Towards Autonomic Computing: Effective Event Management

Sterritt, R. (2002). Towards Autonomic Computing: Effective Event Management. In *Unknown Host Publication* (pp. 40-47). IEEE Computer Society. https://doi.org/10.1109/SEW.2002.1199448

Link to publication record in Ulster University Research Portal

Published in:
Unknown Host Publication

Publication Status:
Published (in print/issue): 01/12/2002

DOI:
10.1109/SEW.2002.1199448

Document Version
Publisher's PDF, also known as Version of record

Towards Autonomic Computing: Effective Event Management

Roy Sterritt
Centre for Software Process Technologies (CSPT),
School of Computing and Mathematics, Faculty of Informatics,
University of Ulster, Northern Ireland.
r.sterritt@ulster.ac.uk

Abstract

Autonomic Computing is emerging as a significant new approach for the design of computing systems. Its goal is the production of systems that are self-managing, self-healing, self-protecting and self-optimizing. Achieving this goal will involve techniques from both Software Engineering and Artificial Intelligence. This paper discusses one particular aspect of Autonomic Computing: event management. It considers the range of event handling techniques in use, particularly in relation to distributed systems. Intelligent approaches are illustrated using the example of event handling in telecommunication systems. In particular, the telecom survivable network architecture is analyzed to identify lessons and potential pitfalls for Autonomic Computing.

1. Introduction

Autonomic Computing is emerging as a significant new strategic and holistic approach to the design of computing systems. Its goal is the production of systems that are self-managing (sM), self-healing (sH), self-protecting (sP) and self-optimizing (sO), in effect bringing pre-emptive and proactive approaches to all areas of a computer system.

As the name implies the influence for the new paradigm is the human body's autonomic nervous system, which regulates vital bodily functions such as telling the heart how many times to beat, monitoring the body's temperature and adjusting the blood flow, all without conscious effort.

There is an increasing realization for the need for a paradigm shift in computing to fulfill equivalent functions to the Autonomic Nervous System (ANS).

This paper will discuss one possible approach to facilitating an awareness of the external environment which is required for the holistic approach, that being *event messages*. It will look at how the telecommunications domain relies on this approach and will identify lessons and potential pitfalls for Autonomic Computing (AC).

2. Back to the Future

IBM launched its new strategic direction [1], Autonomic Computing, by describing a growing crisis in the IT industry and comparing it with what occurred in telephony in the 1920s.

In the 1920s, with the rapid expansion and infiltration into daily life of the telephone, there was serious concern whether there would be enough trained operators to work the manual switchboards. Analysts predicted that by the 1980s, given that growth would continue, half the population of the USA would have to work as telephone operators to meet the demand [3]. AT&T/Bell System's implementation of the automated switching protocol and the developments since curtailed this crisis.

Currently unfilled IT jobs in the US alone number in the hundreds of thousands, even in uncertain economic conditions. Demand for IT workers is expected to increase by over 100% in the next five years. Some estimates for the next decade indicate that a billion people and millions of businesses using a trillion devices connected via the Internet will require over 200 million IT workers, close to the entire population of the US [1][2].

Despite this emphasis on systems support costs or total cost of ownership (TCO) and the potential benefit of reducing these metrics, AC is also related to several other research themes, which encompass other useful benefits.

Introspective Computing involves proactive and reactive approaches to improve overall system behavior by sharing and utilizing excess computing, memory, storage and other resources. These are very similar aims to AC's sM and sO.

Ubiquitous Computing emphasizes usability. It compares the current state of computing with early scribes who had to know how to prepare and make a parchment and ink just to be able to write. Autonomic functions will go a long way in making computing more usable.

