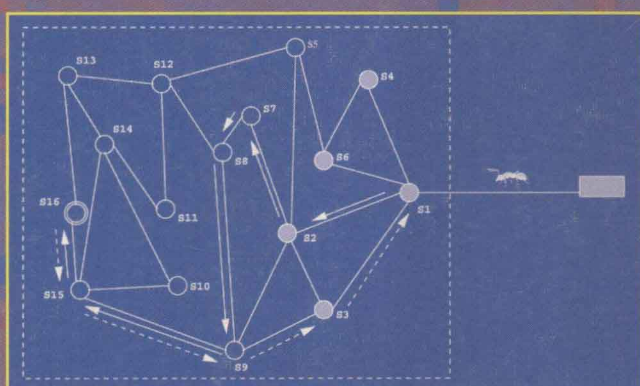
State-of-the-Art Survey

LNAI 2977

Giovanna Di Marzo Serugendo
Anthony Karageorgos
Omer F. Rana
Franco Zambonelli (Eds.)

Engineering Self-Organising Systems

Nature-Inspired Approaches to Software Engineering



Springer

Giovanna Di Marzo Serugendo
Anthony Karageorgos Omer F. Rana
Franco Zambonelli (Eds.)

Engineering Self-Organising Systems

Nature-Inspired Approaches to Software Engineering

Springer

Volume Editors

Giovanna Di Marzo Serugendo
University of Geneva
Centre Universitaire d'Informatique, 1211 Geneva 4, Switzerland
E-mail: Giovanna.Dimarzo@cui.unige.ch

Anthony Karageorgos
University of Thessaly
Department of Computer and Communication Engineering
37 Glavani, 28th October Str., Deligiorgi Building
4th floor, room D3/4, 382 21 Volos, Greece
E-mail: karageorgos@computer.org

Omer F. Rana
Cardiff University
School of Computer Science
5 The Parade, P.O.Box 916, Cardiff CF24 3XF, UK
E-mail: o.f.rana@cs.cardiff.ac.uk

Franco Zambonelli
Università di Modena e Reggio Emilia
Dipartimento di Scienze e Metodi dell'Ingegneria
42100 Reggio Emilia, Italy
E-mail: franco.zambonelli@unimore.it

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): I.2.11, I.2, C.2.4, F.1.1, D.2

ISSN 0302-9743
ISBN 3-540-21201-9 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by DA-TeX Gerd Blumenstein
Printed on acid-free paper SPIN: 10989791 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 2977

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Preface

The spread of the Internet, mobile communications and the evolution of traditional market models, such as the proliferation of e-commerce, has led to much existing IT infrastructure operating as a global dynamic system. It is therefore important that software that operates over this infrastructure adapt accordingly, to provide highly customised services to a huge user population. New maintenance requirements have to be met, for example software that cannot be stopped must still evolve. Furthermore, software must cope with changes in interconnectivity requirements, such as network topology workload fluctuations, mobility of devices, node failures, and interoperation of unreliable, possibly erroneous distributed code.

As information handling systems get more complex, it becomes more difficult to manage them using traditional approaches based on centralised and predefined control mechanisms. Over recent years there has been a significant increase in interest in taking inspiration from biology, the physical world, chemistry, or social systems to more effectively manage such systems – generally based on the concept of Self-organisation. This gave rise to Self-organising Applications (SOA) – generally applications that undertake problem solving (in a particular application domain) based on distributed interactions between uncoupled system components. Typical examples of such distributed interactions in SOA include systems that reproduce behaviours of social insects or multicellular organisms. Another example is multiagent systems that aim to reproduce behaviours found in human societies – thereby providing a useful paradigm for managing large, distributed information handling systems. In multiagent systems self-organising behaviour results from the autonomous behaviour of system components and the locality of their interactions. The key characteristic of all these applications is their ability to achieve complex collective tasks with relatively simple individual behaviours.

There is currently a consensus that modern applications can gain in robustness, management, and simplicity if they are developed according to the principles of self-organisation (which one finds in nature). The central question related to the engineering of SOA is how to specify the local behaviours of individual components to achieve the desired (self-organising) global system behaviour. This involves, in particular, defining a global goal, designing local behaviours and interactions that will result in the desired global behaviour. Subsequently, it is necessary to monitor system execution to verify the emergence of the correct system behaviour, and the achievement of the global goal. This is a difficult exercise because the global system behaviour is not directly predictable as the "sum" of the local behaviours. It is also important to note that self-organisation is not useful in all instances – as it may lead to "parasitic" and unwanted behaviours. The need to monitor the system as it evolves from some initial state is therefore an important requirement in many applications that make use of self-organising principles.

