Autonomic Systems

Wolfgang Reif, Gerrit Anders,
Hella Seebach, Jan-Philipp Steghöfer,
Elisabeth André, Jörg Hähner,
Christian Müller-Schloer,
Theo Ungerer, Editors

Trustworthy Open Self-Organising Systems

Birkhäuser

Autonomic Systems

The Autonomic Systems series aims at creating a platform of communication between universities and industry by publishing research monographs, outstanding PhD theses, and peer reviewed compiled contributions on the latest developments in the field. It covers a broad range of topics from the theory of autonomic systems that are researched by academia and industry. Hence, cutting-edge research, prototypical case studies, as well as industrial applications are in the focus of this series. Fast reviewing provides a most convenient way to publish latest results in this rapid moving research area.

For further volumes:
http://www.springer.com/series/8123

Wolfgang Reif • Gerrit Anders • Hella Seebach •
Jan-Philipp Steghöfer • Elisabeth André •
Jörg Hähner • Christian Müller-Schloer •
Theo Ungerer
Editors

Trustworthy Open Self-Organising Systems

Birkhäuser

Editors
Wolfgang Reif
Institute for Software and Systems Engineering
University of Augsburg
Augsburg, Germany

Gerrit Anders
Institute for Software and Systems Engineering
University of Augsburg
Augsburg, Germany

Hella Seebach
Institute for Software and Systems Engineering
University of Augsburg
Augsburg, Germany

Jan-Philipp Steghöfer
Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg
Gothenburg, Sweden

Elisabeth André
Human-Centered Multimedia
University of Augsburg
Augsburg, Germany

Jörg Hähner
Organic Computing Group
University of Augsburg
Augsburg, Germany

Christian Müller-Schloer
Institute of Systems Engineering
University of Hannover
Hannover, Germany

Theo Ungerer
Systems and Networking Group
University of Augsburg
Augsburg, Germany

Autonomic Systems
ISBN 978-3-319-29199-4 ISBN 978-3-319-29201-4 (eBook)
DOI 10.1007/978-3-319-29201-4

Library of Congress Control Number: 2016941940

Mathematics Subject Classification (2010): 68T42, 68T37, 68T20, 68T05, 68W15, 68W27, 68N30

This book is published under the trade name Birkhäuser
The registered company is Springer International Publishing AG Switzerland
(www.birkhauser-science.com)

Foreword

OC-Trust, this is an acronym representing a research cooperation that addressed one of the core challenges of the emerging digitalisation of almost every facet of our professional and private lives. How can we develop trust into the widely autonomous provisioning of digital functionality and associated services? We expect those services to know what we want them to provide, but we are not physically capable and do not want to programme those multitudes of devices explicitly. So, we increasingly depend on their capability to self-configure, self-optimise, self-heal and self-protect, to name a few of the many so-called self-* properties. But how do we know to what extent they will actually satisfy our expectations? They should be aware of our personal preferences, but will they respect our privacy? If agents act autonomously, how can their operating environment distinguish between trustworthy and malicious agents? This kind of almost contradictory questions and requirements is concerned with the trustworthiness of artefacts that are meant to be self-organising and widely autonomous but nevertheless capable to adapt to potentially changing requirements of their execution environment. Research initiatives like autonomic computing and organic computing have emphasised from the beginning that trustworthiness should be seen as one of the key requirements, but they more or less focused on the development of generic architectures and methodology for providing desired functionality and organic behaviour in the best possible way. So, the German priority programme on organic computing successfully addressed fundamental system concepts supporting controlled self-organisation, as summarised in the compendium on "Organic Computing – A Paradigm Shift for Complex Systems". But it needed the additional initiative of research groups at Augsburg and Hanover to establish this complementary DFG research unit on "OC-Trust – Trustworthiness of Organic Computing Systems".

Wolfgang Reif, the spokesperson of this research unit, continued his work on software design for organic computing systems but focused now on "Formal Analysis and Software Architectures for Trustworthy Organic Computing". Christian Müller-Schloer, one of the core initiators of the organic computing research programme, and Jörg Hähner concentrated on top-down and bottom-up approaches to the "Generation of Self-organising Trust Communities". Theo Ungerer, another

core member of the Organic Computing Initiative, investigated "Trust Relationships in Between the Autonomous Units of OC Systems". Finally, since the interaction between man and machine is one of the key aspects of trustworthiness, Elisabeth André joined the research unit with her topic "HCI Design for Trustworthy Organic Computing". Looking at the research unit's record of meetings, workshops and special spring schools, it is obvious that they have been extremely active and productive. The TSOS workshop series on "Trustworthy Self-Organising Systems" as well as its successor, the SASOST Workshop, were essential for significant international recognition and provided a forum for exchange of ideas with other research groups. The "International Spring Schools on Trustworthy Self-Organising Systems" added specific input from international experts for the doctoral researchers in this research unit with a significant outreach to other research groups. This book now summarises the major results of this research unit on a topic that might prove to become most decisive for the public acceptance of technologies that are developed under a range of different, but highly related, headlines like "Internet of Things", "Cyber Physical Systems", "Industry 4.0" and "Smart City" (including energy and traffic systems as well as all kinds of citizen services), to name a few.

An interesting aspect of this book is the fact that it extends beyond the members of the research unit by including external experts on topics that are of interest for a more complete view on trustworthiness.

So, the DFG research unit OC-Trust not only generated a range of interesting concepts and results on trustworthiness of and within self-organising systems, but they also had a significant impact on the international research community and clearly showed the necessity and benefits of a transdisciplinary approach for a thorough understanding of the role of trustworthiness.

