Trustworthy Organic Computing Systems: Challenges and Perspectives

Jan-Philipp Steghöfer[1], Rolf Kiefhaber[1], Karin Leichtenstern[1],
Yvonne Bernard[2], Lukas Klejnowski[2], Wolfgang Reif[1], Theo Ungerer[1],
Elisabeth André[1], Jörg Hähner[2], and Christian Müller-Schloer[2]

[1] Insitut für Informatik
Universität Augsburg, Universitätsstrasse 6a, D-86159 Augsburg
{steghoefer,kiefhaber,leichtenstern,reif,
ungerer,andre}@informatik.uni-augsburg.de
[2] Institut für Systems Engineering, FG System- und Rechnerarchitektur
Leibniz Universität Hannover, Appelstrasse 4, D-30167 Hannover
{bernard,klejnowski,haehner,cms}@sra.uni-hannover.de

Abstract. Organic Computing (OC) systems differ from classical software systems as the topology and the participating components of the system are not predefined and therefore are subject to unforeseeable change during the systems' runtime. Thus, completely new challenges to the verification and validation of such systems as well as for interactions between system components and, of course, between the system and the user arise. These challenges can be subsumed by the terms trustworthiness or trust.

This paper proposes – after exploring the notions and principles of trust in the literature – a definition of trust which encompasses all aspects that define the trustworthiness of an Organic Computing system. It then outlines the different research challenges that have to be tackled in order to provide an understanding of trust in OC-systems and gives perspectives on how this endeavour can be taken on. Current research initiatives in the area of trust in computing systems are reviewed and discussed.

1 Introduction

Organic Computing (OC) systems [34] are highly dynamic, composed of a possibly vast number of adaptable components and are located in an ever changing environment. To cope with these circumstances, OC systems employ self-organisation mechanisms which yield a number of highly desirable properties, e.g. the ability to self-heal, to self-adapt, or to self-configure. However, classical techniques for analysis and design of software systems are no longer suitable for systems of such complex structure. Novel aspects that could not be observed in other systems, such as emergent properties, and the extreme dynamics of OC-systems require a new way to think about such systems as well as the development of new mechanisms to describe, measure and harness these properties.

An important aspect that becomes especially prominent in this kind of systems is *trust*. How can a system that changes all the time, that reacts autonomously and potentially in unforeseeable ways be trusted? One answer to this question is that a paradigm shift is needed. People will have to learn to give up some of the control they possess over a system at the moment. However, in many domains, such a shift will never fully occur. A nuclear reactor will never be controlled solely by a completely autonomous system. There will always be humans in the loop making the final decisions and able to intervene at critical points. Nonetheless, there are a lot of domains where autonomy and adaptivity can increase system performance and decrease the need for human intervention. An important prerequisite for the acceptance of OC-systems in such domains is trust. Trust is thus an enabling concept.

But trust is also important with regard to the entities that constitute such a complex system. They are expected to work together for a common aim in a highly dynamic and potentially hostile environment. As the size of autonomous systems and the number of constituting entities grows, the relationships and interdependencies between the entities become increasingly difficult to manage and maintain. The increasing complexity of modern computing systems leads to unexpected errors which can not be prevented with conventional software design methods only. In these situations, trust is a decisive factor: it has been linked with reduction of such complexity in large-scale systems [27]. As a societal mechanism it provides a framework for cooperation and a defence against malicious intruders. Again, trust acts as an enabling concept.

Furthermore, research in Organic Computing and other areas that deal with complex artificial systems (e.g. Autonomic Computing, bio-inspired self-organisation, and Distributed Artificial Intelligence, to name only a few), strives for the same end: make complex systems graspable and manageable. Trust is thus not only a concept that inevitably has to be regarded to make such systems accessible for human users, but one which – when investigated directly – contributes to the very purpose of the research in complex artificial systems.

In order to achieve these goals, different aspects of trust have to be regarded. We argue that trust is a multi-faceted concept that includes functional correctness, safety and security as well as reliability, credibility and finally usability. Among other things, it will be necessary to devise methods to describe and measure the trustworthy interaction of parts of the system, enable the observation of predefined policies at runtime, and the development of algorithms that take into account aspects of trust in self-organising systems. In particular, the user interface can no longer be implemented in a conventional fashion. Questions dealing with the transparency of self-organisation processes and adaptive representation of information on different kinds of displays have to be investigated.

In this paper, we propose a multi-faceted definition of trust and substantiate it with related definitions from the literature. There is a long tradition of research on trust and Section 2 points out the common principles that are evident in the different fields. In Section 3, we will then propose a definition suitable for self-organising systems that honours this tradition and incorporates the knowledge of the different facets. Section 4 points out the challenges associated with trust in Organic Computing systems. The paper concludes with an overview of related initiatives and a discussion of concepts proposed and existing ones in Section 5.

