Available online at www.sciencedirect.com
SCIENCE DIRECT
ADVANCED ENGINEERING INFORMATICS
Advanced Engineering Informatics 19 (2005) 181–187
www.elsevier.com/locate/aei

A concise introduction to autonomic computing

Roy Sterritt[a,*], Manish Parashar[b,1], Huaglory Tianfield[c,2], Rainer Unland[d,3]

[a]School of Computing and Mathematics, Faculty of Engineering, University of Ulster at Jordanstown, Shore Road, Newtownabbey, County Antrim BT37 0QB, Northern Ireland, UK
[b]The Applied Software Systems Laboratory (TASSL), Electrical and Computer Engineering, Rutgers, The State University of New Jersey, 94 Brett Road, Piscataway, NJ 08854-8058, USA
[c]School of Computing and Mathematical Sciences, The SRIF/SHEFC Centre for Virtual Organization Technology Enabling Research (VOTER), Glasgow Caledonian University, 70 Cowcaddens Road, Glasgow G4 0BA, UK
[d]Institute for Computer Science and Business Information Systems (ICB), University of Duisburg-Essen, Schützenbahn 70, 45117 Essen, Germany

1. Introduction

The advances in computing and communication technologies and software have resulted in an explosive growth in computing systems and applications that impact all aspects of our life. However, as the scale and complexity of these systems and applications grow, their development, configuration and management challenges are beginning to break current paradigms, overwhelm the capabilities of existing tools and methodologies, and rapidly render the systems and applications brittle, unmanageable and insecure.

This has led researchers to consider alternative approaches based on strategies used by biological systems to successfully deal with similar challenges of complexity, dynamism, heterogeneity and uncertainty. Autonomic computing is emerging as a significant new strategic and holistic approach to the design of complex distributed computer systems. It is inspired by the functioning of the human nervous system and is aimed at designing and building systems that are self-managing. More specifically, an autonomic system is a self-managing, autonomous and ubiquitous computing environment that completely hides its complexity, thus providing the user with an interface that exactly meets her/his needs. The system will always decide on its own, using high-level guidance from humans, what needs to be done to keep it stable. It will constantly check and optimize its status, and automatically adapt itself to changing conditions. Self-management is achieved through key aspects such as self-governing, self-adaptation, self-organization, self-optimization, self-configuration, self-diagnosis of faults, self-protection, self-healing, self-recovery, and autonomy. Achieving these goals come down to bringing pre-emptive and proactive approaches to all areas of a computing system. Meeting these challenges of autonomic computing requires scientific and technological advances in a wide variety of fields, and new architectures that support effective integration of the constituent technologies.

2. Concepts

Biological systems have inspired systems design in many ways—Artificial Intelligence, Artificial Neural Networks, Genetic Algorithms, Genetic Programming, and Holonic Systems to name a few. The most recent is the inspiration to create self-managing systems.

2.1. Autonomic nervous system

The human body's Autonomic Nervous System (ANS) is the part of the nervous system that controls the vegetative functions of the body such as circulation of the blood, intestinal activity and secretion, and the production of chemical 'messengers', i.e. hormones, that circulate in

* Corresponding author. Tel.: +44 28 9036 8198; fax: +44 28 9036 6068.

E-mail addresses: r.sterritt@ulster.ac.uk (R. Sterritt), parashar@caip.rutgers.edu (M. Parashar), h.tianfield@gcal.ac.uk (H. Tianfield), unlandr@informatik.uni-essen.de (R. Unland).

URLs: http://www.infc.ulst.ac.uk/cgi-bin/infdb/homePage?email=r.sterritt (R. Sterritt), http://www.caip.rutgers.edu/~parashar/ (M. Parashar), http://www.gcal.ac.uk/cms/contact/staff/Hua/index.html (H. Tianfield), http://www.cs.uni-essen.de/dawis/ (R. Unland).

[1] Tel.: +1 732 445 5388; fax: +1 732 445 0593.
[2] Tel.: +44 141 331 8025; fax: +44 141 331 3608.
[3] Tel.: +49 201 183 3421; fax: +49 201 183 4460.