Proceedings of the 27 th Annual NASA Goddard/IEEE Software Engineering Workshop (SEW-27'02)
0-7695-1855-9/03 $17.00 © 2003 IEEE

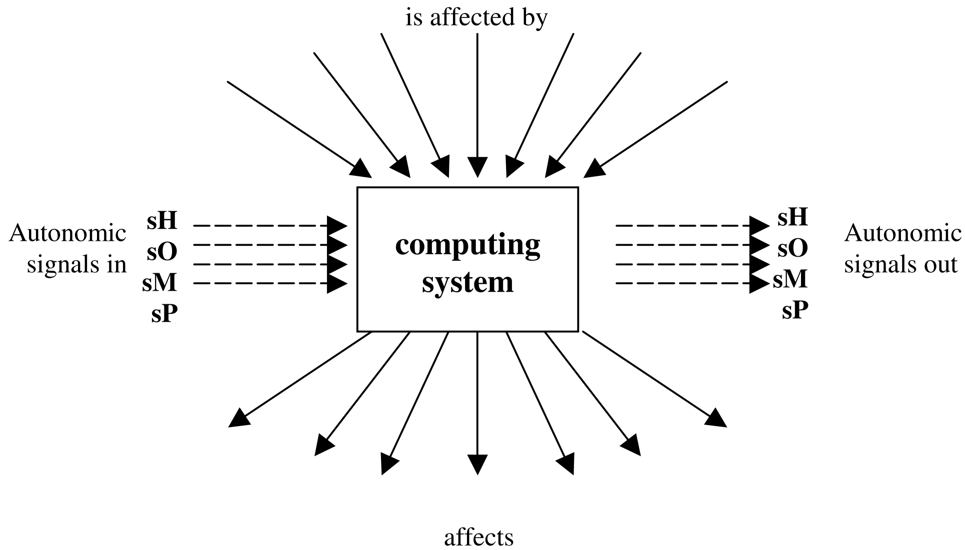

Figure 1 Event messages required with the environment to facilitate an Autonomic Computing system

The Ambient network view of the world is in effect a single system with billions of 'networked information devices'. Although currently the research emphasis for making this a reality is on usability, dependability will increasingly become an issue [6].

3. Developing Autonomic Computing

The creation of Autonomic Computing, or methods of enabling electronic systems to respond to problems, recover from outages and repair faults, all on their own without human intervention is by no means a small undertaking [4].

The desire for automation and effective robust systems is not new, in fact this may be considered the prime motivation behind the pursuit of best practice software engineering (SE). The desire for systems self-awareness, awareness of the external environment and the ability to adapt are not new either being major goals of Artificial Intelligence (AI) research for many years. What may be considered new in Autonomic Computing is the holistic vision and perhaps the facing up to the need in computing for a leap forward.

This may be viewed as the natural fulfillment of what has been increasingly occurring, the collaboration of AI and SE fields. Such collaboration is part of an ongoing trend prompted by increasing system complexity and a more demanding user community. For example, software engineers have used AI techniques to provide more sophisticated support for user interfaces and to help address soft issues in the development and operation of software. Likewise, the AI community has increasingly been looking to SE for disciplined methodologies to support the production of intelligent systems.

There are two perceived approaches to make Autonomic Computing a reality [1]:

- Making autonomic computing
- Achieving autonomic computing

Making autonomic computing has an implied Software Engineering view – to engineer autonomic function into the individual systems. It also has a near-to-medium time frame where human experts generate rules for autonomic functions.

Achieving autonomic computing has an implied AI view – to utilize self-learning algorithms and processes to achieve autonomic behavior. This has a medium-to-long term time frame.

Both rule type systems and adaptive algorithms require a system identity and to be *aware* of the system's components, its current status, interconnections with other systems and available resources in order to manage itself.

Another vital aspect along with the self identity is the ability to communicate and relate to the external environment, to receive signals about the changing environment and to inform the environment of autonomous activity performed by these system that may effect the environment. This is depicted in Figure 1.

The telecommunications industry, discussed in the next section, reached this crisis point back in the 1920's.

4. Telecom Survivable Networks

Since the 1920s Telephony has evolved substantially in terms of automation. The Internet revolution saw telecommunications providing the vast infrastructure necessary for millions of computers to be interconnected, and perhaps provided a glance of the future - the merging of these two great industries. This section looks at the automation within modern telecommunications, and in particular fault management, to facilitate the discussion on lessons for Autonomic Computing.

As the world becomes increasingly reliant on computer networks the complexity of the networks has grown along a number of dimensions [8]. The phenomenal growth of the Internet has shown a clear example of the extent to which the use of computer networks is becoming ubiquitous [9]. As users demands and expectations on networks become more varied and complex so do the networks themselves. As such, heterogeneity has become the rule rather than the exception [8]. Data, in any form, voice, movie, or actual information, may travel under the control of different protocols through numerous physical devices manufactured and operated by large numbers of different vendors. There is a general consensus, in dealing with such data, the trend towards increasing complexity will continue rather than abate.

