

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 germinated during the meetings of the Technical Forum Group on Self-Organisation in Multi-Agent Systems,¹ supported by the EU-funded AgentLink² 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

Self-organising Software

From Natural to Artificial Adaptation

Di Marzo Serugendo, G.; Gleizes, M.-P.; Karageorgos, A.
(Eds.)

2011, XVIII, 462 p., Hardcover

ISBN: 978-3-642-17347-9