Karlsruhe, Germany Hartmut Schmeck
November, 2015

Preface

Our technological landscape is ever-changing. Interconnected devices interact with other devices as well as people in an increasingly autonomous fashion. This core idea manifests itself in several aspiring areas of technology – from the "Internet of Things" to "Industry 4.0". It seems all too obvious that these entities cannot be controlled by individuals or even organisations but rather require sophisticated self-organisation mechanisms to implement various self-* properties without centralised control. This scientific challenge led to initiatives such as autonomic computing or organic computing that proposed important basic architectures, models and algorithms. Particularly in terms of robustness towards failures, these systems show the potential of outperforming conventional, rigid systems. When widening the scope of application of self-organising systems to critical domains that are more open and consist of heterogeneous participants, an essential question accompanies the more widespread adoption: How can we make these systems *trustworthy*?

More specifically, in 2009 the DFG[1] research unit "Trustworthiness of Organic Computing Systems" (OC-Trust) set out to develop methods to construct self-organising multi-agent systems that are deemed trustworthy by their users, by other systems interacting with them and by authorities and even organisations that certify and deploy systems in safety- or mission-critical environments. Positive aspects of self-organisation, such as increased robustness or other positive emergent effects, shall, however, not be sacrificed. The common denominator of the bundled research efforts is the scientific treatment of various facets of *trust* in technical systems. Trust manifests itself in the system design, e.g. by countermeasures against ill-behaving or little predictable agents, and helps to reduce the impact of such entities on the overall system performance. Among technical systems benefiting from trust management, one particular system class is selected to serve as a prominent representative. It can be roughly categorised as *open, heterogeneous, self-organising, multi-agent systems* and is visualised in Fig. 1. Systems in this class share several features that require individual attention:

[1] German Research Foundation (*Deutsche Forschungsgemeinschaft*)

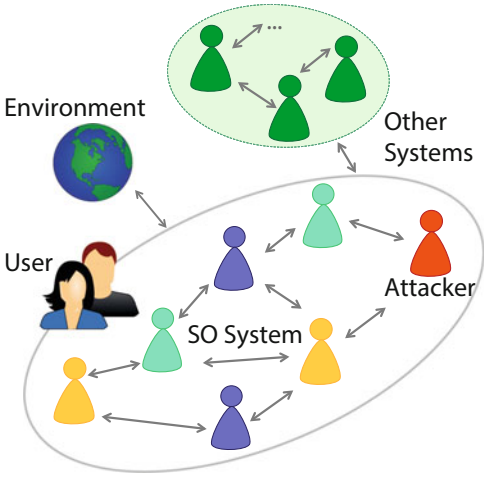

Fig. 1 Open self-organising multi-agent systems composed of heterogeneous agents. Examples thereof are detailed in subsequent chapters

- Components are represented by agents that interact via a self-organised communication and collaboration structure to, e.g. avoid excessive broadcasting and enable effective problem decomposition.
- The system interacts with other systems, and a single agent may even act on behalf of a larger subsystem in a systems-of-systems approach.
- Due to its deployment as an autonomous entity in a dynamic environment, uncertainty about interaction partners (and their possibly malicious intentions in the case of attackers) and exogenous factors is omnipresent – hence, the benevolence assumption is abandoned.
- Users are present "in the loop" and constantly interact with the software surrounding them – as long as they trust it.

Clearly, these diverse challenges require a collaborative effort that is reflected in the projects the research groups undertook and whose results of 6 years of research form the core of this book. Chapter 1 provides an overview of the properties of computational trust and its different uses. These are concretised in the subsequent chapters. Wolfgang Reif and his group (see Chap. 2) investigated methods that enable scalable, robust optimisation to control systems subject to strong environmental influences and physical constraints. Christian Müller-Schloer (see Chap. 4) and his team provided means to incentivise cooperative or to sanction malicious behaviour in a group of agents. In this context, Jörg Hähner (see Chap. 5) and his team devised mechanisms to form groups of agents that mutually trust each other. Theo Ungerer (see Chap. 6) established with his group how various self-* properties can be efficiently monitored and allowed for selective service placement in middlewares for parallel algorithms and distributed systems, in general. To accommodate the users' interests, in particular its trust in a self-organising system,

Elisabeth André (see Chap. 3) and her team worked on explicit user trust models that capture the effects of system actions on the users' experienced trust and take these factors into consideration. Measuring, formalising and interpreting various facets of trust as well as the incorporation of this knowledge into decision-making is a common theme that transcends all OC-Trust projects. Many of the concepts and algorithms were developed in close cooperation of the project partners, reflected by 35 joint publications. More than 20 internal project meetings over the course of 6 years offered room and time for the fundamental discussions that led to those results.

To illustrate the developed techniques and to instantiate the system class, three jointly used case studies were devised. All of them are based on the Trust-Enabling Middleware that offers communication interfaces and access to a generic infrastructure for application-specific trust metrics. The *Trusted Desktop Grid* deals with open, social agent environments that jointly process computing tasks. As a self-organised collaboration structure, the concept of trusted communities consisting of trustworthy agents is in the focus. Trust-based *Autonomous Virtual Power Plants* allow for a self-organised, robust and scalable control of a large number of power plants in a hierarchical way. Uncertainty introduced by volatile energy sources poses tremendous challenge to the system which has to keep supply and demand of power in balance at all times. *Multi-user multi-display environments* have users interact with a system on both public and private devices. With several participants at the same device, privacy and usability concerns become relevant when it comes to deciding which content should be shown. User preferences guide these decisions which are evaluated at runtime on a dynamic user trust model.