The complexity lies in the accumulation of several factors; the embedded increasing complexity of network elements, the need for sophisticated services and the heterogeneity challenges of customer networks [10].

Network management encompasses a large number of tasks with various standards bodies specifying a formal organization of these tasks. The International Standards Organization (ISO) divides network management into six areas as part of the Open Systems Interconnection (OSI) model; configuration management, fault management, performance management, security management, accounting management and directory management which sit within a seven layer network hierarchical structure.

Yet with the Internet revolution and the convergence of the Telcos and Data Comms other realities are forming. The trend is towards a flatter structure.

Telecommunication Systems through the Synchronous Digital Hierarchy (SDH) or SONET (Synchronous Optical NETworks) in the USA, provides extensive high bandwidth that has become the backbone of today's Internet. Millions of calls are multiplexed together onto backbone fibre optics from tributary lines. STM-1 (Synchronous Transfer Mode, which equates to STS-3, Synchronous Transport Signal, in SONET) the first level provides 155.52 MBit/s (mega bits per second). STM-64 is 64 times this first level, 9953.28 MBit/s, and equates to 1000 paperback books per second down a glass fibre thinner than a human hair.

The systems are designed with robustness in mind since it is simply not acceptable for millions of calls to be cut-off due to a faulty network element (NE) or a software upgrade. This results in design decisions such as utilizing a ring topology to implement a survivable network architecture. In the SDH world traffic travels both directions as a matter of course, with protection configured into the network. Any faulty components preventing travel will cause an automatic switch in direction of the traffic to avoid the failure area, thus preventing loss of traffic.

For major hub traffic applications this survivability tends to be implemented through an additional dedicated protection ring (1+1 protection). In metropolitan, junction and trunk network applications this may be achieved through the less expensive deployment option of a shared protection ring, where protection capacity is reserved in the existing ring in case of failure.

This level of robustness can only be achieved by providing extensive built-in redundancy into the hardware and software components in the multiplexers. This can result in the system being quite complex, which is added to with the intentional design to allow old non-synchronous traffic to exist along side synchronous traffic within its structure (legacy support) as well as facilities for network management.

Central to the management of these complex networks is event messages. In relation to the autonomic nervous system analogy these may be compared to the electric pulses that travel along the nerves.

When a fault does occur in the SDH network the complexity and built-in redundancy often results in a large number of alarm events being raised and cascaded to the Element Controller (the manager). The behavior of the alarms is so complex it appears non-deterministic [13], making it very difficult to isolate the true cause of the fault [14]. Failures in the network are unavoidable but quick detection and identification of the faults responsible is essential to ensure robustness. To this end the ability to correlate alarm event messages becomes very important [15]. The major telecommunication equipment manufacturers deal with event correlation through alarm monitoring, filtering and masking as specified by ITU-T [16] and other international standard bodies. Resulting rule type diagnostic systems provide assistance to the operator whose expertise is then used to determine the underlying fault (or faults) from the filtered set of alarms reported.

The skill of the operator is central to identifying the fault. As such, although automation prevents the immediate loss of traffic and general function of the system, intervention is necessary to determine and resolve the problem - the approach is not as far advanced as Autonomic Computing aims to become.

5. Event Messages & Autonomic Computing

The principle aim behind alarm event correlation is the determination of the cause. The event messages represent the symptoms of what is occurring and since the domain is likely to be complex and uncertain a single symptom (event message) is unlikely to directly indicate a cause or causes, such as a fault. It is the combination or correlation of the symptoms that is likely to indicate the causal event. This is necessary to then empower the system to make self-managing, self-optimizing, self-protecting and/or self-healing decisions.

This need is depicted in figure 1 where the relationship of interconnected systems is expressed. Since the majority of systems are highly interconnected, a trend which is likely to continue with the move towards the ambient network, expressed in the research efforts towards the semantic web and the semantic grid, a system's actions affects other systems and that system is affected by the actions of those other systems.

The figure also depicts autonomic signals both entering and leaving the system. This indicates the need for the environmental awareness in autonomic computing to facilitate the autonomic functions of sM, sH, sO and sP.

