Autonomic Analytics

Roy Sterritt
University of Ulster
School of Computing and Mathematics
Jordanstown Campus, Northern Ireland
r.sterritt@ulster.ac.uk – @RoysterUlster

***Abstract*—As the size and complexity of cyber-physical systems continue to grow, there is a heightened need to develop new analytical techniques capable of achieving a level of service with successful operations upon which users can place even more reliance. This paper presents an emerging strategy for meeting this demand 'Autonomic Analytics', utilizing the autonomic computing paradigm to deliver real-time, self-managing, context and situation aware analytics. A three-tier rule-discovery framework and associated support and analysis tools are described. These assist with the development, management and maintenance of analytical rules and beliefs to allow for the progressive development evolution from 'human-in-the-loop' to 'human-on-the-loop' towards the long-term (and some believe impossible and undesirable) vision of 'human-out-of-the-loop'.**

***Index Terms*—Autonomic Computing, Analytics, CPS**

I. Introduction

Autonomic computing is rapidly becoming established as a significant strategic approach to the design of more reliable, easier-to-manage computer based systems. When launching the autonomic computing initiative, IBM highlighted the growing complexity crisis in the IT industry, comparing it with telephony in the 1920s. There, the rapid increase in use of the telephone led to estimates that by the 1980s half of the population of the USA would have to be employed as telephone operators to meet the demand [1]. The implementation of automated switching and other technological developments avoided this crisis. By analogy, IBM is expecting autonomic system implementations to achieve similar productivity gains. It is anticipated, however, that significant research and development will be required to achieve that goal.

The envisaged goal of autonomic computing is the production of systems that are self-managing in four main respects: *self-configuring*, *self-healing*, *self-protecting* and *self-optimizing*. Some of the prerequisites for autonomic computing include complete visibility of the managed platform, complete control of that platform without undesirable side effects, and complete knowledge of how to relate visible situations to concrete actions. Most importantly is the ability to capture and represent both enterprise and personal policy (rules). Because of the need for differing levels of human involvement, autonomic computing maturity and sophistication has been categorized into five "stages of adoption" [2][3]: *Basic, Managed, Predictive, Adaptive,* and *Autonomic.* These prerequisites are priorities in the work reported in this paper while evolving along the autonomic computing maturity stages.

There are two strategies for introducing autonomic behavior. The first is to engineer it into systems and the second is to achieve it through adaptive learning. The first approach can be progressed immediately, with human experts generating or overseeing the generation of rules for autonomic functions. Over time, this could be increasingly supplemented with self-learning processes [4].

The initial vision for autonomic computing was that these rules, policies and beliefs were "management" oriented to facilitate self-management of the system. Yet given the advantage that AC provides – localized monitoring and adapting, this may be extended to "analytics" along with other areas, - deeper rules, policies and beliefs and not just for the automation of the system management (self-management) but the application and about the wider world in which the cyber-physical system operates.

II. Autonomic Computing Paradigm

The basic building blocks of any autonomic system architecture include sensors and effectors [5]. By monitoring behavior through sensors, comparing this with expectations (historical and current data, rules and beliefs), planning what action is necessary (if any) and then executing that action through effectors, creates a control loop [6]. The control loop, a success of manufacturing science for many years, provides the basic backbone structure for each system component [7].

Figure 1 & 2 depicts IBM's view of the necessary components within an autonomic manager. (For an alternative artifacts view, see [8].) It is assumed that an autonomic manager is responsible for a managed element within a self-contained autonomic element. Interaction will occur with remote autonomic managers through virtual, peer-to-peer, client-server [9] or grid [10] configurations.

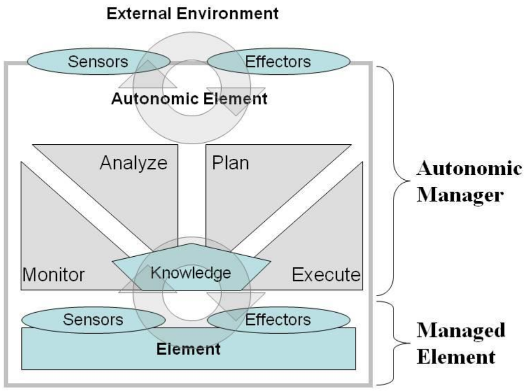

Figure 1 IBM's view of the architecture of an Autonomic Element [6].

The monitor and analyze parts of the structure process information from the sensors to provide both self-awareness and an awareness of the external environment. The plan and execute parts decide on the necessary self-management behavior that will be executed through the effectors.