Certainly, the research unit did not work in isolation on these fundamental topics but rather built on top of established theories, models and algorithms and extended the literature substantially. This fact is reflected by the structure of this book which includes three invited contributions by selected experts from the domain of trust in multi-agent systems. Jeremy Pitt (see Chap. 7) discusses formal models of several social processes for open distributed systems and, in a sense, removes the restriction on the social concept of trust otherwise so prominent in this book. Cristiano Castelfranchi and Rino Falcone (see Chap. 8) add various other factors to the discussion on trust in self-organising, sociotechnical systems. Natasha Dwyer and Stephen Marsh (see Chap. 9) conclude the book by asking the interesting and relevant question whether a digital environment empowered users to proceed on their own terms.

These contributions are witness to the fact that the research unit enjoyed great visibility in the scientific community and put serious efforts into the dissemination of its results. Papers that resulted from the projects were regularly presented at international conferences such as the IEEE International Conference on Self-Organising and Self-Adaptive Systems (SASO), the International Conference on Autonomic Computing (ICAC) or the International Conference on Architecture of Computing Systems (ARCS), to name a few. Especially at SASO, nine editions of workshops on topics related to OC-Trust were held, comprising the workshops on trustworthy self-organising systems (TSOS), sociotechnical concepts (SASOST) and

quality assurance for self-organising systems (QA4SASO). These workshops turned out to be valuable regular additions to the programme of SASO and led to fruitful discussions. But of course, until sound publications can be written, doctoral students need to be exposed to and guided towards recent scientific work. It is for this cause that the research unit conducted two spring schools on "trustworthy self-organising systems" and three gender workshops to invite prominent researchers and foster future cooperations. Furthermore, due to this encouraging culture, several doctoral researchers were already invited to personally serve in programme committees or panels at both conferences and workshops. Additionally, the 10th edition of SASO will be held in Augsburg in 2016 with demonstrations of the OC-Trust projects.

Besides these community-oriented activities and OC-Trust-internal cooperations, some of the results emerged from collaborations with external partners. Especially papers at the frontiers of trustworthy self-organising systems that could benefit from input from other disciplines were written with OFFIS at the University of Oldenburg, the Imperial College London, the Max-Planck-Institute in Tübingen and the KU Leuven. Interesting meetings took place with NEC Laboratories, the University of Calgary, the University of Duisburg-Essen and the LMU in Munich. Additionally, invited talks at the Stadtwerke Munich, Phoenix Contact, the SORules workshop in London and the Helmholtz centre in Munich showed increased interest from both industry and academia. Wolfgang Reif and Christian Müller-Schlöer furthermore spent sabbatical terms at NICTA in Australia and Telecom ParisTech, respectively, to work intensively on related topics. All shall be mentioned to value their feedback that influenced and shaped the research unit.

Results of OC-Trust found their way into three courses at the universities of Augsburg and Hanover. Therefore, motivated students were well-prepared to conduct their own research in self-organisation in their thesis works. Many of those results found their way into proper publications. Finally, 13 doctoral researchers found challenging questions to complete their dissertations in the research unit. It is due to their continuous efforts that the project succeeded the way it did, in answering some questions but *asking* many important new ones. As a starting point for new directions, a Dagstuhl seminar on "Social Concepts in Self-organising Systems" was initiated by the research unit in December 2015. We are confident that the achieved results presented in this book show great promise for both research and applications and look forward to an increasing number of trustworthy self-organising systems in our future environment.

Finally, many thanks go to the contributing authors, in particular of the invited contributions that enriched the book tremendously. We are indebted to the German Research Foundation for sponsoring the research unit OC-Trust (FOR 1085).

Augsburg, Germany
January, 2016

Wolfgang Reif (head of OC-Trust)
Alexander Schiendorfer
Hella Seebach
Gerrit Anders

Contents

List of Contributors

Gerrit Anders Institute for Software and Systems Engineering, University of Augsburg, Augsburg, Germany

Elisabeth André Human-Centered Multimedia, University of Augsburg, Augsburg, Germany

Yvonne Bernard Institute of Systems Engineering, Leibniz Universität Hannover, Hannover, Germany

Cristiano Castelfranchi Institute of Cognitive Sciences and Technologies, National Research Council, Roma, Italy

Natasha Dwyer College of Arts, Victoria University, Melbourne, VIC, Australia

Benedikt Eberhardinger Institute for Software and Systems Engineering, University of Augsburg, Augsburg, Germany

Sarah Edenhofer Organic Computing Group, University of Augsburg, Augsburg, Germany

Rino Falcone Institute of Cognitive Sciences and Technologies, National Research Council, Roma, Italy

Jörg Hähner Organic Computing Group, University of Augsburg, Augsburg, Germany

Stephan Hammer Human-Centered Multimedia, University of Augsburg, Augsburg, Germany

Jan Kantert Institute of Systems Engineering, Leibniz Universität Hannover, Hannover, Germany

Lukas Klejnowski Institute of Systems Engineering, Leibniz Universität Hannover, Hannover, Germany

Oliver Kosak Institute for Software and Systems Engineering, University of Augsburg, Augsburg, Germany

Stephen Marsh Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, ON, Canada

Christian Müller-Schloer Institute of Systems Engineering, University of Hannover, Hannover, Germany

Nizar Msadek Systems and Networking Group, University of Augsburg, Augsburg, Germany

Jeremy Pitt Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Wolfgang Reif Institute for Software and Systems Engineering, University of Augsburg, Augsburg, Germany

Alexander Schiendorfer Institute for Software and Systems Engineering, University of Augsburg, Augsburg, Germany