1474-0346/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.aei.2005.05.012

the blood [4]. The system is subdivided into the *sympathetic* (SyNS) and *parasympathetic* (PaNS) nervous systems. These separate systems tend to have opposite effects, for example, parasympathetic slows the heart rate, whereas the sympathetic speeds it up. The sympathetic nervous activity increases in response to fear, i.e. the 'fight or flight' response, while the parasympathetic nervous activity acts to calm, i.e. the 'rest and digest' response, for instance [4]. This biological self-management is influencing a new paradigm for computing that aims to create similar self-management within systems (Autonomic Computing, Autonomic Communications and Autonomic Systems). For a discussion on the ANS in relation to Ashby's Ultrastable System [55] see [2].

2.2. Autonomic computing systems

IBM introduced the Autonomic Computing initiative in 2001, with the aim of developing self-managing systems [5–7]. With the growth of the computer industry, notable examples being highly efficient networking hardware and powerful CPUs, autonomic computing is an evolution to cope with the rapidly growing complexity of integrating, managing, and operating computing system. Computing systems should be effective [8], they should serve a useful purpose when they are first launched and continue to be useful as conditions change. The realization of autonomic computing will result in a significant improvement in system management efficiency. The disparate technologies that manage the environment work together to deliver the best performance results [9].

As has been mentioned, the Autonomic Computing initiative is inspired by the human body's autonomic nervous system [9]. The autonomic nervous system monitors the heartbeat, checks blood sugar levels and maintains normal body temperature without any conscious effort from the human. There is an important distinction between autonomic activity in the human body and autonomic responses in computer systems. Many of the decisions made by autonomic elements in the body are involuntary, whereas autonomic elements in computer systems make decisions based on tasks which are chosen to be delegated to the technology [9].

Upon launching the Autonomic Computing initiative, IBM defined four general properties a system should have to constitute self-management: *self-configuring*, *self-healing*, *self-optimising* and *self-protecting*. These are accompanied by four enabling properties or attributes, namely *self-awareness*, *environment-awareness*, *self-monitoring* and *self-adjusting* [5–7]. Essentially, these objectives represent broad system requirements, while the attributes identify basic implementation mechanisms [10]. Since the 2001 launch of Autonomic Computing, the *self*-* list of properties has grown substantially [11,12]. It now also includes features such as *self-anticipating*, *self-adapting*, *self-critical*, *self-defining*, *self-destructing*, *self-diagnosis*, *self-governing*, *self-organized*, *self-recovery*, *self-reflecting*, and *self-simulation* [3,11]. Moreover, previous properties such as *self-stabilizing* were re-discovered [46,47]. Yet the initial set still represents the general goal.

To meet this autonomic *selfware* vision, systems should be designed with components that are allocated an autonomic manager.

Fig. 1 represents an Autonomic Element (AE) which consists of a managed component (MC) and an autonomic manager (AM) [3,2,7]. Control loops with sensors (self-monitor) and effectors (self-adjuster) together with system knowledge and planning/adapting policies allow the autonomic element to be self-aware and to self-manage—this is depicted as the local (L) control loop in Fig. 1. A similar scheme, depicted as the global (G) control loop in Fig. 1, facilitates environment awareness, allowing self-management if necessary, but without the immediate control to change the environment—this is effected through communication with other autonomic managers that have the relevant influence through reflex or event messages as well as mobile agents [8]. In this scheme, every component in a system, and every system within systems of systems, is self-managing with management communications between autonomic managers.

The influence of the autonomic nervous system (ANS) may imply that the Autonomic Computing initiative is concerned only with low level self-managing capabilities such as reflex reactions. Yet the vision behind the initiative has as its overarching goal system-wide policy-based self-management, where a human manager will state a business-critical success factor and the Information and Communications Technologies (ICT) systems will take care of realizing it, self-configuring and self-optimising to meet the policies, and self-protecting and self-healing to ensure the policies are maintained in light of changing context. It may be reasoned that due to our ANS we are freed (non-conscious activity) from the low-level complexity of managing our bodies, to perform high-level complex tasks. Similarly, for computing to develop further and provide equivalent high-level system-wide tasks, a corresponding low-level 'non-conscious' architecture is

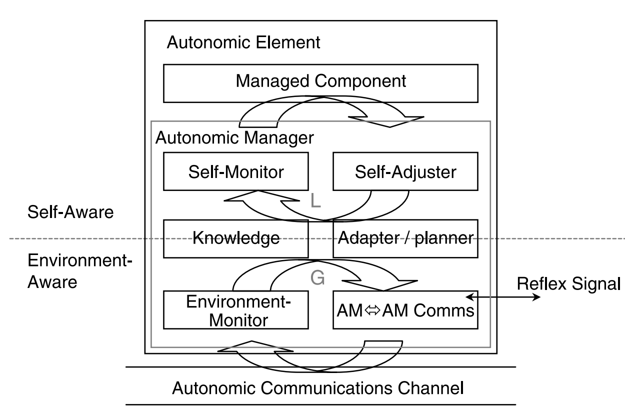

Fig. 1. Autonomic element: managed component plus autonomic manager.

necessary. As such, growing this initiative will converge and cross-influence other emerging fields such as ubiquitous and pervasive computation and communications.