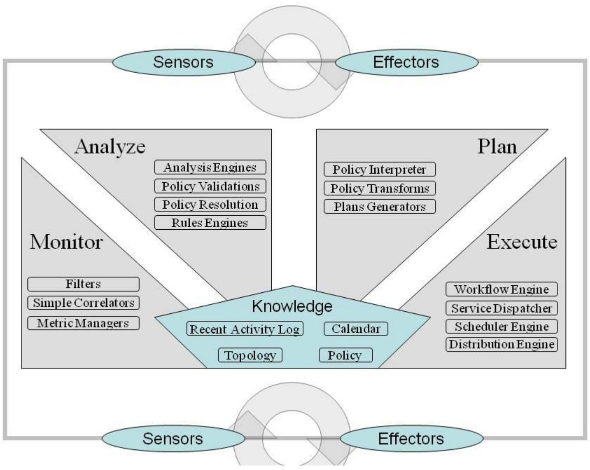

Figure 2 Necessary Components within IBM's view of an Autonomic Manager[3]

The simple correlator in the monitor parts and the rules engine in the analyze part use correlations, rules, beliefs, expectations, histories and other information known to the autonomic element, or available to it.

Figure 3 logically depicts each element in a system having an AM and achieving global self-management through cooperative communication (sH:self-healing; sO:self-optimizing; sC:self-configuring; sH:self-healing and s*: other self-management events).

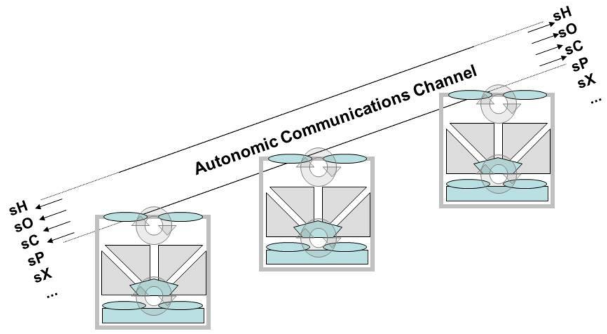

Figure 3 Cooperative environment of AEs

III. Analytics

A. Analytics

Wikipedia simply describes Analytics as: the discovery and communication of meaningful patterns in data. Especially valuable in areas rich with recorded information, analytics relies on the simultaneous application of statistics, computer programming and operations research to quantify performance. Analytics often favors data visualization to communicate insight and that firms may commonly apply analytics to business data, to describe, predict, and improve business performance. Specifically, arenas within analytics include enterprise decision management, retail analytics, store assortment and stock-keeping unit optimization, marketing optimization and marketing mix analytics, web analytics, sales force sizing and optimization, price and promotion modeling, predictive science, credit risk analysis, and fraud analytics. Since analytics can require extensive computation (re Big Data), the algorithms and software used for analytics harness the most current methods in computer science, statistics, and mathematics.

From a computer science perspective this is concerned with event correlation.

B. Event Correlation

The principle aim of event correlation is the interpretation of the events involved. The event signals or messages represent *symptoms*. Rules and beliefs identify which events to correlate and how they should be transformed. These tend to vary over time creating a significant maintenance burden [11]. Machine learning, data mining and other AI techniques can assist in the discovery of correlation rules and beliefs [12][13]. However, a human-centered discovery process is more effective than either a human or computer operating independently [14]. For example, it is useful to provide various visualizations of data throughout the knowledge discovery process to build user trust in the process and hence instill more confidence in the mined patterns. The transformation from data to knowledge requires interpretation and evaluation, which can also benefit from visualization of the processes involved.