Hella Seebach Institute for Software and Systems Engineering, University of Augsburg, Augsburg, Germany

Florian Siefert Institute for Software and Systems Engineering, University of Augsburg, Augsburg, Germany

Jan-Philipp Steghöfer Department of Computer Science and Engineering, Chalmers University of Technology | University of Gothenburg, Gothenburg, Sweden

Sven Tomforde Organic Computing Group, University of Augsburg, Augsburg, Germany

Theo Ungerer Systems and Networking Group, University of Augsburg, Augsburg, Germany

Michael Wißner Human-Centered Multimedia, University of Augsburg, Augsburg, Germany

Chapter 1
The Social Concept of Trust as Enabler for Robustness in Open Self-Organising Systems

Gerrit Anders, Hella Seebach, Jan-Philipp Steghöfer, Wolfgang Reif, Elisabeth André, Jörg Hähner, Christian Müller-Schloer, and Theo Ungerer

Abstract The participants in open self-organising systems, including users and autonomous agents, operate in a highly uncertain environment in which the agents' benevolence cannot be assumed. One way to address this challenge is to use computational trust. By extending the notion of trust as a qualifier of relationships between agents and incorporating trust into the agents' decisions, they can cope with uncertainties stemming from unintentional as well as intentional misbehaviour. As a consequence, the system's robustness and efficiency increases. In this context, we show how an extended notion of trust can be used in the formation of system structures, algorithmically to mitigate uncertainties in task and resource allocation, and as a sanctioning and incentive mechanism. Beyond that, we outline how the

G. Anders (✉) • H. Seebach
Institute for Software and Systems Engineering, University of Augsburg, Augsburg, Germany
e-mail: anders@isse.de; seebach@isse.de

J.-P. Steghöfer
Department of Computer Science and Engineering, Chalmers University of Technology | University of Gothenburg, Gothenburg, Sweden
e-mail: jan-philipp.steghofer@cse.gu.se

W. Reif
Institute for Software and Systems Engineering, University of Augsburg, Augsburg, Germany
e-mail: reif@isse.de

E. André
Human-Centered Multimedia, University of Augsburg, Augsburg, Germany
e-mail: elisabeth.andre@informatik.uni-augsburg.de

J. Hähner
Organic Computing Group, University of Augsburg, Augsburg, Germany
e-mail: jorg.hahner@informatik.uni-augsburg.de

C. Müller-Schloer
Institute of Systems Engineering, University of Hannover, Hannover, Germany
e-mail: cms@sra.uni-hannover.de

T. Ungerer
Systems and Networking Group, University of Augsburg, Augsburg, Germany
e-mail: theo.ungerer@informatik.uni-augsburg.de

W. Reif et al. (eds.), *Trustworthy Open Self-Organising Systems*,
Autonomic Systems, DOI 10.1007/978-3-319-29201-4_1

users' trust in a self-organising system can be increased, which is decisive for the acceptance of these systems.

Keywords Computational trust • Uncertainty • Self-organisation • Open MAS • Robustness

1.1 Trust as a Measure of Uncertainty in Open Self-Organising Systems

In open self-organising systems, different participants, such as autonomous agents, human users, and other systems, work together with a strong influence of the environment. These participants communicate and cooperate at runtime in unforeseeable ways and do not always follow the intent of the system designers. They can pursue different goals, and it cannot be assumed that they are intrinsically motivated to contribute towards a common system goal [1, 2]. Beyond that, a participant's behaviour can vary over time. As there is also limited knowledge about and control over the behaviour of the participants in the system, only weak assumptions about them can be made – in particular, we have to abandon assumptions of benevolence of the autonomous agents. The system participants therefore have to deal with both unintentional as well as intentional misbehaviour of others. This situation is aggravated by additional factors that increase uncertainties as they influence the system in unpredictable ways. These factors comprise the environment, other systems the agents interact with, or the users. Another form of openness often regarded in multi-agent systems (MAS) research is present when agents can arbitrarily enter and leave the system [3]. Especially in safety- or mission-critical domains, such as manufacturing or power management, these challenges have to be taken very seriously.

In this chapter, we argue that trust – as a measure of uncertainty – is a key concept for achieving robustness and efficiency in open self-organising systems. The classic notion of *computational trust* in the MAS community is focused on the credibility of agents, i.e. the degree to which they fulfil their commitments. This view stems mainly from psychological and sociological research [4] and boils down to the selection of interaction partners in order to maximise the utility of individual interactions. Economic [5, 6] and computer science [7, 8] literature characterise trust as instrumental to manage *expectations* about others. In computer science, the term "computational trust" is used to stress that the trust in a system or a system's part, such as an agent, is assessed by means of a well-defined metric. Since both (a part of) the system or a human being can act in the role of the trustor, we can differentiate between system-to-system and user-to-system trust. Often, a strong connection between trust and risk is emphasised [9] since interactions that incur a high risk for the participating agents require a high expectation of the others' willingness to contribute in a beneficial manner. An empirically justified expectation reduces the *uncertainty* about the behaviour of another agent [10]. In computing systems, this is often captured by a numerical *trust value* [11].

For these reasons, trust is an essential constituent of ensembles of cooperating agents, be they human or technical systems. Game-theoretical considerations show that trust can help to avoid getting trapped in the tragedy of the commons. Kantert et al. [12] provide such lines of thoughts in the context of Desktop Grid Computing. In general, trust induces a probability distribution over types of interaction partners of different trustworthiness in a Bayesian game. In this setting, agents have to choose their actions given probabilistic knowledge about each other's trustworthiness.

