, H.L. Larsen (Eds.), Flexible Query Answering Systems. X, 500 pages. 2004.

Vol. 3048: P. Faratin, D.C. Parkes, J.A. Rodríguez-Aguilar, W.E. Walsh (Eds.), Agent-Mediated Electronic Commerce V. XI, 155 pages. 2004.

Vol. 3040: R. Conejo, M. Urretavizcaya, J.-L. Pérez-de-la-Cruz (Eds.), Current Topics in Artificial Intelligence. XIV, 689 pages. 2004.

Vol. 3035: M.A. Wimmer (Ed.), Knowledge Management in Electronic Government. XII, 326 pages. 2004.

Vol. 3034: J. Favela, E. Menasalvas, E. Chávez (Eds.), Advances in Web Intelligence. XIII, 227 pages. 2004.

Vol. 3030: P. Giorgini, B. Henderson-Sellers, M. Winikoff (Eds.), Agent-Oriented Information Systems. XIV, 207 pages. 2004.

Vol. 3029: B. Orchard, C. Yang, M. Ali (Eds.), Innovations in Applied Artificial Intelligence. XXI, 1272 pages. 2004.

Vol. 3025: G.A. Vouros, T. Panayiotopoulos (Eds.), Methods and Applications of Artificial Intelligence. XV, 546 pages. 2004.

Vol. 3020: D. Polani, B. Browning, A. Bonarini, K. Yoshida (Eds.), RoboCup 2003: Robot Soccer World Cup VII. XVI, 767 pages. 2004.

Vol. 3012: K. Kurumatani, S.-H. Chen, A. Ohuchi (Eds.), Multi-Agents for Mass User Support. X, 217 pages. 2004.

Vol. 3010: K.R. Apt, F. Fages, F. Rossi, P. Szeredi, J. Váncza (Eds.), Recent Advances in Constraints. VIII, 285 pages. 2004.

Vol. 2990: J. Leite, A. Omicini, L. Sterling, P. Torroni (Eds.), Declarative Agent Languages and Technologies. XII, 281 pages. 2004.

¥396.48元

Preface

Information systems have become the backbone of all kinds of organizations today. In almost every sector – manufacturing, education, health care, government and businesses large and small – information systems are relied upon for everyday work, communication, information gathering and decision-making. Yet, the inflexibilities in current technologies and methods have also resulted in poor performance, incompatibilities and obstacles to change. As many organizations are reinventing themselves to meet the challenges of global competition and e-commerce, there is increasing pressure to develop and deploy new technologies that are flexible, robust and responsive to rapid and unexpected change.

Agent concepts hold great promise for responding to the new realities of information systems. They offer higher-level abstractions and mechanisms which address issues such as knowledge representation and reasoning, communication, coordination, cooperation among heterogeneous and autonomous parties, perception, commitments, goals, beliefs, intentions, etc., all of which need conceptual modelling. On the one hand, the concrete implementation of these concepts can lead to advanced functionalities, e.g., in inference-based query answering, transaction control, adaptive work flows, brokering and integration of disparate information sources, and automated communication processes. On the other hand, their rich representational capabilities allow for more faithful and flexible treatments of complex organizational processes, leading to more effective requirements analysis and architectural/detailed design.

The Agent Oriented Information Systems (AOIS) workshop series focusses on how agent concepts and techniques will contribute to meeting information systems needs today and tomorrow. To foster greater communication and interaction between the information systems and agents communities, the AOIS workshop is organized as a bi-conference event. It is intended to be a single "logical" event with two "physical" venues. This arrangement encourages greater participation from, and more exchange between, both communities.

AOIS 2004 was the sixth edition of the workshop. The first part was hosted on the 8th of June at CAiSE 2004 – the 16th International Conference on Advanced Information Systems Engineering – in Riga (Latvia). The second part was held on the 20th of July at AAMAS 2004 – the 3rd International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2004) – in New York (USA). The workshop received in total 36 submissions, 23 of which were accepted for presentation. These papers were reviewed by at least 3 members of an international Program Committee composed of 31 researchers. The submissions followed a call for papers on all aspects of agent-oriented information systems and showed the range of results achieved in several areas, such as methodologies, applications, modelling, analysis and simulation.

This volume contains the revised versions of 14 selected papers presented at the workshop and an invited paper by Terry Halpin who gave a keynote speech at the CAiSE event. The papers are grouped into four categories: *information systems*, *analysis and modelling*, *methodologies*, and *applications*.

We believe that this carefully prepared volume will be of particular value to all readers in these key topics, describing the most recent developments in the field of agent-oriented information systems.

