Autonomic computing with self-governed super-agents

Kostas Stathis
Department of Computer Science
Royal Holloway, University of London
Egham, Surrey, UK
Email: kostas.stathis@rhul.ac.uk

***Abstract*—This position paper explores the possibility of building autonomic systems using *super-agents*, virtual organizations of one or more agents that seek to govern a complex resource. The paper discusses first a control-loop suitable for building self-governed agents as an attempt to extend the original idea of autonomic managers with self-governance. It then sketches the structure of super-agents and discusses the main challenges of applying these ideas to self-governed autonomic systems.**

***Keywords*-autonomic computing, control loops, self-governed agents, virtual organizations, super-agents, agent societies.**

I. Motivation

The vision of autonomic computing is to develop computer systems capable of self-management in order to overcome their growing complexity and to reduce the barrier this complexity poses to their further growth [1], [2]. Autonomic computing refers to the self-managing characteristics of distributed computing resources, adapting to unpredictable changes whilst hiding intrinsic complexity to operators and users. Autonomic systems are envisaged to make decisions on their own using high-level policies, they will constantly analyze and optimize their status and automatically adapt themselves to the continuous changing conditions of their environments.

An autonomic system is typically understood as an interactive collection of *autonomic elements*, viz., individual system components that contain resources and deliver services to humans and other autonomic elements. Autonomic elements manage their own behaviour and their relationships with other autonomic elements in accordance with policies that humans or other elements have established. System self-management will arise at least as much from the multiple interactions among autonomic elements as it will from the internal self-management of the individual autonomic elements - just as social intelligence of an ant colony arises largely from the interaction among individual ants [1].

Fig. 1 shows IBM's reference model for an autonomic element [2], which is based on a control-loop often referred to as MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) loop. The managed element is typically a hardware resource such as storage, a printer, a directory service, or at a higher-level, an e-utility, an application service, or even an individual business. By monitoring the managed element and its external environment, and constructing and executing plans based on an analysis of this information, the autonomic manager relieves humans of the responsibility of directly managing the managed element.

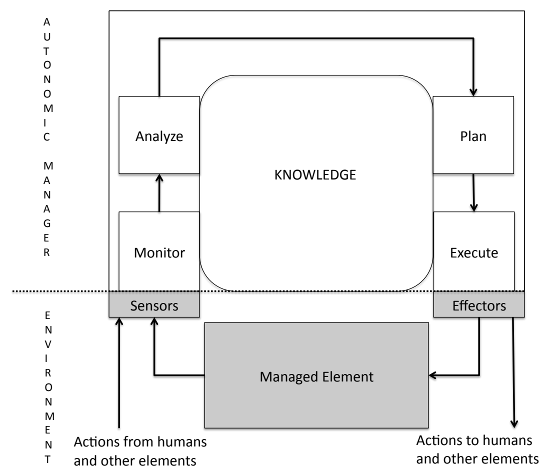

Figure 1. An autonomic element consisting of an autonomic manager and a managed element based on IBM's MAPE-K reference model ([1], [2]).

The term autonomic in IBM's vision is derived from human biology [2]. Still, according to [1] the idea of autonomic systems extends naturally to self-governed systems such as human markets, societies, and the entire world of socio-economy, mirroring computing systems which run from individual devices to the entire Internet. As a result it is being suggested that in order to develop autonomic systems, we need to seek inspiration in the self-governance of social and economic systems as well as purely biological ones. In addition, as stated in [3] MAPE-K has agent-oriented origins. Indeed, as indicated in [1], viewing autonomic elements as agents and autonomic systems as multi-agent systems will be critically important, as agents support autonomy, proactivity, goal-directed behaviour and social interactivity. However, despite the identification of self-governance as being a key to autonomic computing, most of the original work, including the MAPE-K control-loop put the emphasis on self-management, thus abstracting away from an agent-oriented perspective and self-governance.

This position paper explores an alternative to autonomic computing that uses symbolic AI agents to support self-governance as an extension of self-management. The functionality of self-governance complements self-management with the introduction of top-level goals, regulations, and authorization of plans and action selection policies. Self-governance will be also responsible for judging performance of the autonomic system and compliance of the interaction with defined policies, including revising goals and regulations to offer business value and mitigate any risks associated with it.