2 Trust and Trustworthiness in the Literature

> Trustworthiness, the capacity to commit oneself to fulfilling the legitimate expectations of others, is both the constitutive virtue of, and the key causal precondition for the existence of any society. [16]

Much of the original research on trust comes from the humanities. Psychologists and sociologists have tried for a very long time to get a grasp of the inner workings of trust in interpersonal and interorganisational relationships. Other fields, like economics and computer science, have benefitted significantly from those general findings and adapted them to the special requirements of their respective fields and the new context they are applied to.

An excellent overview of the research on trust conducted in the last 50 years in the humanities is given by Tschannen-Moran and Hoy [45]. Although the findings are applied to the American school system, they are usually general enough to be applicable to other forms of organisations and it is easy to transfer the examples involving principals, faculty and students to any hierarchically structured company or artificial system. Trust is seen as a multi-dimensional construct, deeply rooted in the interactions, behaviour and thinking of individuals and paramount to any form of cooperation within an organisation or society. It is the enabler of cooperation, a view that is shared by many researchers on the topic. It is sustained by sociologists [17] and also adapted in papers on trust in artificial systems, most prominently in multi-agent systems [38].

The multi-dimensionality of trust is a common concept. However, which dimensions contribute to trust is diverse. The differences can often be attributed to the concrete field of research the contributions stem from. The fact that the facets of trust depend on the context is however acknowledged [45]. Grandison and Sloman [19] identify reliability, dependability, honesty, truthfulness, security, competence, and timeliness for their definition of trust in the context of internet applications, Kini and Choobineh [24] in their paper on trust in e-commerce systems use "competence, dependability, and security of the system" as the relevant dimensions. The more sociologically inclined [45] define willing vulnerability, benevolence, reliability, competence, honesty, and openness as the constituting facets of trust.

An important aspect that is emphasized in much of the literature is the predictability that comes with trust. Mui et al. [33] even make this the core concept of their definition: "Trust: a subjective expectation an agent has about another's future behaviour based on the history of their encounters." Others make similar, explicit statements (e.g. the notion of *expectations* in the definition of Corritore et. al. [12]) while in some cases, the same meaning is conveyed in a more implicit fashion, by talking about *reliance* [23] or future *intentions* [40].

The definition of Mui et. al. hints at another commonality: Trust depends on experience and is subject to change over time. When two persons meet, their attitude towards each other is influenced by previous encounters in a similar context. They may have a positive or a negative default attitude towards their counterpart and may thus be more or less inclined to cooperate, a phenomenon coined *basic trust* [9] or *initial trust* [31]. This initial trust is then gradually replaced by experiential trust [30] that is accumulated from the evaluations of interactions with the new counterpart. If no interactions take place between two individuals for some time, the trust slowly deteriorates and has to be rebuilt [48]. Often, the initial trust can be supplemented by organisational measures such as reputation (see, e.g., [31,5,48,11]), where the lack of experience with a new interaction partner is replaced by the knowledge of other, more experienced members of the system.

But even if somebody has been deemed untrustworthy, there may be situations where cooperation might still be beneficial. Whenever a common goal has to be achieved, even enemies might decide to put their differences aside for a while and – although explicitly distrusting each other – cooperate to pursue a certain end with a presumed beneficial outcome for both parties. This shows how important it is to regard trust relative to a context [17]. This notion is also relevant in another regard: One trusts a doctor on a medical diagnosis but not for fixing a car [48]. Context, however, can not only be the role one finds itself in but also other environmental circumstances such as location, the time of day, or the presence of other entities [1].

Trust becomes relevant in situations that involve a risk on behalf of the trusting individual [28,40]. It is, however, debatable if individuals are more willing to trust when it is not a personal loss that can result from an interaction but an unfavourable outcome for an entire group [25]. This implies that trust is a concept that has to be regarded separately on different levels: it can be interpersonal [39], i.e. between single individuals, intraorganisational [47], i.e. between all the individuals in a group or interorganisational [44], i.e. between two distinct groups. The former has been studied intensively, e.g. by mixed-motive games such as the prisoner's dilemma (see, e.g., [13]). Within or between organisations, rules, norms, and compensation can play an important role in creating and sustaining trust and help create incentives for trustworthy behaviour [25,31].

Finally, there is a consensus that trust is a highly subjective property [14]. The relationship between trustor and trustee is unique and potentially the result of many interactions and experiences. Even if mechanisms like reputation or a normative framework are in place, each individual still has to decide on its own whether to trust an interaction partner. This fact is witnessed by the strong emphasis on relationships in the literature [45], but also made explicit in some definitions of trust [17]. Cultural differences can also heavily affect the disposition to trust or deceive [21]. As the final decision about whether or not to trust is an individual one and can be based on very different criteria, different general attitudes and different experiences, trust is also not a transitive property that is inherited or passed on. If A trusts B and C trusts A, C can still distrust B.