We thank the authors, the participants and the reviewers for making AOIS 2004 a high-quality scientific event.

March 2005

Paolo Bresciani
Paolo Giorgini
Brian Henderson-Sellers
Graham Low
Michael Winikoff

Organization

Organizing Committee

Paolo Bresciani (AOIS@CAiSE Co-chair)
Institute for Scientific and Technological Research (IRST)
Trento, Italy
Email: bresciani@itc.it

Paolo Giorgini (AOIS@AAMAS Co-chair)
Department of Information and Communication Technology
University of Trento, Italy
Email: paolo.giorgini@dit.unitn.it

Brian Henderson-Sellers (AOIS@CAiSE Co-chair)
Faculty of Information Technology
University of Technology, Sydney, Australia
Email: brian@it.uts.edu.au

Graham Low (AOIS@CAiSE Co-chair)
School of Information Systems, Technology and Management
University of New South Wales, Sydney, Australia
Email: g.low@unsw.edu.au

Michael Winikoff (AOIS@AAMAS Co-chair)
School of Computer Science and Information Technology
RMIT University, Melbourne, Australia
Email: winikoff@cs.rmit.edu.au

Steering Committee

Yves Lespérance
Department of Computer Science
York University, Canada
Email:lesperan@cs.yorku.ca

Gerd Wagner
Department of Information and Technology,
Eindhoven University of Technology, The Netherlands
Email: G.Wagner@tm.tue.nl

Eric Yu
Faculty of Information Studies,
University of Toronto, Canada
Email: eric.yu@utoronto.ca

Program Committee

B. Blake (USA)
H.-D. Burkhard (Germany)
L. Cernuzzi (Paraguay)
L. Cysneiros (Canada)
F. Dignum (The Netherlands)
P. Donzelli (USA)
B. Espinasse (France)
B.H. Far (Canada)
I.A. Ferguson (USA)
T. Finin (USA)
A. Gal (Israel)
U. Garimella (India)
A.K. Ghose (Australia)
L. Kendall (Australia)
M. Kolp (Belgium)
G. Lakemeyer (Germany)
Y. Lespérance (Canada)
F. Lin (HK)
H. Mouratidis (UK)
J.P. Mueller (Germany)
D.E. O'Leary (USA)
J. Odell (USA)
O.F. Rana (UK)
A. Sturm (Israel)
N. Szirbik (The Netherlands)
G. Wagner (The Netherlands)
C. Woo (Canada)
M. Wooldridge (UK)
B. Yu (USA)
Eric Yu (Canada)
F. Zambonelli (Italy)

Table of Contents

Information Systems

Analysis and Modeling

Methodologies

Applications

An Agent-Based Collaborative Emergent Process Management System

Aizhong Lin, Igor Hawryszkiewycz, and Brian Henderson-Sellers

Faculty of Information Technology,
University of Technology, Sydney,
POBox 123, Broadway,
NSW 2007, Australia
{alin, igorh, brian}@it.uts.edu.au

Abstract. An emergent process is a process whose goal and activities to achieve the goal are unable to be specified in advance but emerge over time as knowledge gained from the activities performed earlier shapes the subsequent goal and activities. Collaborative emergent process management needs functions to support the representation and storage of emergent process instances, process automation and knowledge sharing. Traditional process management systems lack the full functionality of collaborative emergent process management. Our research provides an agent-based collaborative emergent process management system that provides the full functionality needed for managing emergent process instances collaboratively. This paper presents the system including the management model, system architecture, major components, key modules and an application.

1 Introduction

An *emergent process* is a process whose goal and activities to achieve that goal are unable to be specified in advance but emerge over time as knowledge gained from the activities performed earlier shapes the subsequent goal and activities. A research project is an example of an emergent process. Emergent processes have the following distinctive features. Firstly, an emergent process may not have a predefined goal, or its goal may mutate over time. Secondly, an emergent process does not have a predefined solution. The solution of each emergent process instance differs from the solutions of other emergent process instances. Thirdly, an emergent process is a knowledge-driven process. The process knowledge gives direction to the current process. Next, an emergent process will not terminate until a satisfactory conclusion is reached - achieving a business goal is not the termination condition of an emergent process. Finally, an emergent process is the intertwining of two process stages - process definition (also called process modelling) and process enactment (also called process execution). In the process definition stage, process participants determine or change the process goal and activities. In the process enactment stage, process participants perform the activities and harvest the knowledge.

