Natural Computing Series

For further volumes:
www.springer.com/series/4190

Giovanna Di Marzo Serugendo •
Marie-Pierre Gleizes • Anthony Karageorgos
Editors

Self-organising Software

From Natural to Artificial Adaptation

Springer

Editors

Dr. Giovanna Di Marzo Serugendo
Centre Universitaire d'Informatique
Université de Genève
Battelle, Bâtiment A, route de Drize 7
1227 Carouge Geneve
Switzerland
Giovanna.DiMarzo@unige.ch

Prof. Marie-Pierre Gleizes
Institut de Recherche en Informatique de Toulouse (IRIT)
Université Paul Sabatier
route de Narbonne 118
31062 Toulouse Cedex 9
France
Marie-Pierre.Gleizes@irit.fr

Dr. Anthony Karageorgos
Department of Forestry and Natural Environment Management
Technological Educational Institute of Larissa
Karditsa Branch, Terma Mavromihali
431 00 Karditsa
Greece
karageorgos@computer.org

ISSN 1619-7127 Natural Computing Series
ISBN 978-3-642-17347-9 e-ISBN 978-3-642-17348-6
DOI 10.1007/978-3-642-17348-6
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011938285

ACM Computing Classification (1998): I.2, F.1, D.1

Cover design: KünkelLopka GmbH

Printed on acid-free paper

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

Preface

Stable and dependable IT services and infrastructures are nowadays of paramount importance not only for modern enterprises but also for home users. However, as distributed information infrastructures continue to spread and grow, resulting in Internet-based, wireless and mobile systems, traditional solutions for managing and controlling the software that sustains them appear to have reached their limits. As a result, new challenges in software engineering have arisen demanding reliable, robust and scalable software systems operating in extremely dynamic and unstable environments, able to take care of themselves with a minimum of user intervention.

The main issue is that engineers of contemporary software systems and services can now only seldom rely on centralised control or management, high reliability of devices, or secure execution environments. For example, high-speed Internet connections, ad hoc sensor networks and ubiquitous computing devices have made possible to embed millions of sophisticated software components into interconnected and dynamically changing local environments. In such cases, centralised and deterministic control is practically impossible or at best prohibitively expensive. As a result, a natural solution to the problem can be building software systems capable of efficiently adapting to failures, component replacements and changes in the environment, without human intervention or centralised management. In other words, such systems should be able to autonomously change their organisation, or *self-organise*, as and when needed until they achieve, or *emerge to*, a satisfactory or selected state.

Self-organisation and emergence phenomena have long been observed in numerous natural systems, both living and non-living. Examples are the social order observed in human and animal social systems and the ordered orientation of magnetic spins appearing with lowering temperature in magnetic materials. As of recently, the idea that self-organisation and emergence can be harnessed for the purpose of solving tricky engineering problems inherent in modern IT systems has become increasingly popular. Researchers working in many diverse IT fields, such as computer networks, distributed software systems, operating systems and software agents, have begun to apply these ideas in a variety of problems with quite promising results.

These efforts have given rise to the term *Self-organising Software*. Self-organising software systems are able to dynamically change their structure and functionality without direct user intervention in response to changes occurring in user requirements, their environmental context and their internal state. The overall functionality delivered by self-organising software typically changes progressively, mostly in a nonlinear fashion, until it reaches (emerges to) a state where it satisfies the current system requirements, and therefore it is commonly referred to as *self-organising behaviour*. In the majority of cases, the overall self-organising behaviour is the result of execution of a number of interrelated individual components, which locally interact with each other aiming to achieve their local goals. Typical examples are systems based on software agents or distributed objects. The main characteristic of such systems is their ability to achieve complex collective results with relatively simple individual behaviours, applied without central or hierarchical control.

Self-organising software engineers often take inspiration from the real world, for example from biology, chemistry, sociology and the physical world and apply the observed principles to implement self-organising functionality in software. Typical such examples are software systems that reproduce socially based insect behaviour, such as ant-based systems, artificial life systems and robot swarms. Furthermore, detailed methodologies specifically targeting the engineering and control of self-organising behaviour in software have started being increasingly used. However, despite that advances made so far have started maturing, the majority of the work done is still scattered throughout research publications and technical reports, and there is no clear starting point for those wanting to get acquainted with the field, for example students and junior researchers.

