SELMAS 2004 Panel: Agent Dependability in Open Systems

Moderator:
Rogério de Lemos (University of Kent, UK)
Panellists:
Gul Agha (University of Illinois - USA)
Gruia-Catalin Roman (Washington University - USA)
Holger Giese (University of Paderborn - Germany)

The moderator, Rogério de Lemos, in order to promote the discussion, started the panel by providing a brief introduction to its theme. Open systems, such as the Internet, create conditions where systems can interact and collaborate with one another. Agents, which are autonomous, adaptive and interactive elements that have a mental state, can be part of these systems. However, if agent system has to be dependable in the services it provides, additional mechanisms and techniques have to be incorporated in the design of agent systems. In this context, dependability is understood as the ability of an agent or agent system to deliver its service that can justifiably be trusted. However, before starting the theme of the Panel, the moderator provided a brief overview of the previous year's Panel, which was on a related topic "agents and dependability". Three fundamental issues were raised concerning the ability of agents providing dependable services. First, agents have some features that might not be that useful for enabling dependability, for example, autonomy that impair to identify the failure assumptions to be associated with agents, and the mental state of an agent that restricts the ability for introspections. The latter is particularly fundamental for observing and controlling the state of an agent system in the presence of faults. The second issue raised was related to the role to be played by agents, should it be considered as a basic building block or an additional layer of services? The final issue mentioned was the restrictions that should be imposed on agents for guaranteeing the provision of dependable services, such as, what failure assumptions should be imposed on agent systems for them to reach a distributed consensus? Different from last year's Panel in which the emphasis was on the properties of agents, the theme of this year's Panel was on "agent communities and dependability", in which issues like, collection of agents, communication, and coordination were emphasised. Also the focus of the Panel was on "open systems" rather than architectures, which are exemplified by the Open Architecture Agent (OAA), Jade, Zeus, etc. Multi-agent communities in open systems are characterised as heterogeneous since they are created by different people with different intents at different times, using different languages, and autonomous since they have own goals, and own thread of control. It should also be considered that in terms of interaction and interoperability, agents join/leave at any time, interact with anyone, and perform any action. In addition, in order to achieve dynamic composition and coordination agents should rely on contract creation for emulating business relationships. However, there are four essential problematic issues that are associated with multi-agent communities. First, autonomy, how to use, control and manage it? Second, communication, how to ensure interoperability considering that standard protocols might restrict agent autonomy? Third, coordination, how to ensure coherent actions? Finally, the fourth issue is related to knowledge, how to enable automatic and interactive discovery of requirements and instructions? In addition to these well known problems associated with multi-agent communities there is dependability, which is usually considered as an afterthought in systems design, and which is concerned on how to ensure trust on the services delivered by the system? Taking as basis dependability technologies, which are a collection of methods and techniques by which dependability is attained, the moderator presented some existing attempts in the provision of dependability in multi-agent systems. In the context of rigorous designs, which aims to prevent the occurrence or introduction of faults, there has been some contribution mainly in the area of standards. These standards are mostly related to the semantic Web and Web standards (XML, SOAP, WSDL, UDDI, etc.), and agent standards, such as, the DAML+OIL (DARPA Agent Markup Language Activity), FIPA Agent Communication Language (ACL), Knowledge Interchange Format (KIF), Knowledge Query Manipulation Language (KQML), etc. In addition, there has been some activity in the area of formal approaches to agent-oriented software, methodologies for design and analysis of agent systems, like Gaia and Tropos, and the provision of tool support for modelling and reasoning about security in agent-oriented software engineering. In the context of fault tolerance, which is concerned with the provision of services despite the presence of faults, most of the work has focused on exception handling techniques. The problem with such approach is that application dependent techniques cannot exploit autonomy, and can restrict system coordination. In terms of general solutions for supporting fault tolerance, it is not known of any work that exploits classes of faults. In particular, it is not clear in a multi-agent environment how to reach agreement in the presence of malicious faults, using, for example, group communication algorithms. In terms of verification and validation, which aims to reduce the number and the severity of faults, and system evaluation, which aims to evaluate the presence of faults, their future incidence and consequences, so far there have not been major outcomes. Concerning verification and validation, some contributions have been made in tests for multi-agents systems, and in model checking the behavioural description of the different architectural modules of an agent. Both contributions are still in their infancy, and dependent on the application domain.

After the brief introduction by moderator, the questions to be handle by the Panel were presented. The first set of questions was related to the complexity of open systems, and the environment heterogeneity that requires multiple coordination strategies. The main question was how to obtain dependable agent communities for open systems? This question was further partition into three other questions. What are the challenges in terms of dependability technologies during design time (rigorous design, and verification & validation), and run time (fault tolerance)? What are the restrictions that have to be imposed on agents and/or architectures for open systems? How features commonly associated with agents, for example, adaptability, autonomy, learning, mobility etc., can be exploited? The second set of questions was related to fact that agents' autonomy and adaptability should have legal and social implications. The main question was who should be held responsible if an agent fails, causing the services delivered by its community to be catastrophic? This question was further partition into two other questions. What are the safeguards that a community should have to deal with entities "misdemeanours"? How a society of agents can tolerate an agent failure, or in other words, what kind of redundancies should be considered in societal terms?