Traditional software engineering methodologies also make it possible to define a global behaviour only as a closed function of the behaviour of the various parts of the system. They involve techniques based on assumptions, such as interfaces fixed at design time and statically established ontologies, which therefore makes them insufficient for engineering self-organisation. Furthermore, traditional practices in multiagent systems have introduced basic techniques for autonomously interacting or retrieving information, such as agent coordination, service description, and ontology-based interaction. However, these techniques rely on preprogrammed interaction patterns, preventing large-scale adaptation to unexpected environmental changes.

Current engineering practices that directly address self-organisation primarily involve designing distributed algorithms according to natural life metaphors, such as the social insect metaphor and the immune system. More recently, appropriate interaction mechanisms and middleware technologies specifically targeting the development of self-organising applications have been introduced. However, many issues such as verification and systematic methodologies covering the whole engineering life cycle still remain open.

This book intends to serve as a reference and starting point for establishing the field of *Engineering Self-Organising Applications*. It comprises papers presented at the Engineering Self-Organising Applications (ESOA 2003) workshop (held at the Autonomous Agents and Multi-Agents Systems conference (AAMAS 2003) in Melbourne in July 2003), and selected invited papers from leading contributors in the self-organisation field.

The book presents the current state-of-the-art perspective on the field. Part I contains several papers related to self-organising applications: Di Marzo Serugendo et al. review natural self-organising systems and their interaction mechanisms and discuss several examples of artificial self-organising applications; Brueckner et al. present SWARMing principles applied to mobile ad hoc networks; Thompson then discusses the use of self-organising systems for agent-based implementations.

In Part II, translations from a diverse collection of natural-life metaphors are presented. Naghpal presents self-organising principles from multicellular organisms and their application to multiagent systems. Tateson et al. apply self-organising behaviour of cells in a developing fruit fly to wireless communication networks. Guo et al. present a genetic algorithm for designing self-assembling objects. This is followed by two papers inspired from ant-based behaviour properties such as indirect communication of interacting entities via the environment (stigmetry): Handl et al. apply ant mechanisms in data set clustering, and Hadeli et al. in manufacturing control. Foukia et al. combine ant-based behaviour and the immune system metaphor to build an intrusion detection and response system in a computer network. Ulieru et al. design logistics infrastructures based on the holonic structure of the Universe. Subsequently, Babanov et al. take their

Programme Committee

Karl Aberer	EPFL, Switzerland
Yaneer Bar-Yam	New England Complex Systems Institute, USA
Rene te Boekhorst	University of Hertfordshire, UK
Sven Brueckner	Altarum, USA
Marco Dorigo	IRIDIA, Université Libre de Bruxelles, Belgium
Noria Foukia	University of Geneva, Switzerland
Nigel Gilbert	University of Surrey, UK
Maria Gini	University of Minnesota, USA
Salima Hassas	University of Lyon, France
Jeff Kephart	IBM Thomas J. Watson Research Center, USA
Mark Klein	MIT Sloan School of Management, USA
Ghita Kouadri Mostefaoui	University of Fribourg, Switzerland
Soraya Kouadri Mostefaoui	University of Fribourg, Switzerland
Marco Mamei	Università di Modena e Reggio Emilia, Italy
Philippe Massonet	CETIC, Belgium
Evangelos Milios	Dalhousie University, Canada
Radhika Nagpal	MIT Artificial Intelligence Lab, MIT, USA
N.C. Narendra	Hewlett-Packard, India
Andrea Omicini	University of Bologna, Italy
H. Van Dyke Parunak	Altarum, USA
Daniel Polani	University of Hertfordshire, UK
David Servat	Laboratoire d'Informatique Appliquée, LIP6, Paris, France
Simon Thompson	BT Exact Technologies, UK
Mihaela Ulieru	University of Calgary, Canada
Chris Van Aart	University of Amsterdam, Netherlands
Tom Wagner	Honeywell Laboratories, USA
David Wolpert	NASA Ames Research Center, USA

We are grateful to the Programme Committee of the ESOA 2003 workshop for returning their reviews on time, and for offering useful suggestions on improving the workshop. All papers submitted to the workshop were reviewed by a minimum of two members of the programme committee.

inspiration from evolutionary computation to build an additional layer on top of heterogeneous strategies in electronic markets.