Managing emergent processes requires a management system equipped with the functions to support the representation and storage of emergent process instances, support process enactment, support the intertwining of process definition and

P. Bresciani et al. (Eds.): AOIS 2004, LNAI 3508, pp. 1 – 18, 2005.

enactment, and support process knowledge management and sharing. Traditional process management, which focuses on managing routine processes, is inadequate for managing emergent processes because (1) early systems [1,2] do not provide a mechanism to bridge process definition and process enactment, and (2) recent systems (such as workflow management systems) lack the capability to support the evolution of process goals and activities.

Our research contributes an agent-based collaborative emergent process management system to support emergent processes by interacting with human process participants to define and change goals and activities; perform process activities; and manage and share process knowledge. The system consists of two components: a process workspace manager and a personal process agent framework. The process workspace manager resides on a server to manage process workspaces that are used to represent and store emergent process instances. The personal process agent framework is used to construct personal process agents that run on a client allowing process participants to define or change process instances (including goals and solutions), achieve goals or execute activities, support interactions between process participants, and manage and share process knowledge. An experimental emergent process management system has been implemented as demonstrated here.

2 The Management Model

An emergent process instance can be represented by a meta-model (Figure 1). An *emergent process meta-model* defines a collection of elements (such as process goal, process activity, process constraint) and their relationships (such as activity "achieves" a goal) used in emergent processes.

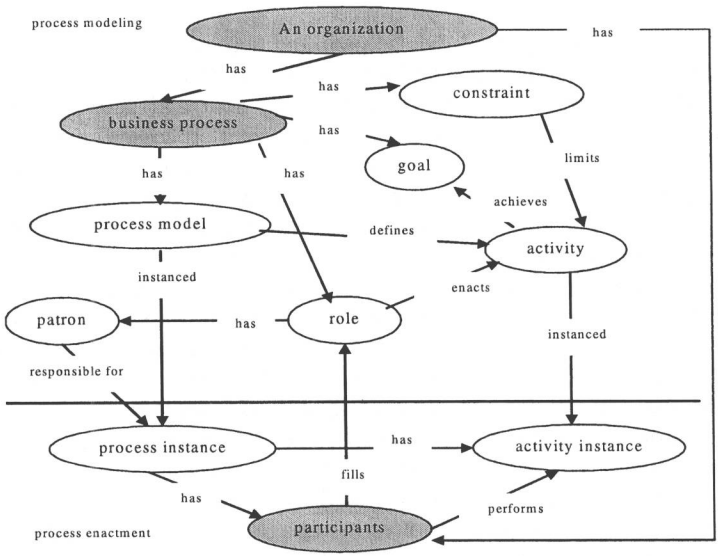

Fig. 1. The meta-model to represent emergent process instances

An emergent process instance (left arrow in Figure 2) can be defined (create, change, or mutate the goals, activities, constraints) by process participants who can also achieve goals or perform activities defined in the instance. Process participants can perceive the changes that take place in the process instance and harvest process knowledge from performing the process instance (right hand arrow in Figure 2). The newly gained process knowledge combined with existing knowledge is used to define process goals and activities (back to the left arrow).

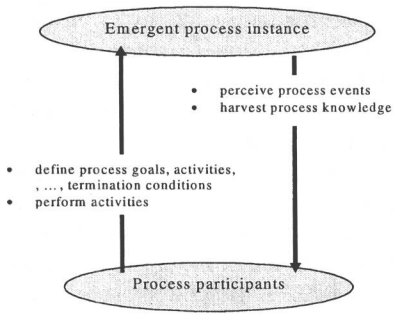

Fig. 2. The management model

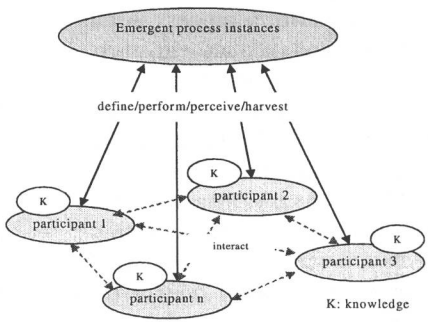

Fig. 3. The detailed management model

During the enactment of an emergent process instance, distributed process participants interact with each other. Figure 3 expands the system architecture shown in Figure 2 for managing emergent business processes. From these two figures, process functions can be summarized as:

- *define*: create, modify or mutate process elements
- *perform*: execute activities to achieve process goals
- *perceive*: perceive changes in process instances
- *harvest*: harvest process knowledge from emergent process instances
- *manage*: manage process knowledge and the path of changes for process instances
- *interact*: support interactions between process participants

3 The System Architecture

The emergent process management model determines the emergent process management system architecture. To convert the management model to the system architecture, two decisions have to be made. Firstly, a process workspace component is developed to represent and store instances of emergent processes. The process workspace component consists of

- *a process workspace (PW) model* is a set of process elements, a set of relationships between process workspaces and a set of links each of which connects a process workspace to another process workspace.