The position taken for the Panel by Gul Agha was that dependability is an aggregate property that can only be approximated. Agha started his presentation by saying that dependability is a group property. Dependable individual agents do not imply dependable groups of agents, issues like competition and cooperation between agents have to be considered, and these might affect system dependability. Another issue that is important in the design of multi-agent systems is the representation of aggregate behaviours in the form of parametric models of state. For example, in order to deal with denial of service attacks, the total number of processor cycles should be considered. Another important issue is to control resource consumption, and this can be achieved through cybercash, which can be converted to resources. For example, a host provides resources to principals, which then distributed the resources between the dependent agents. In the design of dependable complex systems, designers have to pick the worst cases observed and stretch them to some limit. The system design should consider these cases, and incorporate safety envelopes. For supporting the process of adaptation for the purpose of obtaining dependable multi-agent systems it is important to learn from other technologies, such as, dynamic program modification, reflection and dynamic adaptation of the environment and evolutionary algorithms.

The position taken for the Panel by Gruia-Catalin Roman was that dependability is the desired outcome of a game played on multiple levels. He begun his talk by presenting what he understood by dependability: understand what is to be feared, compensate for what cannot be controlled, and reason about what will happen. After that, Roman gave his view on open systems operating over ad hoc networks. In terms of system structure, we should consider physical mobility of devices, logical mobility of agents, interactions constrained by logical connectivity, and the availability of resources (data, code, sensors, etc.). In terms of network profile, the essential features of wireless communication should be considered, which might cause, for example, frequent and unpredictable disconnections. In order to deal with these issues Roman proceed to present several design strategies: reduce complexity through coordination, increase productivity through transparent context maintenance, increase stability through stability enhancement, evolve and adapt through code migration. In addition to the above design strategies there are other issues that should be considered: a unified model for reasoning about logical and physical mobility, an abstract treatment of connectivity and access controls, a different way of performing query consistency and guarantees, and a spatiotemporal communication services. He concluded his presentation by stating that dependability is the desired outcome of a game played on multiple levels: the essential traits of the physical environment, the nature of devices and communication, the virtual world offered by the middleware designer, and the predictability achieved through formal analysis.

The position taken for the Panel by Holger Giese was that safety is different. He started his presentation by analysing the dependability attributes in the context of multi-agent systems (MAS). MAS can result in high reliability and availability because MAS naturally lead to heterogeneous and redundant processing of tasks. Concerning confidentiality, the design of open agent systems naturally includes the question whether agents can trust each other or not. The strong encapsulation provided by agents builds a natural basis for integrity. Finally, concerning maintainability, the flexibility and adaptability of open agent architectures as well as the strong encapsulation provided by agents simplifies maintenance to a great extend. He argued that although it is a challenging task, multi-agent systems could be made dependable according to the above attributes, however, with safety was different. Giese then enumerated the problems with safe MAS. Open MAS result in more complex or even open systems with more interactions and emergent behaviours. The adaptation and learning aspects lead to non-predictable behaviours. Since safety is a system property, safe agents do not result in a safe MAS. Moreover, MAS is only safe with respect to a given environment, hence open MAS and safety do not fit. He then proceeded to explain why safety was different in the context of MAS. MAS are inspired by the intelligent social behavior observed for humans and other natural species that have resulted from (cultural) evolution. Social system "designs" must ensure some degree of reliability, availability, confidentiality, integrity, and maintainability to enable them survive. A social system "design" only survives evolution when it protects enough members from severe and undesirable consequences - one specific design is "safe" for one specific context. The environment will only be protected by a natural social system, while destroying the environment would have severe consequences for the social system itself - safety for the environment occurs only in rare "altruistic" cases. Natural social systems tend to continue operation (being reliable or at least available) under all circumstances while safety-critical systems can behave fail-safe assuming that the operators will restart them. On the other, safety is different because safety always includes the members of the society as well as environment, and safety has usually higher demands than only enough surviving society members. In addition, in a safety-critical system catastrophic accidents have to be limited to a probability of at most 10-4 per year (SIL 4), while in natural social systems much lower probability bounds are sufficient to survive. He finished by saying that safety cannot simply be programmed into MAS using the common architectural principles! In his concluding remarks, Giese emphasised three issues: first, that open MAS architectures fit very well to all dependability attributes except safety, second, safety could only be "borrowed" from nature in rare altruistic cases and when the context/design is exactly the same (which usually is not the case!), and thirdly, the requirements for safety-critical systems are much more demanding than the ones in nature! His proposal for intelligent mechatronic agents are: separation of concerns, design for safety, and formal methods can enable us to build safe open MAS systems, if we restrict the relevant concerns in such a manner that we can ensure safety.