Of course, the definitions one finds in the literature are never all-encompassing. They usually focus on those parts of the trust concept that are relevant for the work the definition has to be applied to and leave out concepts that do not fit into the general framework of the field. As an example, the psychological treatment of trust relies heavily on the notions of moods, emotions and shared values. In an economic setting, researchers assume the "homo oeconomicus", a purely rational being, not driven by human shortcomings. Computer scientists take this to the extreme: their agents do not suffer from anything that is illogical, irrational or not based on fact.

As such, definitions from the field of Computer Science emphasise other properties of trust. Interactions of systems are still important, as are subjectivity, risk and context. But trust has to be measurable, has to be quantifiable, traceable, and be subject of calculations as well as intentional modification. Often, trust is modularized into concepts more familiar to experts of the field. Safety, reliability and performance are factors that play a role in such cases [20]. Likewise, the definition proposed in the next section is an adaptation of the principles of trust to the domain of life-like, self-organising systems.

3 A Definition of Trust in Organic Computing Systems

Trust is a multi-faceted concept that incorporates all constituting entities and users of a system and thus enables cooperation in systems of distributed entities. It allows the entities to gauge the confidence they place in their interaction partners in a given context and evolves with the experiences of the entities over time. It is comprised of the following facets:

Functional correctness: The quality of a system to adhere to its functional specification under the condition that no unexpected disturbances occur in the system's environment.

Safety: The quality of a system to be free of the possibility to enter a state or to create an output that may impose harm to its users, the system itself or parts of it, or to its environment.

Security: The absence of possibilities to defect the system in ways that disclose private information, change or delete data without authorization, or to unlawfully assume the authority to act on behalf of others in the system.

Reliability: The quality of a system to remain available even under disturbances or partial failure for a specified period of time as measured quantitatively by means of guaranteed availability, mean-time between failures, or stochastically defined performance guarantees.

Credibility: The belief in the ability and willingness of a cooperation partner to participate in an interaction in a desirable manner. Also, the ability of a system to communicate with a user consistently and transparently.

Usability: The quality of a system to provide an interface to the user that can be used efficiently, effectively and satisfactorily that in particular incorporates consideration of user control, transparency and privacy.

This definition adheres to the facets' description in the literature and is very well compatible to their intuitive, colloquial meaning. At the same time, it is precise and avoids ambiguities as best as possible. All the definitions speak about an abstract "system" to avoid focus on either hardware or software-systems and even the domain of Organic Computing. This allows explicitly to apply the definition to all aforementioned systems.

For some of the definitions, it is important to clearly define the system boundary. If, e.g., the system under consideration is a software program, then the boundary is defined in a way that other software, middleware, or hardware is excluded. In such a case, an error in the underlying hardware (like, e.g., the infamous Pentium-FDIV-Bug) can of course still manifest in the system but does not render the consideration of its functional correctness, safety, security, etc. void. This implicit reliance on the trustworthiness of the underlying system is defined as *infrastructure trust* [2].

The *functional correctness* of a system can be proved formally if a sufficiently abstract specification of the desired functionality is available [15]. Techniques such as model checking or logical deduction are used to show that the implementation adheres to the specification. This is in contrast to validation techniques such as testing and simulation that can not exhaustively show correct system behaviour. Of course, a proof of the correctness of a software system is bound to fail if there is no assumption about the correctness of the underlying platform.

A system's *safety* is compromised if it can reach a state where it causes damage (e.g., a nuclear reactor core that reaches the state "meltdown") or when the system's output causes damage (e.g., if a software that controls a robot arm commands the robot to move through the workspace in an uncontrolled fashion) [43]. The notion of system boundary is also important in this case: the control software can not be held responsible for errors in the robot's microcontroller that causes the same behaviour.

Security encompasses all aspects that have to do with the resilience of a system against attacks by a malicious party from the outside [37]. Here, again, it is important to define the system boundary and thus "outside". Typical aspects of security are authentication, authorization, and encryption. They guarantee that system users and components are who they claim they are, only authorized operations can be performed and that information is hidden from unauthorized system components or users. Security becomes especially relevant in open, dynamic systems where entities can enter and leave the system at any time [49], a condition that usually holds for self-organising systems.

The notion of *reliability* as defined above is close to the definition of the IEEE standard 610.12-1990 [22] and applies to systems consisting of several interdependent hardware or software components. Disturbances can be unexpected input values, performance-intensive computations, or failure of an external service. A system has to be robust with regards to such disturbances and maintain its functionality as long as possible. As it is an unreasonable demand to guarantee that the system will always be available, certain metrics are introduced that measure reliability and enable a practical definition of a reliable system, usually a statistical measure of the probability of failure or the time a system will probably be functional before failures can be expected [36].