The idea of this book germed during the meetings of the Technical Forum Group on Self-Organisation in Multi-Agent Systems,[1] supported by the EU-funded AgentLink[2] Network of Excellence. Some concepts and topics covered by this book have been the subject of debate, discussions and presentations during the group meetings. The decision to write a book then derived from the need to provide a unified view of self-organisation and its applicability to software in a neat way so that to be able to be used by instructors and readers in relevant courses, as well as by young researchers seeking an introductory, and at the same time a comprehensive, discussion of the issues involved.

As a result, this book provides an introductory yet comprehensive review of recent work done in the field of self-organising software. The first chapters elaborate extensively on *self-organisation concepts*, *mechanisms* and *engineering techniques*. They are supported by examples which aim to facilitate the reader in gaining a better understanding of the self-organisation approach and its applicability. In the subsequent chapters, the book pays attention to providing instructive descriptions of *application areas* where self-organisation has successfully been used in software to provide the solution. Such areas include manufacturing control, computer network management and security, P2P protocols, and optimisation problem solving.

All chapters are supplemented with puzzle questions, unsolved exercises and mini-projects aiming to be useful for teaching purposes. The solutions together with

[1] http://www.irit.fr/TFGSO

[2] http://www.agentlink.org

additional teaching materials are contained in an instructor's manual accompanying the book and available through the Technical Forum Group on Self-Organisation in Multi-Agent Systems web page.

London, UK

Giovanna Di Marzo Serugendo
Marie-Pierre Gleizes
Anthony Karageorgos

Acknowledgements

This book is the result of the collective effort of the community working on Self-organisation and Multi-Agent Systems. Its realisation would not have been possible without the contributions of a number of persons, to which we are greatly indebted.

First of all, we thank all the authors who accepted to participate in this book and to put notable efforts in contributing original pedagogical chapters and questions/exercises related to their chapters.

We are particularly grateful to all researchers that have since 2003 participated in the Technical Forum Group on Self-organisation in Multi-Agent Systems which we have been organising. Without them, we would have never reached that broad view of the research in the self-organising systems area that enabled us to produce this book. A detailed list of the participants, their activities and the meeting outcomes can be found on the Technical Forum Group on Self-organisation in Multi-Agent Systems web page.[1]

We thank the AgentLink EU Network of Excellence and particularly the Technical Fora organisers for supporting the meetings of the Technical Forum Group on Self-organisation in Multi-Agent Systems from 2003 to 2005. We also thank the European Workshop on Multi-Agent Systems (EUMAS) for hosting the group meetings as a separate event since 2006.

[1] http://www.irit.fr/TFGSO

Contents

Part III Engineering Artificial Self-organising Systems

Part IV Applications of Self-organising Software

Contributors

Carole Bernon IRIT, Université Paul Sabatier, Toulouse, France, Carole.Bernon@irit.fr

Christine Bourjot LORIA, Université Nancy, Nancy, France, Christine.Bourjot@loria.fr

Valérie Camps IRIT, Université Paul Sabatier, Toulouse, France, camps@irit.fr

Matteo Casadei Università di Bologna, 47023 Cesena, Italy, m.casadei@unibo.it

Vincent Chevrier LORIA, Université Nancy, Nancy, France, Vincent.Chevrier@loria.fr

Didier Desor URAFPA, Université Nancy-UHP, Nancy, France, Didier.Desor@scbiol.uhp-nancy.fr

Giovanna Di Marzo Serugendo Birkbeck College, University of London, London, UK, dimarzo@dcs.bbk.ac.uk

Pierpaolo Dondio Trinity College Dublin, Dublin, Ireland, dondiop@cs.tcd.ie

Noria Foukia Information Science Department, University of Otago, P.O. Box 56, Dunedin, New Zealand, noria.foukia@hesge.ch