Part III describes artificial self-organisation mechanisms specifically introduced for computer systems applications. Capera et al. propose an adaptive multiagent system theory for providing self-organising behaviour in mechanical design. Hales et al. propose the notion of tags, a marking attached to agents and observable by other agents, as an interaction mechanism for realising self-organisation. Yolum et al. then establish the notion of trust as a self-organising mechanism in referral networks.

In Part IV, self-organisation specific to coordination models is discussed by Menezes et al. in the case of the Linda-based systems. Along this line, Mamei et al. describe the TOTA coordination environment propagating tuples according to a propagation rule.

Part V concludes the book with methods and tools for engineering emergent behaviour. Airiau et al. discuss the COllective INtelligence framework for accessing shared resources among several users. Jelasity et al. describe simple components that can be combined to implement complex functions. Finally, Bernon et al. present the Adelfe methodology that guides a programmer during the design of SOA.

December, 2003

Giovanna Di Marzo Serugendo,
Anthony Karageorgos,
Omer F. Rana,
Franco Zambonelli

Lecture Notes in Artificial Intelligence (LNAI)

Vol. 2980: A. Blackwell, K. Marriott, A. Shimojima (Eds.), Diagrammatic Representation and Inference. XV, 448 pages. 2004.

Vol. 2977: G. Di Marzo Serugendo, A. Karageorgos, O.F. Rana, F. Zambonelli (Eds.), Engineering Self-Organising Systems. X, 299 pages. 2004.

Vol. 2961: P. Eklund (Ed.), Concept Lattices. IX, 411 pages. 2004.

Vol. 2953: K. Konrad, Model Generation for Natural Language Interpretation and Analysis. XIII, 166 pages. 2004.

Vol. 2934: G. Lindemann, D. Moldt, M. Paolucci (Eds.), Regulated Agent-Based Social Systems. X, 301 pages. 2004.

Vol. 2930: F. Winkler (Ed.), Automated Deduction in Geometry. VII, 231 pages. 2004.

Vol. 2926: L. van Elst, V. Dignum, A. Abecker (Eds.), Agent-Mediated Knowledge Management. XI, 428 pages. 2004.

Vol. 2923: V. Lifschitz, I. Niemelä (Eds.), Logic Programming and Nonmonotonic Reasoning. IX, 365 pages. 2004.

Vol. 2915: A. Camurri, G. Volpe (Eds.), Gesture-Based Communication in Human-Computer Interaction. XIII, 558 pages. 2004.

Vol. 2913: T.M. Pinkston, V.K. Prasanna (Eds.), High Performance Computing - HiPC 2003. XX, 512 pages. 2003.

Vol. 2903: T.D. Gedeon, L.C.C. Fung (Eds.), AI 2003: Advances in Artificial Intelligence. XVI, 1075 pages. 2003.

Vol. 2902: F.M. Pires, S.P. Abreu (Eds.), Progress in Artificial Intelligence. XV, 504 pages. 2003.

Vol. 2892: F. Dau, The Logic System of Concept Graphs with Negation. XI, 213 pages. 2003.

Vol. 2891: J. Lee, M. Barley (Eds.), Intelligent Agents and Multi-Agent Systems. X, 215 pages. 2003.

Vol. 2882: D. Veit, Matchmaking in Electronic Markets. XV, 180 pages. 2003.

Vol. 2871: N. Zhong, Z.W. Raś, S. Tsumoto, E. Suzuki (Eds.), Foundations of Intelligent Systems. XV, 697 pages. 2003.

Vol. 2854: J. Hoffmann, Utilizing Problem Structure in Planing. XIII, 251 pages. 2003.

Vol. 2843: G. Grieser, Y. Tanaka, A. Yamamoto (Eds.), Discovery Science. XII, 504 pages. 2003.

Vol. 2842: R. Gavaldá, K.P. Jantke, E. Takimoto (Eds.), Algorithmic Learning Theory. XI, 313 pages. 2003.

Vol. 2838: N. Lavrač, D. Gamberger, L. Todorovski, H. Blockeel (Eds.), Knowledge Discovery in Databases: PKDD 2003. XVI, 508 pages. 2003.

Vol. 2837: N. Lavrač, D. Gamberger, L. Todorovski, H. Blockeel (Eds.), Machine Learning: ECML 2003. XVI, 504 pages. 2003.