Many of the aspects of the term trust as used in the literature on multi-agent systems and artificial societies are subsumed in *credibility*. Models and mechanisms are used to find malicious or selfish agents and to exclude them from interactions or to enact special measures when forced to interact with them (as, e.g., in [38]). This includes "liars", agents that claim to provide a certain quality of service and do not deliver or try to deceive other agents or the user in similar ways. Furthermore, credibility describes a property a system has with regard to its interactions with a user. The user has to be able to comprehend the actions of the system and intervene if these actions do not concur with his preferences.

In this way, credibility and usability are closely related. While the former states that the actions of the system have to be transparent for the user and still controllable, the latter deals with the way information and possible actions are presented to the user. This problem becomes accentuated if the user interface has to adapt to the display device. Large, distributed, organic systems will not be limited to a single device or even kind of device for user interaction, but will be available through standard PCs, PDAs, phones, public touchscreens, and other devices with different degrees of privacy and different modes of interactions. But still, efficiency, effectiveness and an overall satisfactory user experience as demanded by ISO 9241-11 [3] have to be guaranteed in OC-systems.

4 Challenges and Opportunities

The broad approach to trust in OC-systems outlined in the previous section brings about many interesting research questions. Some of the most pressing ones are described in the following.

4.1 Trust Models and Trust Metrics

Especially in multi-agent systems (MAS), trust models have been researched thoroughly (see, e.g., [50]). Most of them consider bilateral trust between two agents, some incorporating notions of reputation or trust values provided by the system. None of them, however, capture all the facets of our definition of trust. They are tailored to ensure agent credibility – the notion that an agent will do as it promised. Apart from this agent-to-agent point of view, trust will also have to be considered from other angles: trust between the user and a system and between the constituting agents and the system becomes important.

In many cases, Organic Computing systems will be composed of agents running on diverse hardware devices. These devices have different capabilities and differ, e.g., in the supported communications and security mechanisms available. Such circumstances will have to be incorporated into the trust metrics. They directly influence the ability of a device to contribute to the different facets such as security and reliability. The way a device behaves in an interaction can be analysed automatically and a trust value can be generated that represents the *device trust*.

In contrast to this automatic evaluation, *user trust*, the measure for the trust between two human users of a system – or the agents representing them respectively – can only be generated by the users themselves. The inherent subjectivity of this measure makes it necessary to devise methods with which users are able to enter the trust value after an interaction with another user in a consistent way and in a fashion that allows to compare trust values from different users.

Additionally, the effects of a highly dynamic system on the trust metrics need to be considered. A highly reconfigurable system changes the data basis of the trust models at runtime. The current trust models and trust metrics need to be adapted to consider this circumstance.

The newly established models and metrics will have to be incorporated in a *trust management* [19] system. This system will collect the information required, evaluate the model based on given criteria and metrics and will thus enable the evaluation of new and existing trust relationships. Especially in social settings, the trust management system will have to incorporate *reputation* [33] to allow trust relationships between agents that have not met yet.

4.2 Trusted Algorithms and Controlled Emergence

One of the ways to achieve life-like behaviour in Organic Computing systems is the deployment of self-organisation techniques. Self-organisation allows a system to automatically reconfigure and adapt to changing environmental circumstances by autonomously altering its structure. While this autonomy is inherently detrimental to trust, the problem is exacerbated by *emergent phenomena*, i.e. macroscopic behaviour that results from the microscopic interaction of entities [41]. The very nature of emergence contradicts a treatment of trust by combining the trust values of the parts of a self-organising system that contribute to the emergent phenomenon, as emergence "cannot be derived from the simple summation of properties of its constituent subsystems" [34].

Instead, there are two ways to establish trust in such systems. Firstly, algorithms have to incorporate the notion of trust directly. As trust is mainly important in the interactions of entities, the algorithms need to ensure that communication is safe and an exploitation or defection – either maliciously or unintentionally – is prevented. Secondly, mechanisms to control emergent misbehaviour are required. Emergence can have positive as well as negative effects on overall system performance and functionality. If the emergent behaviour is detrimental to a system, the term *emergent misbehaviour* is used. Such misbehaviour has to be detected and prevented or avoided completely to begin with.

Incorporating trust and mechanisms to achieve trustworthiness is a way to distributedly control emergent effects in complex systems and thus reach a system performance that is near the optimum. Self-organisation mechanisms can use trust models and metrics to form structures that adhere to the trust relationships within the system, thus reducing the need for costly communication protocols that involve a lot of mutual checks and negotiation between interaction partners. Undesired behaviour can also be avoided by observing the system and placing restrictions on the behaviour the entities in the system exhibit. If the system no longer works within the restritions, a reconfiguration can restore correct behaviour.