Stephane Galland UTBM, Belfort, France, stephane.galland@utbm.fr

Nicolas Gaud UTBM, Belfort, France, nicolas.gaud@utbm.fr

Jean-Pierre Georgé IRIT, Université Paul Sabatier, Toulouse, France, george@irit.fr

Marie-Pierre Gleizes IRIT, Université Paul Sabatier, Toulouse, France, Marie-Pierre.Gleizes@irit.fr

Pierre Glize CNRS/IRIT, Université de Toulouse III, Toulouse, France, glize@irit.fr

Vincent Hilaire UTBM, Belfort, France, vincent.hilaire@utbm.fr

Bob Hulsebosch Telematica Instituut, Enschede, The Netherlands, Bob.Hulsebosch@telin.nl

Márk Jelasity University of Szeged and Hungarian Academy of Sciences, P.O. Box 652, 6701 Szeged, Hungary, jelasity@inf.u-szeged.hu

Anthony Karageorgos Technological Educational Institute of Larissa, Larissa, Greece, karageorgos@computer.org

Abderrafiâa Koukam UTBM, Belfort, France, abder.koukam@utbm.fr

Gabriele Lenzini University of Luxembourg, Luxembourg-Kirchberg, Luxembourg, Gabriele.Lenzini@uni.lu

Marco Mamei Università di Modena e Reggio Emilia, 42100 Reggio Emilia, Italy, marco.mamei@unimore.it

Jean-Pierre Mano UPETEC, Ramonville Saint-Agne, France, jean-pierre.mano@upetec.fr

Paul Marrow BT Group plc, Ipswich, UK, paul.marrow@bt.com

Melanie Middlemiss Information Science Department, University of Otago, P.O. Box 56, Dunedin, New Zealand, mmiddlemiss@infoscience.otago.ac.nz

Frédéric Migeon IRIT, Université Paul Sabatier, Toulouse, France, Frederic.Migeon@irit.fr

Gauthier Picard École Nationale Supérieure des Mines de Saint-Etienne, Saint-Etienne, France, picard@emse.fr

Sebastian Rodriguez CITAT, Universidad Tecnológica Nacional—Facultad Regional Tucumán, San Miguel de Tucumán, Argentina, rodriguez.sebastian@gmail.com

Jean-Marc Seigneur University of Geneva, Geneva, Switzerland, Jean-Marc.Seigneur@trustcomp.org

Vincent Thomas LORIA, Université Nancy, Nancy, France, Vincent.Thomas@loria.fr

Cynthia Villalba Università di Modena e Reggio Emilia, 42100 Reggio Emilia, Italy, cynthia.villalba@unimore.it

Mirko Viroli Università di Bologna, 47023 Cesena, Italy, mirko.viroli@unibo.it

Franco Zambonelli Università di Modena e Reggio Emilia, 42100 Reggio Emilia, Italy, franco.zambonelli@unimore.it

Acronyms

ABT	Asynchronous BackTracking
ACL	Act Communication Language
ACO	Ant Colony Optimisation
ADELFE	Atelier de Développement de Logiciels à Fonctionnalité Emergente—Toolkit to develop software with emergent functionality
ADOPT	Algorithm for Distributed Constraint Optimisation
AGR	Agent Group Role
AI	Artificial Intelligence
AIS	Artificial Immune System
AL	Artificial Life
AMAS	Adaptive Multi-Agent Systems
AMAS-ML	AMAS Modeling Language
AMP	Adenosine Monophosphate
ATL	Atlas Transformation Language
AOSE	Agent-Oriented Software Engineering
APC	Antigen Presenting Cell
APER	A Peer Entity Recognition
API	Application Programming Interface
APO	Asynchronous Partial Overlay
AUML	Agent Unified Modeling Language
AWCS	Asynchronous Weak-Commitment Search
B2C	Business to Consumer
B- and T-Cells	B lymphocytes and T lymphocytes
BT	Bluetooth
BTS	Base Transceiver Stations
cAMP	Cyclic Adenosine Monophosphate
CAS	Complex Adaptive Systems
CCD camera	Charge-Coupled Device Camera
COP	Constraint Optimisation Problems
CPU	Central Processing Unit