From a high level perspective there are two possible approaches – a centralized approach or a distributed approach.

The centralized approach is where polling of a sub-system requesting its status may take place from a central source. That central source holds responsibility to determine what is occurring in the environment and would then provide feedback to the sub-system on any necessary self-managing activity required to be undertaken.

The distributed approach is where event messages or signals are transmitted to the environment by individual sub-systems expressing their autonomic behavior that may affect the environment. While in parallel, the sub-systems would receive from the environment relevant event messages/signals that through correlation may indicate the need to undertake some self-managing activity.

In effect the best solution will likely be a combination of the above two approaches. Both require autonomic function (the ability to sM, sH, sO & sP) and autonomic signaling (the ability to send and receive from the external environment) to be engineered into the systems. In effect the practical implementation of such an autonomic signal may be through mobile intelligent agents.

Within the computational grid domain the OGSA (open grid services architecture) has a facility referred to as the Globus Heartbeat Monitor (HBM) which is designed to detect and report the failure of processes that have identified themselves to it through the presence or absence of the heartbeat (assuming network connection is functioning) [17]. At first inspection this approach is a central approach through the HBM, yet each host in the grid has a HBM thus effectively a hybrid solution.

The HBM function potentially only addresses self-healing within autonomic computing. More information is required to facilitate sM, sO and sP. Clearly for autonomic computing to be effective in heterogeneous environments requires a open research agenda on specifying an architecture for autonomic function and signals.

A key element of autonomic computing will be the ability to correlate those signals, potentially expressed as event messages, to determine what is occurring in the environment that may then require autonomic function to be triggered.

It has been established that in the telecommunications exemplar, the survivable network architecture attempts to ensure continued service but does not necessarily determine the fault without human intervention. There has been considerable research effort in utilizing AI to achieving automated fault identification through the correlation of event messages.

The author's work in this area is considered in the next section. It has two main aims, firstly a process to help with the development of new correlation rules and secondly seeking a better approach than just rules for the actual correlation of the events.

6. Case Study: Telecoms Event Management

Essentially the situation has arisen that a large number of uncorrelated alarm event messages may reside on a network at any one time. One estimate concerning BT's UK network was that 95% of all alarm events raised remain uncorrelated, amounting to tens of thousands alarm events being active at any one time. Over time this amounts to a substantial load of data that may be considered for data mining (knowledge discovery – KD) to produce potential rules or define patterns used to further automate the event management task and reduce the overall burden of uncorrelated events.

Uthurusamy (1996) proposed that a challenge of paramount importance for KD was to have a more human-centered view. Pressing the need for highly interactive human-centered environments as outlined by the KD process [11] would enable both human-assisted computer discovery and computer-assisted human discovery. That such tools, by reducing time to understand complex data sets, would enable practical solutions to many real world problems far more rapidly than either human or computer operating independently [12].

This section briefly reviews a three-tier framework which has the objective to integrate human discovery and computer discovery into the development process [18]. A prototype tool that integrates the process through visualization, data mining and knowledge management is also reviewed [19]. Finally, an alternative to a rules solution for the correlation of events is discussed [20].