In the context of OC, it is especially worthwhile to investigate mechanisms which lead to better system performance based on self-organised trust and reputation systems. So-called “Trusted Communities” consist of agents able to modify their behaviour between egoistic and altruistic extremes. A social feedback process punishes or rewards agents for their behaviour, which in turn leads to behavioural adjustment. Such an adaptive feedback loop has the advantage that agents can find the optimal level of egoism/altruism according to the current situation (rather than being preprogrammed to follow a fixed algorithm). It can, e.g., make perfect sense in a sparsely populated environment to act rather selfishly, while in a densely populated area cooperation is the better choice.

Moreover, Trusted Communities allow for individual and role-based trust assignment. This way, a single agent can be part of several Trusted Communities, playing different roles with different trust levels. Being a member of a Trusted Community is an advantage since it saves the agent from checking and rechecking the trustworthiness of its partners. Since trust is also a measure for predictability of an interaction partner, it makes sense to communicate preferably with members of a known group.

4.3 Formal Treatment of Trust in Highly Dynamic Systems

Formal methods come into play in several areas of the proposed holistic approach to trust. First of all, functional correctness, safety, security, and at least in part reliability can be shown with the help of formal analysis and verification. Furthermore, the inner workings of the trust models, their evaluation and their incorporation into algorithms for trustworthy Organic Computing systems will have to be treated formally to ensure that they adhere to their requirements, produce correct results and are not exploitable.

However, existing techniques for formal analysis and verification are unfit to deal with the dynamics of OC-systems. Classical formal methods either rely on system traces, i.e. exemplary executions of the system or a model thereof or on logical deductions about the system's behaviour. The former approach works as long as the number of variables and interactions are limited. If , however, there are many interactions and system behaviour becomes complex, the number of possible traces grows exponentially (state explosion [46]). The latter approach is semi-automatic symbolic execution of the specified system, usually performed with theorem provers such as KIV [7] which is more suitable for complex systems but requires input from an expert during the proof.

While the state explosion problem has been subject to intensive research over the last couple of years (see, e.g., [10]), there is still no feasible solution for systems that exhibit the dynamics and interactivity of self-organising systems. Recent developments in interactive verification by means of symbolic execution however, provide possibilities to verify systems of many concurrent, interacting processes [6]. These techniques will have to be adapted and extended to enable the formalisation, analysis and verification of trust, its models, and its interplay in agent interactions on a sound formal foundation.

4.4 Software Engineering for Trust

Classical software engineering methodologies have been enhanced to include safety and security in recent years [29,32]. This has become very important in embedded systems for safety-critical applications that are ubiquitous in many domains such as avionics and plant-control where human life is at stake and certification requires the application of rigorous standards as well as in security-sensitive domains where systems containing valuable private information are publicly accessible or allow transactions that can result in financial damage.

However, even these extended approaches do not take into consideration all facets of the holistic trust concept. To accommodate these additional aspects, existing software engineering processes will have to be amended with supplementary artefacts and new techniques, ranging from requirements analysis to testing and deployment of trustworthy Organic Computing system. These techniques have to be strongly correlated with the formal methods that will be used to analyse and verify the system. Such a relation allows to develop the formal system model along with the actual system and provide analysis and verification already very early on in the engineering process. The feedback will allow the software engineers to create a more robust and resilient system and help gain insights into the strength and weaknesses of the proposed architecture.

While a generic system architecture for Organic Computing systems is neither feasible nor particularly useful, there are strong indications for the usefulness of reference architectures for particular system classes. The architecture of a system class generically describes the constituting elements and in many cases, also the interactions between the elements. Such encompassing specifications allow a software engineer to make formal statements not about individual applications (i.e., instances of the reference architecture) but about the system class as a whole. These formal guarantees are then inherited by the applications as long as certain parts of the interaction dynamics remain the same. This again allows to build more robust systems and adds to the trustworthiness of a system.

4.5 Trustworthy User Interfaces

It is still unclear, how a user interface (UI) for a self-organising systems has to look and adapt in different contextual situations in order to establish or improve the trust relationship between a user and the system. Should the self-organising processes be conveyed to the user? If they should, how? If not, how can the user be informed about changes in the configuration of the system? These choices not only have a direct consequence for the amount of information that has to be conveyed and the kinds of controls that will be available to the user, but also influences the way a user will interact and perceive the system.

To create and automatically manage trustworthy user interfaces, the four steps of Yan et al.'s trust management model [51] have to be addressed.