CSP	Constraint Satisfaction Problems
DAI	Distributed Artificial Intelligence
DANTE	Domain ANalysis and Trust Extraction. A trust model able to identify its input among application elements and exploit them for a trust computation
DB	Data Base
DBA	Distributed Breakout Algorithm
DC	Dendritic Cell
DCOP	Distributed Constraint Optimisation Problem
DIET	Decentralised Information Ecosystem Technologies
DisCSP	Distributed Constraint Satisfaction Problems
DNA	Deoxyribonucleic Acid
DSR	Danger Signal Receptors
DoS	Denial of Service
EDOS	Environment for the development and Distribution of Open Source software
EMF	Eclipse Modelling Framework
EPE	Emergent Programming Environment
ER	Entity Recognition
ERA	Environment Reactive rules and Agents
FIFO	First In First Out
FIPA	Foundation for Intelligent and Physical Agents
FIRA	Federation of International Robot-soccer Association
FOAF	Friend-Of-A-Friend
GA	Genetic Algorithm
GPD	Generalised Prisoner Dilemma
GPS	Global Positioning System
HIS	Human Immune System
HMAS	Holonic Multi-Agent Systems
ICT	Information and Communication Technologies
ID	Intrusion Detection
IDA	Intrusion Detection Agent
IDRS	Intrusion Detection and Response System
IDReAM	Intrusion Detection and Response executed with Agent Mobility
IDS	Intrusion Detection System
IR	Intrusion Response
IRA	Intrusion Response Agent
IRS	Intrusion Response System
IT	Information Technology
Ja-Net	Jack-in-the-Net
KQML	Knowledge Query and Manipulation Language
LS	Local Search
LTTM	Longo Temporal Trust Factors. A trust model introduced by Luca Longo that bases its computations entirely on the time distribution of entities' activity

MA	Mobile Agent
MAS	Multi-Agent Systems
MAY	Make Agents Yourself
MHC	Major Histocompatibility Complex
μADL	μ Architecture Description Language
MOCA	Modèle Organisationnel et Componentiel pour les systèmes multi-Agents
NCS	Non-Cooperative Situation
OEM	Original Equipment Manufacturer
OGSI	Open Grid Services Infrastructure
OMG	Object Management Group
OSS	Open Source Software
oAW	open-Architecture Ware
P2P	Peer to Peer
PAMP	Pathogen Associated Molecular Pattern
PC	Personal Computer
PD	Prisoner Dilemma
PDA	Personal Digital Assistant
PIN	Postal Index Number
PMI	Project Management Interface
PROSA	Product-Resource-Order-Staff Architecture
PRR	Pattern Recognition Receptors
PSO	Particle SwarmOptimisation
QA	Quality Assessment
RA	Representative Agent
REGRET	REGRET: A Reputation Model for Gregarious Societies
ReSpecT	Reaction Specification Tuples
RFID	Radio Frequency Identification
RIO	Role Interaction Organisation
RT	Recommending Trustworthiness
RUP	Rational Unified Process
SECURE	Secure Environments for Collaboration among Ubiquitous Roaming Entities
SI	Suspicion Index
SI model	Susceptible Infected model
SIR model	Susceptible Infected Removed model
SLAC	Selfish Link-based Adaption for Cooperation
SPEM	Software Process Engineering Metamodel
SQL	Structured Query Language
STAFF	Software Tool for Adaptive Flood Forecast
STM	Short Term Memory
TCP	Transmission Control Protocol
TCR	T-Cell Receptor
TFGSO	Technical Forum Group on Self-Organisation in multi-agent systems

TOTA	Tuples On The Air
Tropos4AS	Tropos for Adaptive Systems
TuCSoN	Tuple Centres over the Network
ULPC	User Local Probability Component
UML	Unified Modeling Language
VER	Vision Entity Recognition
VO	Virtual Organisation
WD	WorkDefinition
WfMS	Workflow Management Systems
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network
WWW	World Wide Web
XML	eXtensible Markup Language