Vol. 2835: T. Horváth, A. Yamamoto (Eds.), Inductive Logic Programming. X, 401 pages. 2003.

Vol. 2821: A. Günter, R. Kruse, B. Neumann (Eds.), KI 2003: Advances in Artificial Intelligence. XII, 662 pages. 2003.

Vol. 2807: V. Matoušek, P. Mautner (Eds.), Text, Speech and Dialogue. XIII, 426 pages. 2003.

Vol. 2801: W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, J.T. Kim (Eds.), Advances in Artificial Life. XVI, 905 pages. 2003.

Vol. 2797: O.R. Zaïane, S.J. Simoff, C. Djeraba (Eds.), Mining Multimedia and Complex Data. XII, 281 pages. 2003.

Vol. 2792: T. Rist, R. Aylett, D. Ballin, J. Rickel (Eds.), Intelligent Virtual Agents. XV, 364 pages. 2003.

Vol. 2782: M. Klusch, A. Omicini, S. Ossowski, H. Laamanen (Eds.), Cooperative Information Agents VII. XI, 345 pages. 2003.

Vol. 2780: M. Dojat, E. Keravnou, P. Barahona (Eds.), Artificial Intelligence in Medicine. XIII, 388 pages. 2003.

Vol. 2777: B. Schölkopf, M.K. Warmuth (Eds.), Learning Theory and Kernel Machines. XIV, 746 pages. 2003.

Vol. 2752: G.A. Kaminka, P.U. Lima, R. Rojas (Eds.), RoboCup 2002: Robot Soccer World Cup VI. XVI, 498 pages. 2003.

Vol. 2741: F. Baader (Ed.), Automated Deduction – CADE-19. XII, 503 pages. 2003.

Vol. 2705: S. Renals, G. Grefenstette (Eds.), Text- and Speech-Triggered Information Access. VII, 197 pages. 2003.

Vol. 2703: O.R. Zaïane, J. Srivastava, M. Spiliopoulou, B. Masand (Eds.), WEBKDD 2002 - MiningWeb Data for Discovering Usage Patterns and Profiles. IX, 181 pages. 2003.

Vol. 2700: M.T. Pazienza (Ed.), Extraction in the Web Era. XIII, 163 pages. 2003.

Vol. 2699: M.G. Hinchey, J.L. Rash, W.F. Truszkowski, C.A. Rouff, D.F. Gordon-Spears (Eds.), Formal Approaches to Agent-Based Systems. IX, 297 pages. 2002.

Vol. 2691: V. Mařík, J.P. Müller, M. Pechoucek (Eds.), Multi-Agent Systems and Applications III. XIV, 660 pages. 2003.

Vol. 2684: M.V. Butz, O. Sigaud, P. Gérard (Eds.), Anticipatory Behavior in Adaptive Learning Systems. X, 303 pages. 2003.

Vol. 2671: Y. Xiang, B. Chaib-draa (Eds.), Advances in Artificial Intelligence. XIV, 642 pages. 2003.

Vol. 2663: E. Menasalvas, J. Segovia, P.S. Szczepaniak (Eds.), Advances in Web Intelligence. XII, 350 pages. 2003.

Vol. 2661: P.L. Lanzi, W. Stolzmann, S.W. Wilson (Eds.), Learning Classifier Systems. VII, 231 pages. 2003.

Vol. 2654: U. Schmid, Inductive Synthesis of Functional Programs. XXII, 398 pages. 2003.

Vol. 2650: M.-P. Huget (Ed.), Communications in Multiagent Systems. VIII, 323 pages. 2003.

Vol. 2645: M.A. Wimmer (Ed.), Knowledge Management in Electronic Government. XI, 320 pages. 2003.

Vol. 2639: G. Wang, Q. Liu, Y. Yao, A. Skowron (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. XVII, 741 pages. 2003.

Vol. 2637: K.-Y. Whang, J. Jeon, K. Shim, J. Srivastava, Advances in Knowledge Discovery and Data Mining. XVIII, 610 pages. 2003.

Vol. 2636: E. Alonso, D. Kudenko, D. Kazakov (Eds.), Adaptive Agents and Multi-Agent Systems. XIV, 323 pages. 2003.

Vol. 2627: B. O'Sullivan (Ed.), Recent Advances in Constraints. X, 201 pages. 2003.