1. The consideration of the *trust establishment* between the trustor and trustee is the first step. Trust (sub-)facets (e.g. privacy and transparency) can influence the trust relationship between the user (trustor) and the user interface (trustee). Thus, a first challenge is to reveal and investigate these trust factors and their interplay for user interfaces of OC-systems. Glass et al. [18] give an interesting starting point. They investigated some trust aspects for a user interface of an adaptive system and revealed some critical trust factors, such as the usability, transparency and the granularity of the feedback.
2. While interacting with the UI, *trust monitoring* is required to control the trust relationship between the user and the user interface. Thus, we see a need to continuously measure the relevant trust factors (e.g. the transparency) as well as the user trust in different contextual situations. A second challenge therefore is to investigate and find new evaluation methods to trace the trust factors and user trust. We are interested in finding and investigating monitoring methods that objectively enable the measurement of users' trust, possibly dependent on their physiological data (e.g. skin conductance).
3. Then, *trust assessment* uses the measured values of the trust factors and the measured user trust to analyse and interpret the current trust relationship.
4. The last factor of Yan and MacLaverty's trust management model is finally to *control* and *re-establish* the trust relationship between the trustor and trustee whenever its value has changed.

Another topic worthy of consideration in this context is the heterogeneity of interaction devices used to interact with an Organic Computing system. They will not only feature different capabilities regarding presentation and input but will also be used in very different contexts. From private home computers to cell phones with a numerical keyboard to public touchscreen devices – the user interface will have to be capable of conveying the right information and making the right controls available to the user depending on context. Especially with regard to privacy, the location and the publicity of a display is very important.

5 Related Initiatives and Discussion

It has been recognized by several researchers that trust is not only a crucial aspect of modern computing systems, but also a multi-dimensional concept (see, e.g., [42]) that has to be re-evaluated for different domains. The original *Trustworthy Computing* whitepaper [35] defined three perspectives on the trustworthiness of computing systems: the user's goals, the industry's means, and the operating organisation's execution. Each was divided into different aspects, many of which (security, safety, usability, transparency, to name just a few) can be found in our definition. An important driver of the aspect description is the interlink between a system and the company that operates or provides it. This stems from the industrial origin of the initiative where the relations between company and customer are of utmost importance for the successful operation of a system. In the domain of self-organising systems however, much more fundamental questions have to be asked before this link can be established. These foundational questions also necessitate the more formal definitions of trust and its aspect, that go beyond the informal descriptions of the original whitepaper.

The TrustSOFT project[1] also uses a multi-dimensional definition of trust [20]. It resembles the above definition as it recognizes the importance of functional correctness, security, safety, reliability and so forth, but neglects usability. As the systems regarded are component systems, the user interface is not a primary concern. It is provided by a GUI that merely uses the underlying trustworthy components and can be regarded separately.

Dependable computing addresses the issue of trust from another perspective. The main goal is to avoid faults and failures. If systems depend on one another and they accept that depencence, they are said to trust each other [4]. However, trust is not something that is acquired during runtime or changes while the system's run. Our definition of trust is more akin to the definition of dependability, although dependability too is something static that is achieved by construction, not by repeated interaction.

The Trustworthy Computing initiative, as well as TrustSOFT and the Dependable Computing community do not explicitly consider systems in which the entities composing a system are very dynamic. Self-Organisation and the dynamics of an execution environment are of none or minor concern. In the domain of multiagent systems, the *Normative Multiagent Systems* (NMAS) community (for a very good introduction to the current research questions in this area see [8]) is dealing with these problems more explicitly. In NMAS, norms control the agents' behaviour as well as the behaviour of entire organisations. Norms lead to the formation of organisations in which adherence to norms is observed and non-compliance is sanctioned. This enables cooperative behaviour and forms a legal and social reality within a system that is very similar to human forms of organisations. So far, however, norms are mainly concerned with the interactions of the agents and their behaviour towards each other and do not consider aspects like functional correctness, safety, or usability.

The definitions in this paper, as well as the challenges and opportunities outlined above are not conflicting with existing initiatives. Instead of a competing proposal, we see our work as complimentary. Each of the research communities mentioned here is focusing on specific classes of systems with different fundamental assumptions and different goals. We strive to position ourselves within this environment and contribute our unique view of self-organisation, emergence and user-centric Organic Computing systems.

6 Conclusion

In this paper, we surveyed the current literature on trust and elaborated a definition of trust that is applicable to self-organising systems. In our holistic approach, trust is composed of the facets functional correctness, safety, security, reliability, credibility and usability as defined in Section 3. We then identified challenges

[1] http://www.trustsoft.uni-oldenburg.de/en/index.html

and research questions that arise from the survey and positioned the initiative within the existing research framework in trustworthy computing systems.