As mentioned above, we claim that trust is a key concept for achieving robustness. In this chapter, we define robustness in two dimensions. The first dimension of robustness addresses a system's ability to resist internal or external disturbances. Such disturbances result from (un)intentional misbehaving agents, for instance. A system exhibiting this type of robustness promises to remain in acceptable states and thus to maintain its functionality despite detrimental influences. The second dimension of robustness considers a system's ability to return into an acceptable state after a disturbance occurred that caused the system to leave the acceptance space. This type of robustness characterises a system's ability to restore its functionality. Consequently, the magnitude of disturbances the system can cope with (first dimension) and the duration of the deviation from acceptable states (second dimension) can be used to quantify the robustness. Both dimensions of robustness quantify the system's ability to fulfil its tasks. In contrast to a mere passive resistance, self-organising systems can actively increase their robustness by means of reactive or proactive measures. In open systems, these measures can be based on participants' trustworthiness, which allows the system to anticipate different sources of uncertainties.

In this chapter, we give an overview of the uses of computational trust (see Sect. 1.3) to deal with uncertainties arising in open self-organising systems. We show that these uses extend the classical use of selecting interaction partners and are based on the same life-cycle describing how trust values evolve over time (see Sect. 1.2). In detail, we demonstrate how trust models can be used to inform self-organisation processes (see Sect. 1.3.1); to optimise for critical or likely situations in uncertain environments (see Sect. 1.3.2); to sanction or incentivise agents in normative systems (see Sect. 1.3.3); and to represent the social relationships of the system's users (see Sect. 1.3.4). Section 1.4 concludes the chapter by emphasising that trust proves to be very useful to increase robustness and efficiency in open self-organising systems.

1.2 Computational Trust

Trust is usually measured as a numerical value, often normalised to values between 0 and 1. In [13], an agent's trust value is either very high or very low if the agent is either always expected to behave beneficially or never; if the value is between these extremes, the agent behaves in an unpredictable fashion and thus interactions with it are afflicted with a high uncertainty. Such a simple representation of trust is used

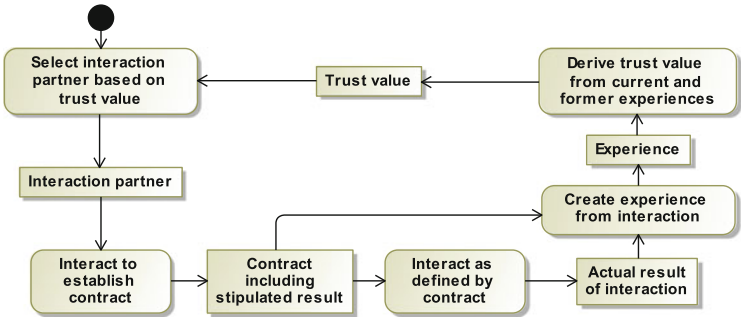

Fig. 1.1 The life-cycle of trust values derived from experiences (adapted from [OCT3])

in many trust models (for an overview, see, e.g. [14]). However, numerous other interpretations and representations of trust exist. Anders et al. [OCT1], for instance, regard a trust value as an expected deviation from a prediction or promise. The lower an agent's trust value, the higher the expected deviation from its predictions or promises. A supplementary value, called *predictability*, quantifies the variance in the agent's behaviour and is used to indicate the certainty that the expected deviation actually occurs. Other representations based on more complex data structures (e.g. trust-based scenarios [OCT2] or elaborate reputation systems [15]) are able to capture further properties, such as time-dependent behaviour in the sense that an agent's behaviour depends on the time of day or that its behaviour depends on those it showed in previous time steps. Before discussing the general properties of trust, we illustrate the life-cycle of trust values which can be transferred to most of the other representations of trust.

The Life-Cycle of Trust Values. There is a general way of thinking about the origin of trust values that is independent of the way they are used (see Fig. 1.1). Two or more parties commit to a (potentially implicit) *contract* [16] that defines an *interaction* (possibly composed of several distinct steps) as well as its *stipulated result*. The *actual result* of the interaction can be compared to what was stipulated in the contract, thus yielding an *experience* for each party [17]. Ultimately, an agent uses its experiences and a trust metric to derive a *trust value* for each of its interaction partners. The trust values, in turn, inform future interactions.

Falcone et al. [18] criticised that many trust models are void of semantics of how the generated trust values have to be interpreted. It is, e.g. often not defined what a trust value of, say, 0.5 actually expresses or which trust value should be assigned to a new agent (the problem of *initial trust*, see, e.g. [19]). If a trust model has precise semantics, meaning a clearly defined way to interpret generated trust values, such an abstracting quantification can still be valid, though.

Properties of Trust. The life-cycle shows why trust values are *subjective*. As each agent makes its own experiences with others, it forms a personal opinion (i.e. a trust value) based on these unique experiences. Thus, the experiences of two agents with the same partner can vary tremendously. Additionally, agents can use different metrics to assess trust values and apply different requirements to the behaviour of others, thus implementing different trust models. The same arguments can be used to argue against *transitivity* of trust [20]. An exception are recommendations as a form of *indirect trust* or *reputation* (see discussion below) that have to be based on a mutual understanding of the valuation of an agent's behaviour.

Further, it is crucial to consider the *context* in which interactions occur. The context includes, e.g. the roles the agents play in the interaction, its contract, or environmental circumstances. Comparing experiences to each other in different contexts is difficult: You might trust your doctor to fix you, but not necessarily to fix your car. Falcone et al. [18] relate to context when they mention the "competence belief" an agent has about another. Competence is specific to a certain goal that the trusting agent believes the potential partner is capable to pursue. Agents that are deemed competent for one goal can be incompetent for another. Other authors use, e.g. "circumstance" [21] or "domain of interaction" [22] to denote context.