3. Autonomic computing

3.1. Innovative self-managing components and interaction

The Autonomic Environment requires that autonomic elements and in particular autonomic managers communicate with one another concerning self-* activities. An additional component within the autonomic element is the pulse monitor, PBM (an extension of the embedded systems heart-beat monitor (HBM) that safe guards vital processes by emitting a regular 'I am alive' signal to another process), with the capability to encode health and urgency signals as a pulse [13], i.e. an 'I am healthy' signal. Together with standard event messages and mobile agents on the autonomic communications channel, this provides dynamics within autonomic responses and multiple loops of control, such as reflex reactions among the autonomic managers [14].

3.2. AI and autonomic components

The scope of *AI and autonomic system* is as wide as the AI research field itself [34]. Of particular interest from AI and other generic technologies for self-managing, autonomic and/or cooperative components are soft computing techniques (neural networks, fuzzy logic, probabilistic reasoning incorporating Bayesian networks and so on), machine learning techniques, cybernetics, optimization techniques, fault diagnosis techniques, feedback control, and planning techniques. Some examples of areas of research are discussed below.

Clockwork, a method for providing predictive self-management, regulates behaviour in anticipation of need using statistical modelling, probabilistic reasoning, tracking and forecasting methods [19]. It includes real-time model selection techniques to fulfil the self-configuration element of Autonomic Computing [20].

The use of probabilistic techniques, such as Bayesian networks (BNs) discussed in [21], are also central to research in autonomic algorithm selection. Systems use the BN approach along with self-training and self-optimization to find the best algorithm [21].

In terms of root cause analysis within Autonomic Communications and Autonomic Networks, it has been highlighted that correlation, rule discovery and root cause analysis activity can benefit from incorporating Bayesian Networks [27,28], either in the rule discovery process or in the actual model learning to assist with self-healing [29].

Large-scale server management and control has also received similar treatment. Event logs from a 250 node large-scale server were analyzed by applying a number of machine learning algorithms and AI techniques to establish time-series methods, rule-based classification and Bayesian network algorithms for a self-management and control system [30].

Another current aspect is the calculation of costs in an autonomic system and the self-healing equation. One approach utilizes naive Bayes for cost-sensitive classification and a feedback approach based on a Markov decision process for failure remediation [31]. The argument is easily made that the autonomic system involves decisions and decisions involve costs [32]. This naturally leads to work with agents, incentives, costs and competition for resource allocation and extensions thereof [32,33].

In effect, the breadth and scope of the autonomic vision is highlighted by such work that uses AI techniques (machine learning, Tabu search, statistical reasoning and clustering analysis) for controlling the detection of the need for re-optimization of enterprise business objectives [22].

An interesting paper, [15], discusses affect and machine design [16]. Essentially it supports those psychologists and AI researchers that hold the view that affect (and emotion) is essential for intelligent behaviour [17,18]. It proposes three levels for the design of systems: reaction, routine and reflection. Reaction is the lowest level where no learning occurs and only involves immediate response to state information coming from sensory systems. Routine is the middle level where largely routine evaluation and planning behaviours take place. It receives input from sensors as well as from the reaction and reflection level. Assessment at this level results in three dimensions of affect and emotion values: positive affect, negative affect and (energetic) arousal. Reflection is the top level, which receives no sensory input or has no motor output; it only receives input from below. Reflection is a meta-process, where the mind deliberates about itself. Essentially operations at this level look at the systems representations of its experiences, its current behaviour, its current environment, etc.

Essentially the reaction level sits within the engineering domain, monitoring current state of both the machine and its environment and rapidly reacting to changing circumstances. The reflection level may reside within the AI domain, utilizing its techniques to consider the behaviour of the system and learn new strategies. The routine level may be a cooperative mixture of both. Interestingly, this approach may be considered as a generic design for an autonomic architecture.

3.3. Autonomic architectures

Research in Autonomic Architectures consists of general architectures for individual components or complete autonomic computing systems, based on the integration of advanced technologies like Open Grid Computing, Web Services, (Multi-) Agent technologies, and/or intelligent/autonomous robotics.