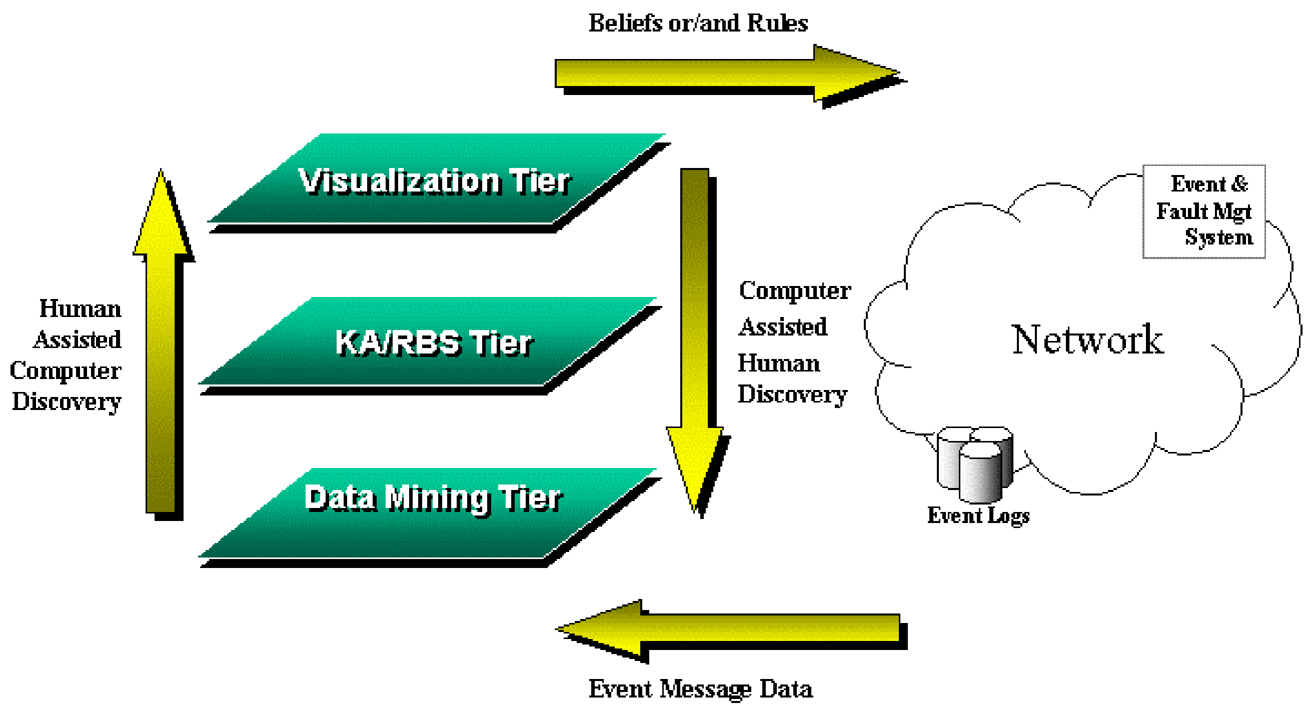

Figure 2 Three tier discovering process -
Computer-Assisted Human Discovery and Human-Assisted Computer discovery.

Figure 2 illustrates a high level three tier development process based on the principle of rule or belief discovery from heterogeneous sources through heterogeneous techniques [7].

The objectives are to help avoid problems typically associated with rule development from individual techniques, such as; knowledge acquisition bottleneck, maintenance burden, hidden implicit and tacit knowledge and black box machine learning and discovery techniques.

The process uses data captured from the domain (in this case the fault management domain within telecommunications) along with data mining algorithms (computer discovery) and appropriate visualization techniques (by a domain expert for human discovery). The visualization potentially opens up the data to assist understanding of the data mined discoveries. The middle tier, knowledge acquisition, potentially captures tacit and implicit knowledge of the experts as well as making explicit knowledge from design specs or standards. It often provides the validation for discoveries be they human or computer.

Figure 3 depicts a screenshot of a prototype tool which implements these principles [19]. It represents the events visually in a Gantt chart. Human discovery (tier 1) is facilitated by allowing the domain expert to scan through the data and choose potential correlations (human discovery) and then automatically write this as a rule.

The computer discovery (tier 3) is facilitated by executing a simple mining algorithm which automatically reports discovered correlations and attempts to display these in the visualization screen when requested. The expert then chooses which of these are valid discoveries and chooses the *write rule* option.

Tier 2, knowledge acquisition, is facilitated by ensuring that in either case (human or computer discovery) when a user instructs a rule to be written the expert users reasoning is captured in a simple input dialog. Discoveries considered to be invalid by the expert may also be recorded in an ignore dialog. As such, a case base of positive and negative cases are built with use and can lead to facilitating future self-adaptive functionality.

In effect this process and prototype tool may be considered a knowledge management system. The visualization is making knowledge *explicit*, the data mining is potentially discovering *unknown* knowledge and the middle tier is potentially capturing *implicit* and *tacit* knowledge.

Although the focus has been on rules, meeting the current demands of the domain, it is a well researched view that rules alone are not enough to provide full autonomic function. There has been much research into utilizing AI within this domain in an attempt to achieve this (summarized in [21],[22]).

Holding with the principle of utilizing open and visible processes and techniques Bayesian belief networks (BBNs) have been researched to provide an intelligent solution for fault management systems [20]-[22]. The technique handles uncertainty, which is inherent within the domain, through utilizing probability theory.