Vol. 2600: S. Mendelson, A.J. Smola (Eds.), Advanced Lectures on Machine Learning. IX, 259 pages. 2003.

Vol. 2592: R. Kowalczyk, J.P. Müller, H. Tianfield, R. Unland (Eds.), Agent Technologies, Infrastructures, Tools, and Applications for E-Services. XVII, 371 pages. 2003.

Vol. 2586: M. Klusch, S. Bergamaschi, P. Edwards, P. Petta (Eds.), Intelligent Information Agents. VI, 275 pages. 2003.

Vol. 2583: S. Matwin, C. Sammut (Eds.), Inductive Logic Programming. X, 351 pages. 2003.

Vol. 2581: J.S. Sichman, F. Bousquet, P. Davidsson (Eds.), Multi-Agent-Based Simulation. X, 195 pages. 2003.

Vol. 2577: P. Petta, R. Tolksdorf, F. Zambonelli (Eds.), Engineering Societies in the Agents World III. X, 285 pages. 2003.

Vol. 2569: D. Karagiannis, U. Reimer (Eds.), Practical Aspects of Knowledge Management. XIII, 648 pages. 2002.

Vol. 2560: S. Goronzy, Robust Adaptation to Non-Native Accents in Automatic Speech Recognition. XI, 144 pages. 2002.

Vol. 2557: B. McKay, J. Slaney (Eds.), AI 2002: Advances in Artificial Intelligence. XV, 730 pages. 2002.

Vol. 2554: M. Beetz, Plan-Based Control of Robotic Agents. XI, 191 pages. 2002.

Vol. 2543: O. Bartenstein, U. Geske, M. Hannebauer, O. Yoshie (Eds.), Web Knowledge Management and Decision Support. X, 307 pages. 2003.

Vol. 2541: T. Barkowsky, Mental Representation and Processing of Geographic Knowledge. X, 174 pages. 2002.

Vol. 2533: N. Cesa-Bianchi, M. Numao, R. Reischuk (Eds.), Algorithmic Learning Theory. XI, 415 pages. 2002.

Vol. 2531: J. Padget, O. Shehory, D. Parkes, N.M. Sadeh, W.E. Walsh (Eds.), Agent-Mediated Electronic Commerce IV. Designing Mechanisms and Systems. XVII, 341 pages. 2002.

Vol. 2527: F.J. Garijo, J.-C. Riquelme, M. Toro (Eds.), Advances in Artificial Intelligence - IBERAMIA 2002. XVIII, 955 pages. 2002.

Vol. 2522: T. Andreasen, A. Motro, H. Christiansen, H.L. Larsen (Eds.), Flexible Query Answering Systems. X, 383 pages. 2002.

Vol. 2514: M. Baaz, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIII, 465 pages. 2002.

Vol. 2507: G. Bittencourt, G.L. Ramalho (Eds.), Advances in Artificial Intelligence. XIII, 417 pages. 2002.

Vol. 2504: M.T. Escrig, F. Toledo, E. Golobardes (Eds.), Topics in Artificial Intelligence. XI, 427 pages. 2002.

Vol. 2499: S.D. Richardson (Ed.), Machine Translation: From Research to Real Users. XXI, 254 pages. 2002.

Vol. 2484: P. Adriaans, H. Fernau, M. van Zaanen (Eds.), Grammatical Inference: Algorithms and Applications. IX, 315 pages. 2002.

Vol. 2479: M. Jarke, J. Koehler, G. Lakemeyer (Eds.), KI 2002: Advances in Artificial Intelligence. XIII, 327 pages. 2002.

Vol. 2475: J.J. Alpigini, J.F. Peters, A. Skowron, N. Zhong (Eds.), Rough Sets and Current Trends in Computing. XV, 640 pages. 2002.

Vol. 2473: A. Gómez-Pérez, V.R. Benjamins (Eds.), Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web. XI, 402 pages. 2002.

Vol. 2466: M. Beetz, J. Hertzberg, M. Ghallab, M.E. Pollack (Eds.), Advances in Plan-Based Control of Robotic Agents. VIII, 291 pages. 2002.

Vol. 2464: M. O'Neill, R.F.E. Sutcliffe, C. Ryan, M. Eaton, N.J.L. Griffith (Eds.), Artificial Intelligence and Cognitive Science. XI, 247 pages. 2002.

Vol. 2448: P. Sojka, I. Kopecek, K. Pala (Eds.), Text, Speech and Dialogue. XII, 481 pages. 2002.