A trust value can also be supplemented by a measure of *confidence* [OCT4] or *certainty* [23, 24] that indicates the degree of certainty that a trust value describes the actual observable behaviour of an agent. Such an additional value can be based on several criteria, such as how many experiences were used for the calculation of the trust value, how old these experiences are, or how much the experiences differed. It is also possible to take the social relationships between the agents into account [25] or to distinguish short-term and long-term behaviour in order to identify changing behaviour. As with trust values themselves, the initialisation of confidence can be problematic. In human interactions, different trust dispositions are common where people approach newcomers differently and are willing to put more initial trust in them than others [26]. The experiences made by these trusting individuals can then be used by others to judge newcomers. Such a mechanism is especially useful during the exploratory phase after the start of a system [19].

Reputation. In open self-organising systems, interaction partners can change often, e.g. due to alterations in system structure or inclusion of new agents. Since the agents' benevolence cannot be assumed, they might not be willing to communicate their true intentions [27]. To deal with this situation, a reputation system can be used which combines the opinions of agents and generates recommendations [7]. This enables cooperation between agents that do not know or have only little experience with each other. To make adequate decisions, agents can rely on a combination of direct trust and reputation. To this end, several approaches [15, OCT5] propose to use confidence or similar metrics to dynamically weigh the influence of direct trust and reputation, e.g. depending on the number of direct experiences. Due to the subjective nature of trust and because agents might lie about the trustworthiness of others, it is often also desirable to weigh the impact a recommending agent, called *witness*, has on the reputation value. The *neighbour trust metric* [OCT6] as well as

DTMAS [28] propose to increase the influence of a witness with the similarity of the provided valuation to the one of the requesting agent. If the difference is too large, the witness can even be excluded from the calculation. This allows the system to deal with false reports. Further approaches that incentivise agents to provide truthful reports are discussed in Sect. 1.3.3. Providing reputation data can also be regarded as a special context in which witnesses are assessed according to the quality of their recommendations. In an even more fine-grained system, the context can also include information for which kind of interaction the recommendation is given. Whenever a reputation system is used, there has to be a consensus among the agents about the meaning of trust and reputation values. A common trust model can fulfil this purpose.

Accountability, Deceit, and Collusion. Open systems with little control over the agents are prone to exploitation from egoistic or malevolent agents. Therefore, special measures have to be taken to provide accountability of the agents and to prevent collusion. For an overview of attacks on trust and reputation management systems, see, e.g. [29]. Specific countermeasures are often system- or domain-specific, such as those presented for mobile ad-hoc networks in [30] or electronic markets in [31]. An important part of fraud prevention is a well-designed incentive system in combination with efficient monitoring facilities [32].

1.3 Different Uses of Trust in Open Self-Organising Systems

As discussed in Sect. 1.1, trust is traditionally used for selecting appropriate interaction partners. Bernard et al. [OCT7] call an agent's set of preferred interaction partners whose trust value is above a predefined threshold its *Implicit Trusted Community* (iTC). From the local view of a single agent, its interaction partners are selected through an implicit formation process. Note that this process is fully decentralised and thus not governed or controlled by an explicit authority. Because the agents do not coordinate their selections, the members of an iTC do not necessarily mutually trust each other. Yet this simple approach successfully excludes notoriously untrustworthy agents from most interactions.

In the following, we give an overview of four different uses of trust that extend this traditional use. First, we consider the trust-based formation of explicit organisations that allow large-scale open systems to deal with untrustworthy agents (see Sect. 1.3.1). Second, robust task and resource allocation promises to improve the system's stability and efficiency in uncertain environments (see Sect. 1.3.2). Third, uncertainties resulting from intentional misbehaviour can be reduced by means of appropriate incentives – employing trust as a sanctioning mechanism is one of several possibilities (see Sect. 1.3.3). Fourth, we outline measures how user trust in open environments can be increased (see Sect. 1.3.4).

1.3.1 Trust to Structure Large-Scale Open Systems

In essence, self-organisation enables a system to autonomously form and adapt a structure that supports its objectives under changing conditions. The main reasons for agents to form organisations are to achieve scalability and promote cooperation in order to accomplish their own or the system's goals [33]. While scalability is the result of the accompanying problem decomposition, cooperation is necessary due to the agents' limited resources and capabilities. There are a multitude of paradigms and algorithms for establishing organisations in literature, such as *teams* [33] and *coalition formation* [34]: While teams assume altruistic behaviour, coalition formation is used in systems consisting of self-interested and individually rational agents.

The participants of open systems might not only show self-interested behaviour but also lie about their capabilities, the utility of performing an action, etc. Consequently, the selection of suitable cooperation partners becomes even more important. Since suitable coalition structures depend on the agents' promised contributions, the system has to make sure that these promises are kept and all coalition members pursue a common goal. To this end, extensions of coalition formation incorporating trust into the agents' decisions have been presented in [35, 36]. In contrast to coalitions, *clans* [37] are long-lived. Given that cooperation is likely to be most beneficial and least uncertain with trustworthy agents, clans are groups of agents that mutually trust each other. A similar concept, called *Explicit Trusted Communities* (eTCs), for the domain of Desktop Grid Computing has been proposed in [OCT8]. The main difference to clans and coalitions is that each eTC is represented by an explicit manager which administrates memberships, deals with conflicts, and governs the participating agents with norms. By preferring interactions with trustworthy agents (or even restricting them to these agents), clans and eTCs incentivise untrustworthy agents to change their behaviour (see Sect. 1.3.3 for incentive mechanisms and norms). Ultimately, this procedure aims at a more efficient and robust system – at least with regard to the members of clans or eTCs. While these types of organisations are not necessarily limited to intentional misbehaviour, they assume that agents can be excluded from other parts of the system without jeopardising the overall system's stability and efficiency. This is why trustworthy agents can form exclusive groups.