II. Agent-based autonomic elements

The idea of applying agents (in combination with service-oriented computing) to support autonomic systems is discussed in [4]. Applications also exist, see for example [5]. Unlike [4], we wish to expand upon the idea of how to support self-governance for an autonomic system. For this purpose we introduce an agent control-loop that generalizes MAPE-K (and similar loops for autonomic computing [6]) with functionality that extends self-management with self-governance.

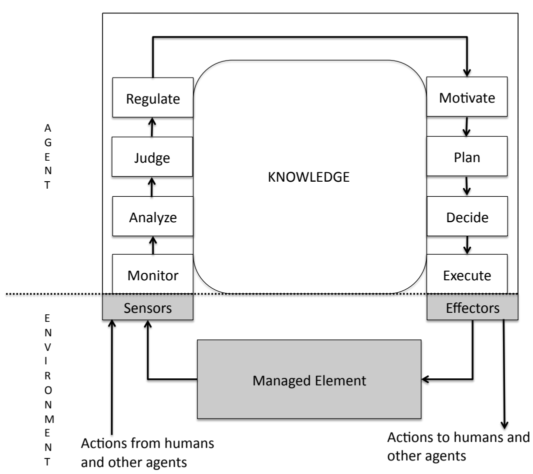

Figure 2. Agent-based autonomic element with self-governance.

As shown in Fig. 2, the agent maintains the knowledge and similar functions to that of MAPE-K, which in the spirit of [7] we will refer to as *cognitive capabilities*. When an event is sensed via the *monitor* capability, it is examined via the *analyze* capability to determine unexpected functioning in the managed element, inconsistencies in the agent's behaviour in the environment or, perhaps, violations of system norms from another agent, whether human or artificial.

The results of any analysis of monitored events are evaluated by the *judge* capability. This determines whether the consequences of the incoming event is a matter of serious policy violation on the part of the agent (or other agents in the system). In case of violation, this capability will generate new internal events that specify how violation can be avoided in the future. In some cases, this capability should also identify problems resulting not from errors in the behaviour of system components but lack of regulation in the environment in which the autonomic system is situated.

Problems with policy about the behaviour of the managed element, the agent, other agents, or insufficient regulation in the environment may result in the agent revising the corresponding rules within its control using the *regulate* capability. This capability, which is key for supporting the self-governance of the agent, will fix existing policies by perhaps extending or reorganizing the rules, and in the worst case generate new policies from scratch using techniques such as machine learning.

Revision of the rules that regulate the behaviour of the agent will make the agent to generate new goals using the *motivate* capability. This capability will take into consideration the needs and preferences of the agent before it generates new internal events that change some of the goals to intentions (goals that are ready to be planned for). Planning for the goals will be dealt by the *plan* capability so that new sub-goals and actions are generated.

The outcome of planning results in more than one action to be selected, so the agent will need to use a *decide* capability in order to choose one action to be carried out next. In doing so the agent will need to evaluate many different criteria. In many business situations these criteria will be utility-based and take into consideration the economic benefits of an action, without ignoring problems and the evaluation of risks. Once such an action has been decided, this is then passed to the *execute* capability, which will carry out the action in the agent environment.

In summary, in this section we have identified the capabilities of a control-loop for a symbolic AI agent that builds on-top of the MAPE-K reference model capabilities that will allow the agent to: (a) judge incoming events, (b) regulate its own behaviour or the behaviour of other agents, (c) motivate itself explicitly to generate new goals and form intentions, and (d) decide which actions to carry out when confronted with many planned ones. (a), (b) and (c) give the agent self-governed behaviour, while (d) makes the agent more suitable for self-management that takes into consideration economic criteria based on utilities, including the evaluation of risks. The challenges of building agents according to this new control-loop will be discussed in section IV. We discuss next autonomic systems based on multiple agents.

III. Super-agents

According to [4], the composition of autonomic elements into autonomic systems is analogous to the composition of agents into multi-agent systems. As a result, we can view an autonomic system as an interactive collection of agents that manage a complex resource such as the computer network of an organization. As in [4] we envisage that, in order to govern such a complex resource, agents will interact and communicate by forming virtual organizations (VOs) [8]. However, unlike [4], we expect that interactions in VOs to have *artificial agent societies* [9] as their breeding environment [10]. The main feature of such VOs will be agent specialization in terms of a set of well-defined roles responsible for governance, regulation, management, and reliable operation of the network resources.