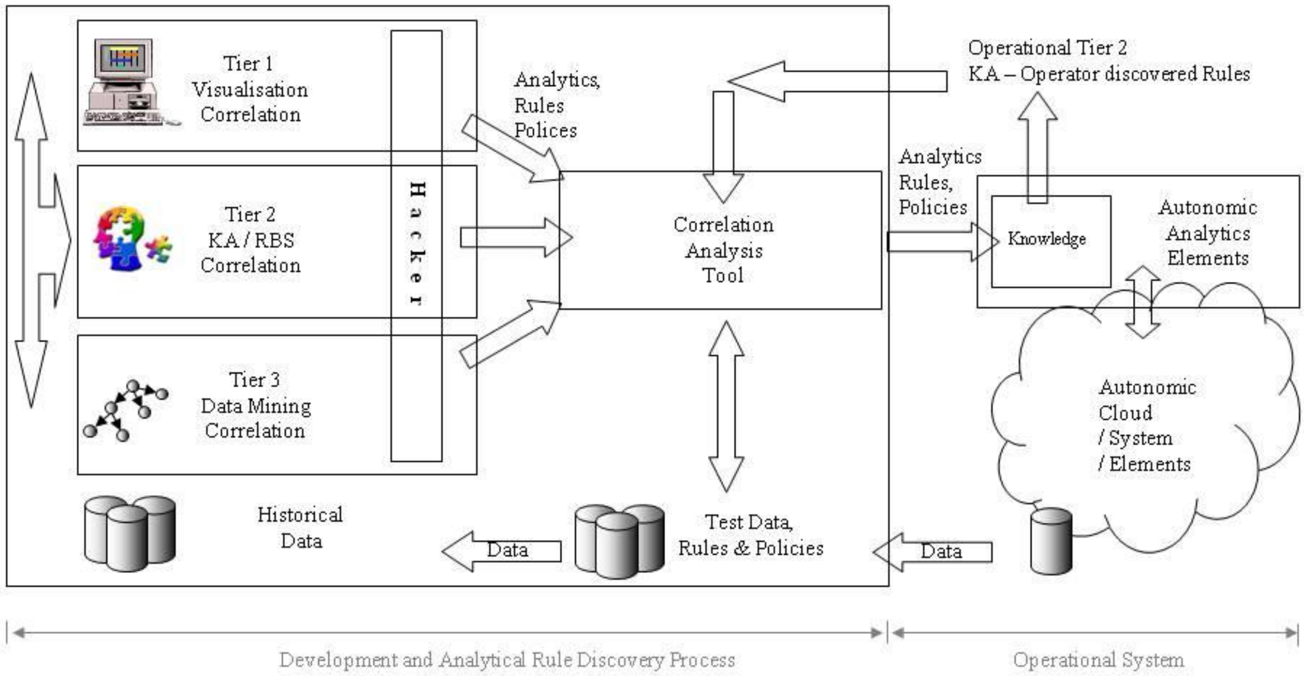

Figure 4 ***Three-tier analytical event correlation rule discovery process***

Visualization techniques can make use of the highly tuned perceptual abilities that humans possess, such as a capacity to recognize images quickly and to detect the subtlest changes in size, color, shape, movement or texture. Any patterns that emerge may indicate the presence of potential for new rules. Human interpretation is then required to transform them into 'knowledge'. Human input typically produces more meaningful insights into the discovered correlations, enabling them to be coded as useful rules for fault identification and management.

The next section describes a framework and support tools to assist such event correlation rule discovery.

IV. Event Correlation Framework

A three-tier architecture model for analytics & rule discovery is shown in Figure 4. This extends earlier work described in [15],[16],[17]. It also makes explicit a recommendation for extending tier 2 activities from the development phase into the operational phase by using knowledge management techniques to capture operators' manual live correlations of events, bring this knowledge into the development lifecycle and test to see if the rules are of general use.

The right-hand side of the diagram represents the managed operational system and the left-hand side the discovery or learning process. Data flows from the system to the discovery process; while rules flow from the discovery process to the autonomic manager. The representation suggests a cycle of activity, reflecting the necessary review that must take place after changes have been made to the system. Computer-assisted human discovery and human-assisted computer discovery techniques can be integrated in the three-tier framework for the discovery of event correlations to support the deduction of analytical management rules. The responsibility of the tiers is as follows:

Tier 1. *Visualization Correlation* (Computer-aided, human discovery). New event correlations are discovered from visualizing the analytical management data.

Tier 2. *Knowledge Acquisition or Rule Based Correlation*. New event correlations are discovered through consultation with experts and analysis of documentation. Correlations from tiers 1 or 3 may also be validated in this tier.

Tier 3. *Data Mining Correlation* (Human-aided, computer discovery). New event correlations are revealed by mining the analytical event data.

New rules may emerge from any of these tiers. The first tier, visualization correlation, supports the visualization of data in several forms. Visualization has a significant role throughout the knowledge discovery process, from data cleaning to mining. In particular, it facilitates the analysis of data to help identify event correlations (knowledge capture). The second tier aims to identify correlations and rules using more traditional knowledge acquisition techniques, with experts and documentation. At the same time it has a supporting role to confirm that discoveries from tiers 1 and 3 are indeed new and useful information. The third tier mines the system data to produce more complex correlation candidates.

V. Autonomic Analytics

The Autonomic Computing paradigm essentially places elements that are traditionally not managed, human managed, or managed centrally, into a cooperative managed environment via localized autonomic managers.

The majority of instances till now of autonomic computing are essentially still centralized, although the original vision was effectively peer-to-peer localized management with awareness of the environment and cooperation with other autonomic managers (AMs) to enable global systems management.

The implication intended behind the earlier statement of “traditionally not managed” is that elements like data, performance metrics, robots, and so forth have not been managed in terms of the overall system. To do so now implies fine grained low level elements are as key as major elements such as servers. Realistically this will increase complexity and naturally there will need to be hierarchical/priority views within this scheme to enable effective vertical orchestration.