Vol. 2447: D.J. Hand, N.M. Adams, R.J. Bolton (Eds.), Pattern Detection and Discovery. XII, 227 pages. 2002.

Vol. 2446: M. Klusch, S. Ossowski, O. Shehory (Eds.), Cooperative Information Agents VI. XI, 321 pages. 2002.

Vol. 2445: C. Anagnostopoulou, M. Ferrand, A. Smaill (Eds.), Music and Artificial Intelligence. VIII, 207 pages. 2002.

Vol. 2443: D. Scott (Ed.), Artificial Intelligence: Methodology, Systems, and Applications. X, 279 pages. 2002.

Vol. 2432: R. Bergmann, Experience Management. XXI, 393 pages. 2002.

Vol. 2431: T. Elomaa, H. Mannila, H. Toivonen (Eds.), Principles of Data Mining and Knowledge Discovery. XIV, 514 pages. 2002.

Vol. 2430: T. Elomaa, H. Mannila, H. Toivonen (Eds.), Machine Learning: ECML 2002. XIII, 532 pages. 2002.

Vol. 2427: M. Hannebauer, Autonomous Dynamic Reconfiguration in Multi-Agent Systems. XXI, 284 pages. 2002.

Vol. 2424: S. Flesca, S. Greco, N. Leone, G. Ianni (Eds.), Logics in Artificial Intelligence. XIII, 572 pages. 2002.

Vol. 2417: M. Ishizuka, A. Sattar (Eds.), PRICAI 2002: Trends in Artificial Intelligence. XX, 623 pages. 2002.

Table of Contents

Part I: Applications

Self-Organisation: Paradigms and Applications
Giovanna Di Marzo Serugendo, Noria Foukia, Salima Hassas, Anthony Karageorgos, Soraya Kouadri Mostéfaoui, Omer F. Rana, Mihaela Ulieru, Paul Valckenaers, and Chris Van Aart 1

Self-Organizing MANET Management
Sven A. Brueckner and H. Van Dyke Parunak 20

Toward the Application of Self Organizing Systems to Agent Infrastructures - A Critical Review & Prospects
Simon Thompson 36

Part II: Natural Metaphors

Multi-cells and Genetic Algorithms

A Catalog of Biologically-Inspired Primitives for Engineering Self-Organization
Radhika Nagpal 53

Nature-Inspired Self-Organisation in Wireless Communications Networks
Richard Tateson, Susan Howard, and Ray Bradbeer 63

Designing Self-Assembly for 2-Dimensional Building Blocks
Ying Guo, Geoff Poulton, Phil Valencia, and Geoff James 75

Stigmergy

Strategies for the Increased Robustness of Ant-Based Clustering
Julia Handl, Joshua Knowles, and Marco Dorigo 90

Self-Organising in Multi-agent Coordination and Control Using Stigmergy
Hadeli, Paul Valckenaers, Constantin Zamfirescu, Hendrik Van Brussel, Bart Saint Germain, Tom Hoelvoet, and Elke Steegmans 105

Managing Computer Networks Security through Self-Organization: A Complex System Perspective
Noria Foukia and Salima Hassas 124

Atoms and Evolution

A Holonic Self-Organization Approach to the Design of Emergent e-Logistics Infrastructures
Mihaela Ulieru and Rainer Unland 139

An Evolutionary Approach for Studying Heterogeneous Strategies in Electronic Markets
Alexander Babanov, Wolfgang Ketter, and Maria Gini 157

Part III: Artificial Interaction Mechanisms

Self-Organizing Agents for Mechanical Design
Davy Capera, Marie-Pierre Gleizes, and Pierre Glize 169

Can Tags Build Working Systems? From MABS to ESOA
David Hales and Bruce Edmonds 186

Self-Organizing Referral Networks: A Process View of Trust and Authority
Pınar Yolum and Munindar P. Singh 195

Part IV: Middleware

Adaptiveness in Linda-Based Coordination Models
Ronaldo Menezes and Robert Tolksdorf 212

Self-Organization in Multi Agent Systems: A Middleware Approach
Marco Mamei and Franco Zambonelli 233

Part V: Methods and Tools

Providing Effective Access to Shared Resources: A COIN Approach
Stéphane Airiau, Sandip Sen, David H. Wolpert, and Kagan Tumer 249