The AgentLink Roadmap [26], published in 2005, claims that reputation mechanisms, norms, and social structures within agent systems have to be subject of research in the medium-term future which was defined to be about five years away. Trust mechanisms for coping with malicious agents were put into the long-term future, a period that falls into the middle of the current decade. As many self-adaptive and self-organising systems use MAS as a basis, research on trust in self-organising systems will always be tightly coupled to research on trust in MAS.

The challenges discussed in the last sections fit into the framework proposed in the roadmap, both with regard to timing and research questions. While they are by far not exhaustive, they hint at the research that will have to be conducted in the next years in order to make trustworthy Organic Computing systems available to the mainstream of software engineers and software developers. By achieving these goals, multiagent systems, self-organising systems and Organic Computing systems will hopefully finally make the step out of the research laboratories to innovative software companies and finally into the real world.

Acknowledgements

This research is partly sponsored by the research unit "OC-Trust" (FOR 1085) of the German Research Foundation (DFG).

References

1. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: Proc. of the 33rd Hawaii International Conference on System Sciences, vol. 6, pp. 1–25 (2000)
2. Abrams, M.D., Joyce, M.V.: Trusted system concepts. Computers & Security 14(1), 45–56 (1995)
3. Abran, A., Khelifi, A., Suryn, W., Seffah, A.: Usability meanings and interpretations in ISO standards. Software Quality Journal 11(4), 325–338 (2003)
4. Avižienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. IEEE transactions on dependable and secure computing, 11–33 (2004)
5. Azzedin, F., Maheswaran, M.: Evolving and managing trust in grid computing systems. In: Proc. of the IEEE Canadian Conference on Electrical & Computer Engineering, pp. 1424–1429. IEEE, Los Alamitos (2002)
6. Balser, M., Reif, W.: Interactive verification of concurrent systems using symbolic execution. In: Proc. of 7th International Workshop of Implementation of Logics, IWIL 2008 (2008)
7. Balser, M., Reif, W., Schellhorn, G., Stenzel, K., Thums, A.: Formal system development with KIV. In: Maibaum, T. (ed.) FASE 2000. LNCS, vol. 1783, Springer, Heidelberg (2000)

8. Boella, G., Pigozzi, G., van der Torre, L.: Normative systems in computer science - ten guidelines for normative multiagent systems. In: Normative Multi-Agent Systems. Dagstuhl Seminar Proceedings, vol. (09121), Dagstuhl (2009)
9. Boon, S.D., Holmes, J.G.: The dynamics of interpersonal trust: Resolving uncertainty in the face of risk. Cooperation and Prosocial Behaviour, 190–211 (1991)
10. Clarke, E.M., Grumberg, O., Jha, S., Lu, Y., Veith, H.: Progress on the state explosion problem in model checking. In: Wilhelm, R. (ed.) Informatics: 10 Years Back, 10 Years Ahead. LNCS, vol. 2000, pp. 176–194. Springer, Heidelberg (2001)
11. Coleman, J.S.: Foundations of social theory. Belknap Press (1994)
12. Corritore, C.L., Kracher, B., Wiedenbeck, S.: On-line trust: concepts, evolving themes, a model. International Journal of Human-Computer Studies 58(6), 737–758 (2003)
13. Deutsch, M.: Trust and suspicion. The Journal of Conflict Resolution 2(4), 265–279 (1958)
14. Deutsch, M.: Cooperation and trust: Some theoretical notes. In: Nebraska Symposium on Motivation, vol. 10, pp. 275–319. University of Nebraska Press (1962)
15. Dunlop, D.D.: An investigation of functional correctness issues. PhD thesis, University of Maryland (1982)
16. Dunn, J.: The concept of trust in the politics of John Locke. Philosophy in History: Essays on the Historiography of Philosophy, 279–301 (1984)
17. Gambetta, D.: Can we trust trust. Trust: Making and Breaking Cooperative Relations, 213–237 (2000)
18. Glass, A., McGuinness, D.L., Wolverton, M.: Toward establishing trust in adaptive agents. In: IUI 2008: Proc. of the 13th International Conference on Intelligent User Interfaces, pp. 227–236. ACM, New York (2008)
19. Grandison, T., Sloman, M.: A survey of trust in internet applications. IEEE Communications Surveys and Tutorials 3(4), 2–16 (2000)
20. Hasselbring, W., Reussner, R.: Toward trustworthy software systems. Computer 39(4), 91 (2006)
21. Hofstede, G.J., Jonker, C.M., Verwaart, T.: A Multi-agent Model of Deceit and Trust in Intercultural Trade. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 205–216. Springer, Heidelberg (2009)
22. IEEE. IEEE Standard 610.12-1990: Glossary of Software Engineering Terminology
23. Jones, S., Morris, P.: TRUST-EC: Requirements for Trust and Confidence in E-Commerce: Report of the Workshop held in Luxembourg, April 8–9 (1999)
24. Kini, A., Choobineh, J.: Trust in electronic commerce: definition and theoretical considerations. In: Proc. of the Hawaii International Conference on System Sciences, vol. 31, pp. 51–61 (1998)
25. Kramer, R.M., Brewer, M.B., Hanna, B.A.: Collective trust and collective action. Trust in organizations: Frontiers of theory and research, 357–389 (1996)
26. Luck, M., McBurney, P., Shehory, O., Willmott, S.: Agentlink Roadmap. Agenlink.org. (2005)
27. Luhmann, N.: Trust and power. Wiley, Chichester (1979)
28. Luhmann, N.: Familiarity, confidence, trust: Problems and alternatives. Trust: Making and Breaking Cooperative Relations, 94–107 (2000)
29. Lutz, R.R.: Software engineering for safety: a roadmap. In: Proc. of the Conference on The Future of Software Engineering, pp. 213–226. ACM, New York (2000)
30. Marsh, S., Meech, J.: Trust in design. In: Proc. of the Conference on Human Factors in Computing Systems, pp. 45–46. ACM, New York (2000)