This autonomic world (systems of systems) view therefore implies that the autonomic computing is not just about managing servers but potentially everything in the cyber-physical system [18]. This world view implies that Autonomic Computing’s peer-to-peer, cooperative, distributed paradigm is ideal for implementing such systems. These localized autonomic managers are in prime position not just to manage the “subconscious” autonomic management functions, but the automation of the applications goals-in this case the analytics. As such the “knowledge” element in the AE (figure 1) would not just contain management data but distributed analytics which the autonomics self-function can propagate updates as necessary.

VI. Conclusion

This paper has discussed the concept of utilizing the Autonomic Computing paradigm as a platform to provide real-time distributed context & situation aware analytics and as such achieve “Autonomic Analytics”. The paper also discussed a three tier analytics correlation discovery/learning process. The overarching vision of AC, and its derivation as AA, will only be achievable through standards, in particular, for communicating between AEs and interfacing with elements. Self-publishing & self-defining properties will assist here.

References

[1] P. Horn, "Autonomic computing: IBM perspective on the state of information technology", IBM T.J. Watson Labs, NY, 15th October 2001. Presented at AGENDA 2001, Scotsdale, AR. (available http://www.research.ibm.com/autonomic/), 2001

[2] D.F. Bantz, D. Frank, “Challenges in Autonomic Personal Computing, with Some New Results in Automatic Configuration Management”, Proceedings of IEEE Workshop on Autonomic Computing Principles and Architectures (AUCOPA 2003) at INDIN 2003, Banff, Alberta, Canada, 22-23 August 2003

[3] IBM, “An architectural blueprint for autonomic computing”, April 2003, revised Oct. 2003.

[4] R. Sterritt, (Dec 2002) "Towards Autonomic Computing: Effective Event Management", Proceedings of 27th Annual IEEE/NASA Software Engineering Workshop (SEW), Maryland, USA, December 3-5 2002, pp 40-47

[5] A.G. Ganek and T. A. Corbi, The dawning of the autonomic computing era, IBM Systems Journal, Vol. 42, no. 1, 2003, pp 5-18

[6] Autonomic Computing Concepts, IBM, 2001

[7] A. Ganek, “Autonomic Computing: Implementing the Vision”, Keynote presentation at the Autonomic Computing Workshop, 5th Int. Workshop on Active Middleware Services (AMS 2003), Seattle, WA, 25th June 2003.

[8] R. Sterritt, D.W. Bustard, “Towards an Autonomic Computing Environment”, 1st International Workshop on Autonomic Computing Systems at 14th International Conference on Database and Expert Systems Applications (DEXA'2003). Prague, Czech Republic Sept. 1-5, 2003.

[9] D.F. Bantz, C. Bisdikian, D. Challener, J.P. Karidis, S. Mastrianni, A. Mohindra, D. G. Shea, and M. Vanover, Autonomic personal computing, IBM Systems Journal, Vol. 42, no. 1, 2003, pp 165-176

[10] G. Deen, T. Lehman, J. Kaufman, “The Almaden OptimalGrid Project”, Proceedings of the Autonomic Computing Workshop, 5th Int. Workshop on Active Middleware Services (AMS 2003), Seattle, WA, pp 14-21, June 2003.

[11] Bratko, S. Muggleton, "Applications of Inductive Logic Programming", Communications of the ACM, Vol. 38, no. 11, 1995, pp 65-70.

[12] R. Sterritt, D.W. Bustard,, Fusing Hard and Soft Computing for Fault Management in Telecommunications Systems, IEEE Trans. Systems Man and Cybernetics part C, 32(2)

[13] R. Sterritt, “Facing fault management as it is, aiming for what you would like it to be”, Soft-Ware: 1st International Conference on Computing in an Imperfect World, Belfast 8-10 Apr., LNCS2311, Eds. D.W. Bustard, W. Liu, R. Sterritt, Springer-Verlag, 2002

[14] R. Uthurusamy, "From Data Mining to Knowledge Discovery: Current Challenges and Future Directions", Advances in Knowledge Discovery & Data Mining, AAAI Press & The MIT Press: California, 1996, pp 561-569.

[15] R. Sterritt, (Apr 2001) “Discovering Rules for Fault Management”. Proceedings of Eighth Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS ‘01)

[16] R. Sterritt, D.W. Bustard, "Practical Intelligent Support for Rule Discovery in Fault Management Systems", Cybernetics & Systems: An International Journal, Vol. 33, No. 6, Taylor & Francis, pp 579-601, Sep 2002

[17] R. Sterritt, (2004) *Autonomic networks: engineering the self-healing property.* Engineering Applications of Artificial Intelligence, 17 (7). pp. 727-739

[18] R. Sterritt, and M.G. Hinchey, (2005) *Why Computer-Based Systems Should be Autonomic.* In: of 12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2005), Greenbelt, MD, USA.