- *a directed process workspace graph (DPWG)* contains a set of related process workspace to represent a process instance.
- a process workspace library (PWL) contains a set of directed process workspace graphs in an organization.
- a process workspace manager (PWM) is a software component that manages a PWL. The management functions provided by PWM include the creation, retrieval, modification, deletion and access control of a single PW in the PWL.

Secondly, software agents are developed to assist human process participants to define and perform process instances, to perceive process events, to harvest process knowledge and interact with each other. Each agent, called the personal process agent (PPA), works on behalf of one and only one human process participant. It is situated in a specific process environment (such as an organization) and is capable of autonomous and flexible actions to respond to changes in the process environment.

The management system has two components: a process workspace manager (PWM) and a set of personal process agents (PPAs) (Figure 4). The PWM maintains DPWGs, each of which contains a set of related PWs, used to represent and store emergent process instances (top of Figure 4). PPAs (middle of Figure 4) are employed to assist the work of human participants. Each PPA is generated from a generic personal process framework. An agent has process functions to define process elements and relationships for process instances in PWs, to perceive the events that have taken place in PWs, to achieve goals or execute activities defined in PWs, to harvest process knowledge from activities performed and to interact with other PPAs.

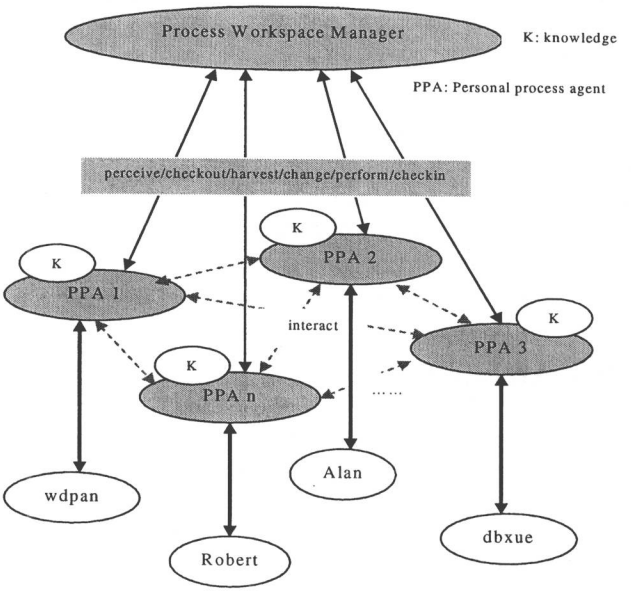

Fig. 4. The architecture of an emergent process management system

The system architecture is a hybrid, combining three traditional architectures: client/server, peer-to-peer and producer/consumer. Firstly, the client/server architecture is necessary in this system architecture because:

- An emergent process instance should be accessed by all participants who have permission to work on it.
- The process instance should appear the same to all participants at all times.

Secondly, the producer/consumer architecture is necessary to:

- Produce: any human participant may define or change (produce) process goals or activities and advertise process knowledge in the process instance.
- Consume: any human participant may perform process activities or achieve process goals.

However, this producer/consumer architecture extends traditional producer/consumer architecture in two respects:

- In a traditional producer/consumer architecture, producers and consumers play different roles. A producer is not a consumer and a consumer does not have to be a producer. In our producer/consumer architecture, however, a PPA can be both producer and consumer, i.e., a PPA may produce goals or activities for others to consume, and consume goals or activities produced by others, even by itself.
- In a traditional producer/consumer architecture, producers do not have to wait for the results of consumers. In this architecture, however, producers may expect the results from consumers to produce new goals or activities.

Finally, the peer-to-peer architecture is needed because:

- Participants in a group are naturally distributed.
- The relationships between human participants are peer to peer because no participant can control other participants. No participant can directly access knowledge owned by other participants.

4 The Implementation of the Emergent Process Management System (EPMS)

Based on the system architecture, the management system is implemented with two components: the process workspace manager (PWM) and the PPA framework.

4.1 Support Techniques

The emergent process management system is implemented as shown in Figures 5 and 6. Figure 5 illustrates the technical support of the PWL (Process Workspace Library) and the PWM. The PWL is built on two technologies: Folders/Files management technology and PWXML (Process Workspace XML – designed by the first author as part of his PhD). Folders/Files management is used to manage folders and files in which each folder is a DPWG (Directed Process Workspace Graph) and each file is a PW. Based on standard XML [3], PWXML is used to represent the PW, each PW