31. McKnight, D.H., Cummings, L.L., Chervany, N.L.: Initial trust formation in new organizational relationships. The Academy of Management Review 23(3), 473–490 (1998)
32. Moebius, N., Reif, W., Stenzel, K.: Modeling Security-Critical Applications with UML in the SecureMDD Approach. International Journal On Advances in Software 1, 59–79 (2009)
33. Mui, L., Mohtashemi, M., Halberstadt, A.: A computational model of trust and reputation. In: Proc. of the 35th Hawaii International Conference on System Sciences, pp. 188–196 (2002)
34. Müller-Schloer, C.: Organic computing: on the feasibility of controlled emergence. In: CODES+ISSS, pp. 2–5 (2004)
35. Mundie, C., de Vries, P., Haynes, P., Corwine, M.: Trustworthy computing. Whitepaper, Microsoft Corporation (2002)
36. Musa, J.D., Iannino, A., Okumoto, K.: Software reliability: measurement, prediction, application. McGraw-Hill, Inc., New York (1987)
37. Poslad, S., Charlton, P., Calisti, M.: Specifying standard security mechanisms in multi-agent systems. In: Falcone, R., Barber, S.K., Korba, L., Singh, M.P. (eds.) AAMAS 2002. LNCS (LNAI), vol. 2631, pp. 163–176. Springer, Heidelberg (2003)
38. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. The Knowledge Engineering Review 19(01), 1–25 (2005)
39. Rotter, J.B.: A new scale for the measurement of interpersonal trust. Journal of Personality 35(4), 651–665 (1967)
40. Rousseau, D.M., Sitkin, S.B., Burt, R.S., Camerer, C.: Not so different after all: A cross-discipline view of trust. Academy of management review 23(3), 393–404 (1998)
41. Ryan, A.J.: Emergence is coupled to scope, not level. Complexity 13(2), 67–77 (2007)
42. Schneider, F.B.: Trust in Cyberspace. National Academy Press, Washington (1998)
43. Storey, N.R.: Safety Critical Computer Systems. Addison-Wesley Longman Publishing Co., Inc., Boston (1996)
44. Sydow, J.: Understanding the constitution of interorganizational trust. Trust within and between organizations: Conceptual issues and empirical applications, 31–63 (1998)
45. Tschannen-Moran, M., Hoy, W.K.: A multidisciplinary analysis of the nature, meaning, and measurement of trust. Review of Educational Research 70(4), 547 (2000)
46. Valmari, A.: The state explosion problem. In: Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, pp. 429–528. Springer, London (1998)
47. van de Bunt, G.G., Wittek, R.P.M., de Klepper, M.C.: The evolution of intra-organizational trust networks: The case of a German paper factory: An empirical test of six trust mechanisms. International Sociology 20(3), 339 (2005)
48. Wang, Y., Vassileva, J.: Trust and Reputation Model in Peer-to-Peer Networks. In: Proc. of the 3rd International Conference on Peer-to-Peer Computing (2003)
49. Wong, H.C., Sycara, K.: Adding security and trust to multiagent systems. Applied Artificial Intelligence 14(9), 927–941 (2000)
50. Yan, Z., Holtmanns, S.: Trust modeling and management: from social trust to digital trust. Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions (2008)
51. Yan, Z., MacLaverty, R.: Autonomic trust management in a component based software system. In: Yang, L.T., Jin, H., Ma, J., Ungerer, T. (eds.) ATC 2006. LNCS, vol. 4158, pp. 279–292. Springer, Heidelberg (2006)