The brief discussion in Section 2 concerning Fig. 1 highlights the general autonomic element architecture. This is similar to the IBM view [5–7,9] that represents the closed control loop within the autonomic element as consisting of four stages: 'MAPE'—monitor, analyze, plan and execute. The monitor and analyse parts of the structure process information from sensors to provide both self-awareness and an awareness of the external environment. The plan and execute parts decide on the necessary self-management behaviour that will be executed through the effectors. The MAPE components use correlations, rules, beliefs, expectations, histories and other information known to the autonomic element, or available to it through the knowledge repository within the autonomic manager (AM). White et al. [56] are extending this architecture.

3.4. Autonomic interaction and policy based self-management

Autonomic Interaction and Policy Based Self-Management are technologies that support inter-element interactions, such as service-level agreements, negotiation protocols and algorithms, and conversation support. Research efforts based on the approach are discussed below.

The ABLE rule engine can be used for interaction and negotiation. In effect, it is an agent building learning environment that includes time series analysis and Bayes classification among others. It correlates events and invokes the necessary action policy [26].

Policy based management becomes particularly important with the future vision of Autonomic Computing, to facilitate high level specification of the goals and aims for the system to achieve [45]. A policy-based management tool may reduce the complexity of product and system management by providing a uniform cross-product policy definition and management infrastructure [26].

3.5. Computer–human interaction and Autonomic Systems

Computer–Human interaction with autonomic systems for engineering environments comprises user studies, interfaces for monitoring and controlling behaviour, and techniques for defining, distributing, and understanding policies. The issues involved in the area are highlighted in the paper 'Dealing with ghosts' [58] and the concerns are being discussed through workshops [59].

Personal Autonomic Computing has a particular emphasis on CHI and Autonomicity since if you consider; autonomic computing makes choices for you while personal computing allows you to make choices yourself as such with PAC the human is still very much in the loop [68,69].

3.6. Science of autonomicity

Science of Autonomicity is the fundamental science of self-managing systems for engineering environments: understanding, controlling, or exploiting emergent behaviour, theoretical investigations of coupled feedback loops, robustness, and other related topics.

Autonomic has been expressed as the automation of systems adaptation [1]. Such work in the science of autonomicity is required for the area to mature and be used as a standard technique within engineering.

3.7. Systems and software engineering for Autonomic Systems

There is a strong need for system and software engineering process models/methodologies for building autonomic systems. As expected, the focus during the early days of the new paradigm was on implementations/prototypes, architectures and tools to prove the concept. Increasingly, we are now seeing the 'Salmon effect', moving back up the waterfall from implementations to specific interest in how to program autonomic systems, how to design for self-management, and how to gather requirements specifically for an autonomic environment. Techniques are starting to be considered for requirements analysis and system design of autonomic systems, for instance Checkland's Soft Systems Methodology and Beer's Viable Systems Model [51], and relevance has been found in architectural approaches for describing an adaptive system [48–50]. Shaw et al.'s efforts on a process control approach to programming is considered particularly relevant to the AC initiative—not surprising since it was influenced by control loops [52,53].

In terms of engineering of self-management into legacy systems, it involves providing an environment that monitors the sensors of the system and provides adjustment through effectors to create a control loop. One such systems engineering infrastructure is KX (Kinesthetics eXtreme), which runs a lightweight decentralized collection of active middleware components tied together via a publish-subscribe (content-based messaging) event system [35]. Similarly, the Astrolabe tool may be used to automate self-configuration and monitoring, and to control adaptation [36].

The AutoMate project, incorporating ACCORD, an autonomic component framework utilizes DIOS to provide mechanisms to directly enhance traditional computational objects/components with sensors, actuators, rules, a control network, management of distributed sensors and actuators, interrogation, monitoring and manipulation of components at runtime through a distributed rule-engine [37].

4. Examples of autonomic systems and applications

There have been a number of research efforts in both academia and industry to develop autonomic systems and applications. A sampling of these research systems are presented below.

Unity is an autonomic system that implements IBM's view of an autonomic architecture [57].

OceanStore [61,62], which is a global, consistent, highly-available persistent data storage system that supports self-healing, self-optimization, self-configuration, self-protection, policy based caching, routing substrate adaptation, autonomic replication, continuous monitoring, testing, and repair.

Storage Tank [63] is a multi-platform, universally accessible storage management system. It supports self-optimization, self-healing, policy based storage and data management, server redirection and log-based recovery.