A Modular Paradigm for Building Self-Organizing Peer-to-Peer Applications
Márk Jelasity, Alberto Montresor, and Ozalp Babaoglu 265

Tools for Self-Organizing Applications Engineering
Carole Bernon, Valérie Camps, Marie-Pierre Gleizes, and Gauthier Picard 283

Author Index 299

Self-Organisation: Paradigms and Applications

Giovanna Di Marzo Serugendo, Noria Foukia, Salima Hassas, Anthony Karageorgos, Soraya Kouadri Mostéfaoui, Omer F. Rana, Mihaela Ulieru, Paul Valckenaers, and Chris Van Aart

Engineering Self-Organising Applications Working Group
http://www.agentcities.org/Activities/WG/ESOA/

Abstract. A self-organising system functions without central control, and through contextual local interactions. Components achieve a simple task individually, but a complex collective behaviour emerges from their mutual interactions. Such a system modifies its structure and functionality to adapt to changes to requirements and to the environment based on previous experience. Nature provides examples of self-organisation, such as ants food foraging, molecules formation, or antibodies detection. Similarly, current software applications are driven by social interactions (negotiations, transactions), based on autonomous entities or agents, and run in highly dynamic environments. The issue of engineering applications, based on the principles of self-organisation to achieve robustness and adaptability, is gaining increasing interest in the software research community. The aim of this paper is to survey natural and artificial complex systems exhibiting emergent behaviour, and to outline the mechanisms enabling such behaviours.

Keywords: Self-organisation, self-organising application, emergence, collective behaviour, multi-agent systems.

1 Introduction

The study of self-organising systems is a field that has been explored at least since 1953 with the work done by Grassé [25], who studied the behaviour of insect societies. Many systems in nature demonstrate self-organisation, such as planets, cells, organisms and societies. All these systems exhibit recurrent properties inherent to self-organisation. The simplest form of self-organisation can be achieved by the arrangement of parts of a system in such a way as to be non-random. Considerable research has already been undertaken to study such systems. Self-organising systems are often encountered in many scientific areas including biology, chemistry, geology, sociology, and information technology.

A large number of software self-organising systems are designed based on natural mechanisms of self-organisation. Furthermore, recent research has been oriented towards introducing self-organisation mechanisms specifically for software applications, as well as entire development techniques supporting self-organisation [19]. This trend originates from the fact that current software applications need to cope with requirements and constraints stemming from the

G. Di Marzo Serugendo et al. (Eds.): AAMAS 2003 Ws ESOA, LNAI 2977, pp. 1–19, 2004.

increased dynamism, sophisticated resource control, autonomy and decentralisation inherent in contemporary business and social environments. The majority of these characteristics and constraints are the same as those which can be observed in natural systems exhibiting self-organisation.

This survey firstly defines self-organisation, and presents examples of self-organising systems taken from natural life. Subsequently, it describes the different mechanisms enabling social, natural and artificial organisations to achieve a coherent global behaviour. Finally, it reviews several software applications exhibiting a self-organising behaviour.

2 Self-Organising Systems

The notion of self-organisation is popular in many different research fields. Therefore, it is difficult to find a precise and concise definition of what is the meaning of the term self-organisation. However, it seems that similar properties are apparent and recurrent among the different definitions and research fields referring to self-organisation. The next sub-section defines what are the inherent characteristics of self-organisation. Subsequently, the following sub-sections describe a number of different types of self-organising systems. It is not the intention of this paper to provide an exhaustive review of all types of self-organising systems. Instead, self-organising systems are classified in three broad categories: physical, living and social systems. For each category a representative example is described.

2.1 Definition

Intuitively, self-organisation refers to the fact that a system's structure or organisation appears without explicit control or constraints from *outside* the system. In other words, the organisation is intrinsic to the self-organising system and results from *internal* constraints or mechanisms, due to *local interactions* between its components [11]. These interactions are often *indirect* thanks to the environment [25]. The system dynamics modifies also its *environment*, and the modifications of the external environment influence in turn the system, but without disturbing the internal mechanisms leading to organisation. The system evolves *dynamically* [9] either in time or space, it can maintain a stable form or can show transient phenomena. In fact, from these interactions, *emergent* properties appear transcending the properties of all the individual sub-units of the system [30]. One well-known example is that of a colony of ants sorting eggs without having a particular ant knowing and applying the sorting algorithm. The emergence is the fact that a structure, not explicitly represented at a lower level, appears at a higher level. With *no central control*, a complex collective behaviour raises then from simple local individual interactions. More generally, the field of complex systems studies emergent phenomenon, and self-organisation [8].