However, there are situations in which untrustworthy agents can or should not be excluded from the system, e.g. if the system depends on their resources or if they can provide them in a particularly cost-efficient way. In power management systems, for instance, although the output of solar power plants is difficult to predict (their volatility is mirrored in low trust values), they should not be turned off because of their low-cost generation. If, in such a situation, scalability requires the agents to self-organise into subsystems, other types of organisations are needed to deal with untrustworthy agents. One possibility is the formation of *homogeneous partitionings* [OCT9] where organisations are as similar as possible with respect to certain criteria that have been identified as supporting the system's goals (including

their mean trustworthiness). This idea is based on the assumption that a centralised system imposes an upper bound on the ratio between trustworthy and untrustworthy agents: Given the uncertainties introduced by untrustworthy agents, the centralised control over trustworthy agents allows the system to fulfil its task as well as possible. If all organisations exhibit similar characteristics with respect to the identified criteria, such as a similar ratio between trustworthy and untrustworthy agents, they approximate the corresponding ratio of the centralised system. Consequently, they also inherit its positive properties. Ideally, this results in an organisational structure in which each organisation can deal with its untrustworthy agents internally without affecting or involving other organisations. In such situations, homogeneous partitioning increases the system's robustness and efficiency, and should be preferred to organisations consisting of homogeneous agents. A similar goal has been pursued in [38] where agents mitigate uncertainties originating from unintentional misbehaviour by forming coalitions in a way that they cancel each other out.

1.3.2 Trust as a Basis for Robust Task or Resource Allocation

In many applications, a MAS has to solve a task or resource allocation problem in which a set of tasks is to be allocated to agents, or a set of the agents have to provide a certain amount of resources in order to satisfy a given demand [39]. Due to the agents' limited resources and knowledge, they usually have to cooperate in order to achieve the goal. In open systems, finding an adequate allocation is even more difficult since agents might not provide resources or fulfil the task as promised and the actual demand that has to be satisfied or the resources required to perform a task might not be known exactly beforehand. Both types of uncertainties can be attributed to unintentional or intentional misbehaviour of the system's participants or its environment [OCT1]. If the system's stability or efficiency hinges on how well the agents fulfil the tasks or meet the demand – e.g. think about the demand of electric load in a smart grid application – techniques for robust task or resource allocation have to be regarded. In general, the way a robust allocation can be obtained depends on the type of misbehaviour.

Unintentional misbehaviour is introduced by external forces, such as current weather conditions. While this type of misbehaviour cannot be actively reduced, trust can be used to quantify and anticipate the uncertainties [10]. Incorporating trust into the decision-making process allows the system to optimise for *expectations*, such as the expected probability of success [40]. In [OCT10, OCT11], a self-organising middleware incorporating a trust-aware load-balancing mechanism assigns important services to trustworthy nodes in order to increase the services' expected availability. Similarly, participants of a Desktop Grid Computing system delegate the calculation of jobs to trustworthy agents, i.e. to members of their eTC, to improve their expected outcome (see Sect. 1.3.1). If the *predictability* (cf. "confidence") of an agent's behaviour depends on its state, allocations can also be made in a way that promotes predictable behaviour [OCT1]. For highly

volatile environments in which dependencies in a sequence of observed behaviour have to be captured, a more expressive trust model, called *Trust-Based Scenario Trees* (TBSTs), has been proposed in [OCT2]. Basically, each TBST represents an empirical probability mass function that approximates the observed stochastic process. In contrast to trust models that capture the expected uncertainty or its variation, a TBST holds multiple possible scenarios, each with a probability of occurrence, of how the uncertainty might develop over a sequence of time steps. As opposed to the concept of *scenario trees* as known from the domain of operations research [41], TBSTs make only few assumptions about the underlying stochastic process. Further, they have been developed with the purpose of being *learned online* by agents with possibly low computational power. Combined with the principle of *stochastic programming* [42], agents can obtain robust allocations dynamically at runtime.

Intentional misbehaviour can be ascribed to agents that lie about some private information needed to decide about an adequate allocation, such as the cost or probability of performing a task successfully [40, 43]. Contrary to unintentional misbehaviour, uncertainties originating from intentional misbehaviour can be avoided. The field of *mechanism design* [40] studies how a system has to work in order to *incentivise* its self-interested, strategic, and individually rational participants to tell the truth. Further details concerning this matter are discussed in the following section.

1.3.3 Trust as a Sanctioning and Incentive Mechanism

Employing the techniques of *mechanism design* (MD) can guarantee efficiency (maximisation of the agents' overall utility), individual rationality (the agents' utility of participating in the scheme is non-negative), and incentive compatibility (the agents are best off revealing their true type) [44]. The latter property is of particular interest in open systems when agents have to be incentivised to disclose their private information needed to make decisions. In other words, MD can be used to incentivise individually rational agents to behave benevolently, that is, to ensure their trustworthy behaviour. *Fault-Tolerant MD* [43] and *Trust-Based MD* [40] address the issue of agents that have a probability of failure – quantified by a trust value – when performing an assigned task. Both approaches investigate the problem that reasonable task allocations depend on truthfully reported trust values. While each agent calculates and reports its own trust value in Fault-Tolerant MD [43], reputation values stemming from subjective trust measurements are considered in Trust-Based MD [40]. The ideas of MD have been adopted in various market-based approaches in which pricing mechanisms prevent agents from gaming the system [38, 45]. Depending on the regarded problem, it is often hard to devise a proper mechanism guaranteeing incentive compatibility, though, especially in case of unintentional misbehaviour. In these cases, it is still possible to use penalty schemes to increase the agents' risk that providing false reports or promises that