Oceano [64] facilitates cost effective scalable management of computing resources for software farms. In terms of autonomic behaviour, it handles self-optimization, self-awareness, autonomic demand distribution, and constant component monitoring.

AutoAdmin [65] sets out to reduce Total Cost of Ownership (TCO) through self-tuning, self-administration by usage tracking, index tuning and recommendations based on workload.

Sabio [66] autonomically classifies large numbers of documents demonstrating self-* properties such as self-organization and self-awareness. It groups documents according to word and phrase usage.

Q-Fabric [67] provides system support for continuous online management through self-organization. It features continuous online quality management through 'customizability' of each application's Quality of Service (QoS).

Applications of autonomic computing in engineering applications, such as autonomic (urban) traffic systems, autonomic industrial/residential building systems, autonomic industrial process systems, or autonomic manufacturing systems will increasingly come to the fore. As would be expected, early adaptors have come from within computing, for instance, efforts to add autonomic capabilities to instant messaging, spam detection, load balancing and middleware have been reported [35].

Database systems in particular, have been an early success within the AC initiative [54] due to the evolution of the DBMS towards more complex features and a resulting move towards self-tuning. SMART DB2 [60] provides for the reduction of human intervention and cost for DB2 through such self-management systems as self-optimization, self-configuration, autonomic index determination, disaster recovery, continuous monitoring of DB2's health and alerting the DBA.

An example of a ubiquitous computing application, Smart Doorplates, seeks to assist visitors to a building in locating an individual who is presently not in his/her office. A module in the architecture utilizes probabilistic reasoning to predict the next location of the individual, which is reported along with their current location [23,24].

Early versions of tools or autonomic functionality updates to existing tools and software suites in this area have recently been released by IBM [26] through their AlphaWorks Autonomic Zone website [25]. The generic Log and Trace Tool correlates event logs from legacy systems to identify patterns. These can be used to facilitate automation or help debugging [26]. The Tivoli Autonomic Monitoring Engine essentially provides server level correlation of multiple IT systems to assist with root cause analysis and automated corrective action [26].

Increasing constraints on resources, and greater focus on the cost of operations, has led NASA and others to utilize adaptive operations and move towards almost total onboard autonomy in certain classes of mission operations [38,39]. Autonomy provides self-governance, giving responsibility to the agents within the system to meet their defined goals. Autonomic Systems provides self-management, in addition to self-governance, to meets one's own functional goals. There is also a shared responsibility to ensure the effective management (through self-* properties) of the system, which may include responsibilities beyond the normal task oriented goals of an individual agent, for instance, to monitor another agents health signs (ensuring self-protection, self-healing and possibly initiate self-configuration and/or self-optimization activities). As such, Autonomic Computing has been identified by NASA as a key area [40–43] and research is underway to utilize it in addition to autonomy [44].

Acknowledgements

The guest editors are supported in part by various bodies in their autonomic research, these include:

M. Parashar: National Science Foundation via grants numbers ACI 9984357, EIA 0103674, EIA 0120934, ANI 0335244, CNS 0305495, CNS 0426354 and IIS 0430826.

R. Sterritt: IEEE TC-ECBS, UU's Computer Science Research Institute and the Centre for Software Process Technologies (CSPT), funded by Invest NI through the Centres of Excellence Programme, under the EU Peace II initiative.

We would like to thank the Editor-in-Chief Professor Dr Ian Smith for his interest in and support of this special issue. We would also like to thank the authors for their high-quality submissions, their cooperation and patience. A special thank also goes to the technical reviewers for their excellent and supportive reviews. All of these efforts have significantly contributed to the technical quality of the works included in this special issue.

References

[1] Tianfield H, Unland R. Towards autonomic computing systems. Eng Appl Artif Intell 2004;17(7):669–99. Elsevier, ISSN 0952-1976.

[2] Parashar M, Hariri S. Autonomic computing: an overview, upp 2004. J.-P. Banâtre (ed). LNCS, Springer Verlag, 2005; 3566: 247–259.