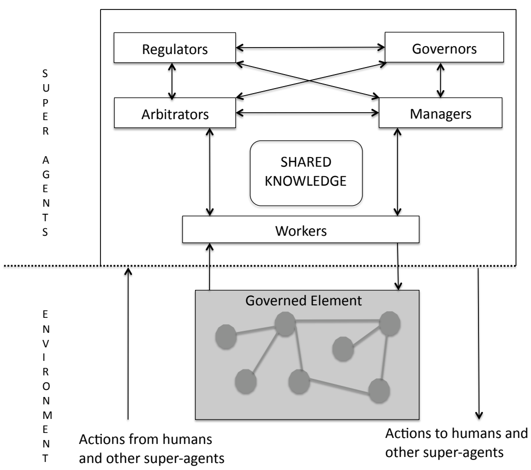

Figure 3. A virtual organization of self-governed agents structured as a super-agent.

To distinguish between the purpose of our VOs from the purpose of existing agent-based VOs [11], we coin the term *super-agents*. This notion acknowledges the duality of a VO regarding (i) the structure of the VO as an organization of interacting agents and (ii) the behaviour of the VO on the environment as a single but complex self-governed agent following the loop described in section II (Fig. 2). Fig. 3 show the main agent roles required to build a super-agent. These roles are described as follows.

- *Worker* - agents in this role will be responsible for collecting information about the resource, monitoring events of interest and executing actions on the resource on behalf of the super-agent.
- *Manager* - agents in this role are responsible for workers, they will ensure that worker policies are correctly implemented, they will analyze worker feedback and will communicate with other roles about policy implementation and progress of plans, including plan failures.
- *Arbitrator* - agents in this role are responsible for judging conflicting situations that arise either internally as a result of problems within the behaviour of the super-agent or externally in the interaction of the super-agent with the resources or humans/other super-agents.
- *Regulator* - agents in this role are responsible for revising existing policies about the governance of the resource by perhaps extending or reorganizing policy rules, and in some cases generate new policies from scratch.
- *Governor* - agents in this role will set the top-level goals according to the needs of the super-agent, the state of the resource, and the state of the environment. Governors will also communicate with managers to inform them about the intentions of the super-agent, further ensuring that these intentions are planned for and achieved by the managers without violating internal/external norms and policies.

We expect that agents assigned to different roles will still utilize the control-loop proposed in Fig 2, however, the different roles will require different levels of sophistication for different capabilities. A worker for instance will not necessarily rely on sophisticated decision making and planning but a manager instead will need to. Moreover, we anticipate that different roles will be further layered, as in human organizations. For example, we expect to have different levels of workers, managers, governors and so on, to reflect their competencies and power according to the application at hand.

IV. Challenges

Further to the issues raised in [1], [2], [6], [4], there are new challenges for building self-governed agents and super-agents for autonomic systems as follows.

Structures for agent and super-agent knowledge components. Knowledge representation frameworks and reasoning mechanisms are essential features of symbolic AI agents that are capable of goal-directed behaviour in a dynamically changing environment [4]. We anticipate that in order to build practical applications, these frameworks need to explore practical mechanisms for reflection, temporal, normative and mathematical reasoning, as well as support for complex objects and inheritance. Models and mechanisms for distributing these knowledge structures over a network and linking them to agents, as in [12], will be useful.

Self-regulation capabilities in agent models. One of the challenges with the reference model of Fig. 2 is developing computational logic models for the cognitive capabilities of the agent. Although such models exist for some of the capabilities (see for example [7]), a capability such as *regulate* will not be trivial to model as it requires techniques from meta-level reasoning, knowledge revision, and machine learning. The same capability at the level of super-agent will require from regulators to use techniques such as *organized adaptation* of the rules [13]. Proving properties of these systems will also be key.

Normative aspects of super-agents. We will need to investigate different organisations of agents and try to specify their normative aspects of protocols, workflows, contracts, and role hierarchies required by applications. How to assign responsibility in terms of power, obligations and permissions of these roles within a super-agent will be a non-trivial task. Specification models already exist, for example see the general framework of [14], [15] and the agent specific works of [16], [17], which can act as starting points.

Self-awareness for super-agents. Another important problem will be to investigate how to create and manage open super-agents that are self-aware. Self-awareness in such an open autonomic system will require cognitive models of agents to be mapped into super-agents. Advanced features such as dynamic formation of global super-agents and their federations will need to be investigated including the reasoning required for basic agents to enter in or old agents to depart from an organization, including the reasoning required to regulate exceptions in such a complex system.