cannot be kept is detrimental to their utilities [44, OCT1]. Often, corresponding incentives can rely on the agents' trustworthiness. In electronic markets, trustworthy agents can obtain price premiums or price discounts [6]. In [OCT1], for instance, agents showing well-predictable behaviour can demand higher payments. Preferring trustworthy interaction partners or creating groups of trustworthy agents that benefit from a mutual increase in efficiency (cf. eTCs discussed in Sect. 1.3.1) also incentivises benevolent behaviour. These examples illustrate that trust in the sense of benevolent behaviour yields and, at the same time, embodies a form of *social capital* [46].

While the rules employed in these mechanisms are created at design time, open systems often have to be able to define, adjust, and implement behavioural guidelines in response to environmental and internal conditions at runtime. Such an adaptability is akin to Ostrom's principle of "congruence" that states that sustainable management of commons requires to "match rules governing use of common goods to local needs and conditions" [47]. While stemming from economic and sociological research, these Ostrom's principles have been recognised as the foundations for self-organising electronic institutions as well [48]. In *normative MAS* [49], *normative institutions* enact and enforce *norms* [50] to influence the agents' behaviour indirectly. Each norm describes a behavioural rule and a sanction that is imposed if the rule is not followed. A sanction might be punitive fines or a (temporary) reduction of the violator's reputation value. The latter type of sanction treats reputation in the sense of social capital such that its reduction incentivises trustworthy behaviour in the long run. If an agent did not violate a norm on purpose, if it compensates for the violation, or if the violation was inevitable, the institution might also abstain from a sanction, which introduces a form of *forgiveness* [51, 52]. Essentially, norms have to contribute to reaching the system's goal. In eTCs (see Sect. 1.3.1), managers take on the role of normative institutions. If a manager detects an attack, it defends its community by adjusting the set of norms, e.g. by regulating the delegation and the acceptance of jobs in case of a trust breakdown – a situation in which even the reputation of benevolent agents declines [OCT12]. To enforce norms, an institution must not only be able to react with sanctions but also to detect their violation. Since monitoring an agent's behaviour comes at a price, Edenhofer et al. [OCT13] proposed to couple the effort put into surveillance to the number of received accusations. Especially when regarding trust as the basis of delegation [18], norms can also be understood as social laws governing the delegation of institutional power [53]. In this case, norms represent explicit permissions that have to be acquired before a specific action may be performed.

1.3.4 Increasing User Trust in Open Environments

Beyond the use of trust to qualify the relationships between software agents (cf. system-to-system trust in Sect. 1.1), it can also be applied to describe the social relationships between the users and the system (cf. user-to-system trust). Recent

advances in sensor technologies and context recognition enable us to capture the users' physical context continuously and to personalise information and services to them in real-time. Apart from simply providing information, context-aware systems can also allow users to manipulate or share data or even act autonomously on their behalf. Combined with advances in display and wireless technologies, users can employ these systems basically anytime and anywhere. While these so-called ubiquitous environments offer great benefits to users, they also raise a number of challenges. In particular, they might show a behaviour that negatively affects user trust. Examples include (1) highly dynamic situations where the rationale behind the system's actions is no longer apparent to the user [54], (2) implicit interactions through proxemic behaviour where the user no longer feels in control [55], or (3) privacy issues [56]. Hence, there is an enormous need for sophisticated trust management in ubiquitous environments in order to ensure that such environments will find acceptance among users.

While most work in the area of computational trust models aims to develop trust metrics that determine, on the basis of objective criteria, whether a system should be trusted or not, not much interest has been shown towards trust experienced by a user when interacting with a system. A system may be robust and secure, but nevertheless be perceived as not very trustworthy by a user, e.g. because its behaviour appears opaque or hard to control. Following the terminology by Castelfranchi and Falcone [57], a focus is put on the affective forms of trust that are based on the user's appraisal mechanisms. Therefore, the objective must be to develop a computational trust model that captures how a system – and more specifically a ubiquitous environment – is perceived by a user while interacting with it.

Many approaches found in literature aim to identify trust dimensions that influence the user's feeling of trust. This is an extension to the trust models as discussed in Sect. 1.2, even though facets of trust play a role in open self-organising systems as well [OCT14]. Trust dimensions that have been researched in the context of internet applications and e-commerce include reliability, dependability, honesty, truthfulness, security, competence, and timeliness, see, e.g. [58, 59]. Tschannen et al. [60], who are more interested in the sociological aspects of trust, introduce willing vulnerability, benevolence, reliability, competence, honesty, and openness as the constituting facets of trust, although their work does not focus on trust in software. Researchers working on adaptive user interfaces consider transparency as a major component of trust, see, e.g. [61]. Trust dimensions have formed the underlying basis of many conceptual models of trust. However, incorporating them into a computational model of trust is not a trivial task.

With the *User Trust Model* (UTM) [62], such a computational model of trust was introduced, along with a decision-theoretic approach to trust management for ubiquitous and self-adaptive environments. The UTM is based on Bayesian networks and, following ideas put forward by Yan et al. [63], assesses the users' trust in a system, monitors it over time, and applies appropriate system reactions to maintain users' trust in critical situations. In a smart office application, for example, the system could automatically switch off the lights because it senses that it is