[3] Sterritt R. Autonomic computing. Innovat Syst Softw Eng 2005;1(1). DOI: 101007/s11334-101005-0001-5.
[4] Oxford reference encyclopaedia. Oxford: Oxford University Press; 1998.
[5] Horn P. Autonomic computing: IBM perspective on the state of information technology, IBM T.J. Watson Labs, NY, 15th October 2001. Presented at AGENDA 2001, Scottsdale, AR (available at http://www.research.ibm.com/autonomic/); 2001.
[6] Kephart J, Chess D. The vision of autonomic computing. IEEE Comput 2003;36:41–50.
[7] Ganek AG, Corbi TA. The dawning of the autonomic computing era. IBM Syst J 2003;42(1).
[8] Sterritt R, Bustard DW. Towards an autonomic computing environment. In: Proceedings of IEEE DEXA 2003 Workshops—First International Workshop on Autonomic Computing Systems, Prague, Czech Republic; September 1–5, 2003, IEEE Computer Society Press, p. 694–8.
[9] IBM White Paper. An architectural blueprint for autonomic computing. IBM; 2003.
[10] Sterritt R, Bustard D. Autonomic computing—a means of achieving dependability? Proceedings of IEEE international conference on the engineering of computer based system (ECBS'03), Huntsville, Alabama, USA; April 7–11 2003.
[11] Tianfield H. Multi-agent based autonomic architecture for network management Industrial Informatics, 2003, INDIN 2003. Proceedings IEEE international conference on, 21–24; Aug 2003. p. 462–9.
[12] Tianfield H. Multi-agent autonomic architecture and its application in e-medicine. Proceedings of the 2003 IEEE/WIC international conference on intelligent agent technology (IAT'2003), Halifax, Canada; 13–16 October 2003. p. 601–4.
[13] Sterritt R. Pulse monitoring: extending the health-check for the autonomic GRID. Proceedings of IEEE workshop on autonomic computing principles and architectures (AUCOPA) at INDIN 2003, Banff, Alberta, Canada, 22–23 August 2003 p. 433–40.
[14] Sterritt R, Gunning D, Meban A, Henning P. Exploring autonomic options in an unified fault management architecture through reflex reactions via pulse monitoring. Proceedings of IEEE workshop on the engineering of autonomic systems (EASe) at the 11th annual IEEE international conference and workshop on the engineering of computer based systems (ECBS 2004), Brno, Czech Republic; 24–27 May 2004. p. 449–55.
[15] IBM Systems Journal. Special issue on autonomic computing 2003; 42(1):197 available at http://www.research.ibm.com/journal/sj/421/.
[16] Norman DA, Ortony A, Russell DM. Affect and machine design: lessons for the development of autonomous machines. IBM Syst J 2003;42(1):38–44.
[17] Sloman A, Croucher M. Why Robots will have emotions Proceedings seventh international joint conference on AI, Vancouver 1981 p. 197–202.
[18] Sloman A. Review of: Rosalind Picard's affective computing AI magazine. MIT Press; 1997.
[19] Russell LW, Morgan SP, Chron EG. Clockwork: a new movement in autonomic systems. IBM Syst J 2003;42(1):77–84.
[20] Russell LW, Morgan SP, Chron EG. On-line model selection procedures in clockwork IJCAI workshop on AI and autonomic computing: developing a research agenda for self-managing computer systems, Acapulco, Mexico August 10 2003.
[21] Guo H. A Bayesian approach for autonomic algorithm selection IJCAI workshop on AI and autonomic computing: developing a research agenda for self-managing computer systems, Acapulco, Mexico, August; 10 2003.
[22] Aiber S, Etzion O, Wasserkrug S. The utilization of AI techniques in the autonomic monitoring and optimization of business objectives IJCAI workshop on AI and autonomic computing: developing a research agenda for self-managing computer systems, Acapulco, Mexico; August 10 2003.
[23] Trumler W, Bagci F, Petzold J, Ungerer T. Smart doorplates—toward an autonomic computing system Autonomic computing workshop fifth annual international workshop on active middleware services (AMS'03); June 2003. p. 42.
[24] Trumler W, Petzold J, Bagci F, Ungerer T. AMUN—autonomic middleware for ubiquitous eNvironments applied to the smart doorplate project International conference on autonomic computing (ICAC'04); May 2004. p. 274–5.
[25] IBM, AlphaWorks Autonomic Computing site, http://www.alphaworks.ibm.com/autonomic.
[26] Ganek AG. Autonomic computing: implementing the vision Keynote presentation at the autonomic computing workshop, (AMS 2003), Seattle, WA, 25th June 2003.
[27] Sterritt R, Bustard DW, McCrea A. Autonomic computing correlation for fault management system evolution. Proceedings of IEEE international conference industrial informatics (INDIN 2003) Banff, Alberta, Canada; 21–24 August 2003. p. 240–7.
[28] Sterritt R, Bustard DW. Fusing hard and soft computing for fault management in telecommunications systems. IEEE Trans Syst Man Cybern part C, 2002; 32(2).
[29] Sterritt R. Autonomic networks: engineering the self-healing property. Eng Appl Artif Intell 2004;17(7):727–73. Elsevier, ISSN 0952-1976.
[30] Sahoo RK, Rish I, Oliner AJ, Gupta M, Moreira JE, Ma S, et al. Autonomic computing features for large-scale server management and control IJCAI workshop on AI and autonomic computing: developing a research agenda for self-managing computer systems, Acapulco, Mexico; August 10 2003.
[31] Littman ML, Nguyen T, Hirsh H. A model of cost-sensitive fault mediation IJCAI workshop on AI and autonomic computing: developing a research agenda for self-managing computer systems, Acapulco, Mexico; August 10 2003.
[32] Cheliotis G, Kenyon C. Autonomic economics: a blueprint for self-managed systems IJCAI workshop on AI and autonomic computing: developing a research agenda for self-managing computer systems, Acapulco, Mexico; August 10 2003.
[33] Parkes DC. Five AI challenges in strategy proof computing IJCAI workshop on AI and autonomic computing: developing a research agenda for self-managing computer systems, Acapulco, Mexico; August 10 2003.
[34] Sterritt R. Autonomic computing: the natural fusion of soft computing and hard computing Proceedings of 2003 IEEE international conference on systems, man and cybernetics, Washington, DC, USA; October 5–8 2003. p. 4754–9.
[35] Kaiser G, Parekh J, Gross P, Valetto G. Kinesthetics eXtreme: an external infrastructure for monitoring distributed legacy systems Proceedings of the autonomic computing workshop, fifth international workshop on active middleware services (AMS 2003), Seattle, WA; 2003. p. 22–30.
[36] Birman KP, van Renesse R, Vogels W. Navigating in the storm: using astrolabe for distributed self-configuration, monitoring and adaptation. Proceedings of the autonomic computing workshop, fifth international workshop on active middleware services (AMS 2003), Seattle, WA; 2003. p. 4–13.
[37] Agarwal M, Bhat V, Liu H, Matossian V, Putty V, Schmidt C, et al. AutoMate: enabling autonomic applications on the grid. Proceedings of the autonomic computing workshop, fifth international workshop on active middleware services (AMS 2003), Seattle, WA; 2003. p. 22–30.
[38] Wyatt J, Sherwood R, Sue M, Szijjarto J. Flight validation of on-demand operations: the deep space one beacon monitor operations experiment Fifth international symposium on artificial intelligence, robotics and automation in space (i-SAIRAS '99), ESTEC, Noordwijk, The Netherlands, 1–3 June 1999.
[39] Truszkowski W, Hinchey M, Rash J, Rouff C. NASA's swarm missions: the challenge of building autonomous software. IEEE IT Professional Mag 2004;September/October:51–6.