Self-organising event infrastructures. Deploying and experimenting with super-agents using event processing systems and event models such as the ambient event calculus [12] of the GOLEM [18] platform will allow provide logic-based specification of interaction and communication in a large scale network. It will be important to develop these event processing systems in a way that they can self-organize according to the application needs and the (super)-agents deployed in them.

Acknowledgment

Some ideas in this paper are motivated by preliminary work for deploying symbolic AI agents to model and support service clouds for storage. The author would like to thank Julian Dean of Thinking Safe Ltd for discussions on the requirements of cloud storage applications. The work was partially supported by the London Development Agency, PARK II fund.

References

[1] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[2] IBM, "An architectural blueprint for autonomic computing," *Technical Report*, 2003.

[3] M. C. Huebscher and J. A. McCann, "A survey of autonomic computing—degrees, models, and applications," *ACM Comput. Surv.*, vol. 40, no. 3, pp. 1–28, 2008.

[4] F. M. T. Brazier, J. O. Kephart, H. V. D. Parunak, and M. N. Huhns, "Agents and service-oriented computing for autonomic computing: A research agenda," *IEEE Internet Computing*, vol. 13, no. 3, pp. 82–87, 2009.

[5] R. Das, J. O. Kephart, C. Lefurgy, G. Tesauro, D. W. Levine, and H. Chan, "Autonomic multi-agent management of power and performance in data centers," in *AAMAS (Industry Track)*, M. Berger, B. Burg, and S. Nishiyama, Eds. IFAAMAS, 2008, pp. 107–114.

[6] S. Dobson, S. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli, "A survey of autonomic communications," *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 2, pp. 223–259, 2006.

[7] A. C. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni, "Computational logic foundations of KGP agents," *J. Artif. Intell. Res. (JAIR)*, vol. 33, pp. 285–348, 2008.

[8] E. Oliveira and A. P. Rocha, "Agents advanced features for negotiation in electronic commerce and virtual organisations formation processes," in *Lecture Notes in Computer Science*. Springer-Verlag, 1999, pp. 78–97.

[9] J. Pitt, "The open agent society as a platform for the user-friendly information society," *AI Soc.*, vol. 19, no. 2, pp. 123–158, 2005.

[10] J. McGinnis, K. Stathis, and F. Toni, "A formal framework of virtual organisations as agent societies," *Electronic Proceedings in Theoretical Computer Science*, vol. 16, pp. 1–14, 2010.

[11] J. Patel, W. T. L. Teacy, N. R. Jennings, M. Luck, S. Chalmers, N. Oren, T. J. Norman, A. Preece, P. M. D. Gray, G. Shercliff, P. J. Stockreisser, J. Shao, W. A. Gray, N. J. Fiddian, and S. Thompson, "Agent-based virtual organisations for the Grid," in *Proceedings of the 1st Int. Workshop on Smart Grid Technologies*, vol. 1:4, Utrecht, Netherlands, July 2005.

[12] S. Bromuri and K. Stathis, "Distributed agent environments in the ambient event calculus," in *DEBS '09: Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*. New York, NY, USA: ACM, 2009, pp. 1–12.

[13] G. A. Vouros, A. Artikis, K. Stathis, and J. V. Pitt, Eds., *Organized Adaption in Multi-Agent Systems, First International Workshop, OAMAS 2008, Estoril, Portugal, May 13, 2008. Revised and Invited Papers*, ser. Lecture Notes in Computer Science, vol. 5368. Springer, 2009.

[14] A. Artikis, M. J. Sergot, and J. V. Pitt, "Specifying norm-governed computational societies," *ACM Trans. Comput. Log.*, vol. 10, no. 1, 2009.

[15] A. Artikis and M. J. Sergot, "Executable specification of open multi-agent systems," *Logic Journal of the IGPL*, vol. 18, no. 1, pp. 31–65, 2010.

[16] J. Broersen, M. Dastani, J. Hulstijn, Z. Huang, and L. van der Torre, "The BOID architecture: conflicts between beliefs, obligations, intentions and desires," in *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*. New York, NY, USA: ACM, 2001, pp. 9–16.

[17] F. Sadri, K. Stathis, and F. Toni, "Normative KGP agents," *Computational & Mathematical Organization Theory*, vol. 12, no. 2-3, pp. 101–126, 2006.

[18] S. Bromuri and K. Stathis, "Situating cognitive agents in GOLEM," *Engineering Environment-Mediated Multi-Agent Systems: International Workshop, EEMMAS 2007, Dresden, Germany, October 5, 2007. Selected Revised and Invited Papers*, pp. 115–134, 2008.