2.2 Physical Systems

A characteristic of physical self-organising systems is that many of them present a so called *critical value* in which the state of the system changes suddenly to another state under certain conditions (temperature, pressure, speed, ...). Thus, self-organisation is observed *globally* when the physical system moves from a chaotic disordered state to a stable one. For instance, a thermodynamic system such as a gas of particles has emergent properties, temperature and pressure, that do not derive from the description of an individual particle, defined by its position and velocity. Similarly, chemical reactions create new molecules that have properties that none of the atoms exhibit before the reaction takes place [8]. Moreover, the magnetisation of a multitude of spins[1] is a clear case of self-organisation because, under a certain temperature, the magnetic spins spontaneously rearrange themselves pointing all in the same direction thanks to a strong emerging magnetic field.

2.3 Living Systems

A scientific aspect in the study of living organisms is the determination of invariant in the evolution of natural systems. In particular the spontaneous appearance of an order in living complex systems due to the self-organisation. In the optic of biological research, the global emergence of a behaviour or a feature that can not be reduced to the properties of each system's component (molecules, agents, cells, ...) defines also the common meaning of self-organisation. One example described in [43] is the self-organisation of the cytoskeleton thanks to collective processes of reaction and diffusion of the cytoskeletal filaments. The cytoskeleton is the basis of the internal architecture of the cytoplasm of eukaryotic cells. Eukaryotic cells are cells of the higher organisms, containing a true nucleus bounded by a nuclear membrane. These cells are founded in animals and plants. Eukaryotic cells are often organised into organs to create higher levels of complexity and function thanks to metabolic processes. The obtained organ has a defined functionality that transcends all the functionality offered by each of its constitutive cells. These cells are the basic functioning units of life and evolve in step through external changes with the environment. These units of life (cells) have to use internal metabolic processes such as mutation or natural selection, to adapt to natural life's evolution. This is also the result of the self-organisation of cells. Other examples in human body are the human nervous system, or the immune system. Such living systems transparently manage vital functions, such as blood pressure, digestion, or antibodies creation.

2.4 Social Systems

Social insects organise themselves to perform activities such as food foraging or nests building. Cooperation among insects is realised through an indirect

[1] A spin is a tiny magnet.

communication mechanism, called stigmergy, and by interacting through their environment. Insects, such as ants, termites, or bees, mark their environment using a chemical volatile substance, called the pheromone, for example, as do ants to mark a food trail. Insects have simple behaviour, and none of them alone "knows" how to find food but their interaction gives rise to an organised society able to explore their environment, find food and efficiently inform the rest of the colony. The pheromonal information deposited by insects constitutes an indirect communication means through their environment.

Apart from animal societies, human beings organise themselves into advanced societies. Human beings use direct communication, they engage in negotiation, build whole economies and organise stock markets.

3 Self-Organising Mechanisms

This section presents the major self-organising mechanisms used in natural and software systems to achieve self-organisation.

3.1 Magnetic Fields

A self-organisation phenomenon has been studied in the structure of a piece of potentially magnetic material. A magnetic material consists of a multitude of tiny magnets or spins. The spins point in different directions cancelling their respective magnetic fields. At a lower level, the orientation of the spins is due to the random movements of the molecules in the material: the higher the temperature, the stronger these random movements of the molecules in the material. These molecular movements affect the spins making them difficult to orient in an ordered way. However, if the temperature decreases the spins spontaneously point in the same direction. In this case, the different magnetic fields now add up, producing a strong overall field. Magnetisation exhibits self-organisation because the orientation of the spins is variable and depends on the local neighbourhood. Under low temperature, the force between neighbouring spins is dominating and they tend to build order. Similar phenomena, are observed in the crystallisation from a liquid state, which is another common example of self-organisation.

3.2 Kohonen Neural Networks

Kohonen neural networks, also called self-organising maps, are useful for clustering applications [36]. They take their inspiration from brain cells, which are activated depending on the subject's location. Such a network is made of two neurons layers (input, and output), and usually follows a regular two-dimensional grid of neurons. This grid represents a topological model of the application to cluster. Indeed, the network maps similar data to the same, or adjacent, node of the grid, by projecting multi-dimensional vectors of data onto a two-dimensional regular grid, preserving the data clustering structure.