[40] Clancy DJ. NASA challenges in autonomic computing, Almaden Institute 2002, San Jose, CA: IBM Almaden Research Center; April 10, 2002.

[41] Sterritt R. Towards autonomic computing: effective event management. Proceedings of 27th annual IEEE/NASA software engineering workshop (SEW), Maryland, USA, December 3–5.: IEEE Computer Society; 2002. p. 40–7.

[42] Hughes PM. Application of autonomic computing concepts for GSFC's next generation missions, Presentation at the woodrow wilson international centre for scholars; October 28 2003.

[43] Sterritt R. Selfwares-episode IV—autonomic computing: a new hope. NASA GSFC IS&T Colloquium; 2004.

[44] Truszkowski W, Rash J, Rouff C, Hinchey M. Asteroid exploration with autonomic systems. Proceedings of IEEE workshop on the engineering of autonomic systems (EASe) at the 11th annual IEEE international conference and workshop on the engineering of computer based systems (ECBS 2004), Brno, Czech Republic; 24–27 May 2004. p. 484–90.

[45] Lymberopoulos L, Lupu E, Sloman M. An adaptive policy-based framework for network services management. J Netw Syst Manage 2003;11(3).

[46] Dijkstra EW. Self-stabilizing systems in spite of distributed control. Commun ACM 1974;17(11):643–4.

[47] Schneider M. Self-stabilization. ACM Comput Surveys (CSUR)m 1993;25(1):45–67.

[48] Garlan D, Chung S, Schmerl B. Increasing system dependability through architecture based self-repair Architecting dependable systems, LCS 2677. Berlin: Springer; 2003.

[49] Georgiadis I, Magee J, Kramer J. Self-organising software architectures for distributed systems. Proceedings of the first ACM SIGSOFT workshop on self-healing systems (WOOS); 2002. p. 33–8.

[50] Oriezy P, Gorlick MM, Taylor RN, Heimbigner D, Johnson G, Medvidovic N, et al. An architecture-based approach for self-adaptive software. IEEE Intell Syst 1999;14(3):52–62.

[51] Bustard DW, Sterritt R, Taleb-Bendiab A, Laws A, Randles M, Keenan F. Towards a systemic approach to autonomic systems engineering. Proceedings of IEEE workshop on the engineering of autonomic systems (EASe 2005) at 12th annual IEEE international conference and workshop on the engineering of computer based systems (ECBS 2005), Greenbelt, MD, USA; 3–8 April 2005. p. 465–72.

[52] Shaw M. Beyond objects: a software design paradigm based on process control. ACM Softw Eng Notes 1995;20(11):27–38.

[53] Shaw M. Self-healing': softening precision of avoid brittleness. Proceedings of the first ACM SIGSOFT workshop on self-healing systems (WOSS) 2002 p. 111–3.

[54] Diao Y, Eskesen F, Froehlich S, Hellerstein JL, Spainhower LF, Surendra M. Generic online optimization of multiple configuration parameters with application to a database server. Proceedings of the 14th IFIP/IEEE workshop on distributed systems: operations and management (DSOM), LNCS 2867. Berlin: Springer; 2003. p. 3–15.

[55] Ashby WR. Design for a brain.: Chapman and Hall Ltd; 1960.

[56] White S, Hanson J, Whalley I, Chess D, Kephart J. An architectural approach to autonomic computing. Proceedings of IEEE first international conference on autonomic computing, New York May 2004. p. 2–9.

[57] Chess D, Segal A, Whalley I, White S. Unity: experiences with a prototype autonomic computing system. Proceedings of IEEE first international conference on autonomic computing, New York May 2004. p. 140–7.

[58] Russell DM, Maglio PP, Dordick R, Neti C. Dealing with ghosts: managing the user experience of autonomic computing. IBM Syst J 2003;42(1).

[59] Conference on the Human Impact and Application of Autonomic Computing Systems (CHIACS) http://www.almaden.ibm.com/asr/chiacs/.

[60] Lohman GM, Lightstone S. SMART: making DB2 (More) autonomic. In: VLDB 28th international conference on very large data bases, Kowloon Shangri-La Hotel, Hong Kong, China; August 20–23 2002.

[61] Berkeley UC. Computer science division. Hildrum K. 'The OceanStore Project', project overview. http://oceanstore.cs.berkeley.edu/info/overview.html, July 8 2002. Project Page.

[62] Kubiatowicz J. 'OceanStore: global-scale persistent storage' Stanford Seminar Series, Stanford University, http://oceanstore.cs.berkeley.edu/publications/talks/StanfordOceanStore.pdf; Spring 2001.

[63] Menon J, Pease DA, Rees R, Duyanovich L, Hillsberg B. IBM storage tank-A heterogeneous scalable SAN file system. IBM Syst J 2003; 42(2):250–67.

[64] IBM Research. The oceano project. http://www.research.ibm.com/oceanoproject/. IBM Corp.

[65] Narasayya V. AutoAdmin: towards self-tuning databases; November 13 2002. Guest Lecture at Stanford University.

[66] Pool R. A new computer program classifies documents automatically 2002 http://domino.watson.ibm.com/comm/wwwr_thinkresearch.nsf/_pages/selection200.html.

[67] Poellabauer C. Q-fabric—system support for continuous online quality management; 2002 http://www.cc.gatech.edu/systems/projects/ELinux/qfabric.html.

[68] Bantz DF, Bisdikian C, Challener D, Karidis JP, Mastrianni S, Mohindra A, et al. Autonomic personal computing. IBM Syst J 2003; 42(1):165–76.

[69] Sterritt R, Bantz DF. PAC-MEN: personal autonomic computing monitoring environments. Proceedings of IEEE DEXA 2004 workshops—second international workshop on self-adaptive and autonomic computing systems (SAACS 04), Zaragoza, Spain, August 30th–September 3rd, IEEE; 2004. p. 737–41.