The three-tier process and tool may be utilized to discover beliefs as opposed to rules which may be utilized directly in constructing a belief network or used as a priori information with a belief network learning algorithm.

The relevance of this work for Autonomic Computing lies in the desire to bridge the gap between making autonomic systems and learning such autonomic behavior.

Both approaches require signals to be expressed in some form throughout the environment indicating system behavior that may highlight the need for some self-managing activity. Both approaches will require the ability to correlate these signals or events in some way to obtain a fuller picture. The software engineering approach will attempt to engineer in rules or beliefs to react to the environment whereas the AI approach will attempt to self-adapt or evolve an optimal solution that copes with the conditions in the environment. For both approaches to be effective requires open processes, systems and architectures.

7. Relevance to NASA

The NASA community has expressed a growing interest and acceptance of adaptive operations and onboard autonomy [23]. NASA missions, particularly those with a deep space dimension, are increasingly looking towards utilizing autonomic decision making to reduce the lag time between seeking and receiving instructions from the ground upon encountering new situations. Two of the first notable missions to utilize autonomy are DS1 (Deep Space 1) and the Mars Pathfinder.

The DS1 mission highlights the increasing interest within NASA in obtaining adaptation and autonomy through remote agents [24]. One of the interesting initiatives to come from the DS1 mission was the beacon monitor concept [25]. With beacon monitoring, the spacecraft sends a command to the ground that instructs how urgent it is to track the spacecraft for telemetry. This marked a paradigm shift over traditional operations, fitting with the new reality of the faster, better, cheaper strategy and taking advantage that the craft may be able to operate autonomously for long periods.

In terms of high-level concepts this beacon monitor is similar to the heart beat monitor in grid computing while including a tone that indicates a degree of urgency for attention. The telecoms example indicated that large amounts of event data can be raised under fault conditions. The concept of a constant autonomic signal of events will create challenges in terms of data handling and efficiency concerns with continuous event correlation. A possible hybrid approach would be to utilize the urgency concept of the beacon monitor – to indicate a rising need to self-heal, self-manage, self-protect or self-optimize – within the heart beat monitor to indicate a need for the correlation of events in the autonomic signal to assess the external environment.

8. Conclusion

Autonomic computing is an emerging holistic approach to computer system development that aims to bring a new level of automation to systems such as self-healing, self-optimizing, self-managing and self-protection.

For Autonomic Computing to reach these goals will require open standards and technologies. Without focusing on its implementation, this paper specifically discussed the concept of event messages for providing the necessary communications with the external environment.

The telecommunications industry was studied as an example that uses event messages within its systems. The telecommunications approach to fault handling attempts to ensure under reasonable circumstances that the functionality of the system continues. Yet the approach does not necessarily identify the actual underlying fault. From a high-level view this may seem to fit with the analogy, where a human may seek medical attention yet at a lower level it is far removed from the self-healing and managing goal of Autonomic Computing.

The research tool discussed in the telecommunications case study indicates a potentially important lesson for Autonomic Computing. This tool retrospectively seeks to find cause and effect rules or patterns between the event messages to determine the fault by capturing experts knowledge concerning correlations as well as utilizing data mining techniques.

This high level three tier process and tool could equally be applied to Autonomic Computing signals between heterogeneous systems to assist in engineering in autonomic function or assist in explaining evolving autonomic behavior.

The NASA community is increasingly accepting and utilizing autonomy. It can only benefit from a paradigm shift within computing towards Autonomic Computing as well as contribute towards it.

It is essential that Autonomic Computing finds a way forward that captures this knowledge within the system, in a transparent and open form, from the start and avoids the situation where thousands of event messages sit in the system uncorrelated.

Acknowledgements

The Centre for Software Process Technologies is supported by the EU Programme For Peace And Reconciliation In Northern Ireland And The Border Region Of Ireland (PEACE II).

The telecommunications research was undertaken in the Jigsaw project (ITS Start 187), funded by Nortel Networks' Belfast Labs and the Industrial Research and Technology Unit (IRTU).

Special thanks must go to my placement student, Fiona O'Donnell for her efforts, who is undertaking a research assistant placement funded by the University of Ulster (2002/03).

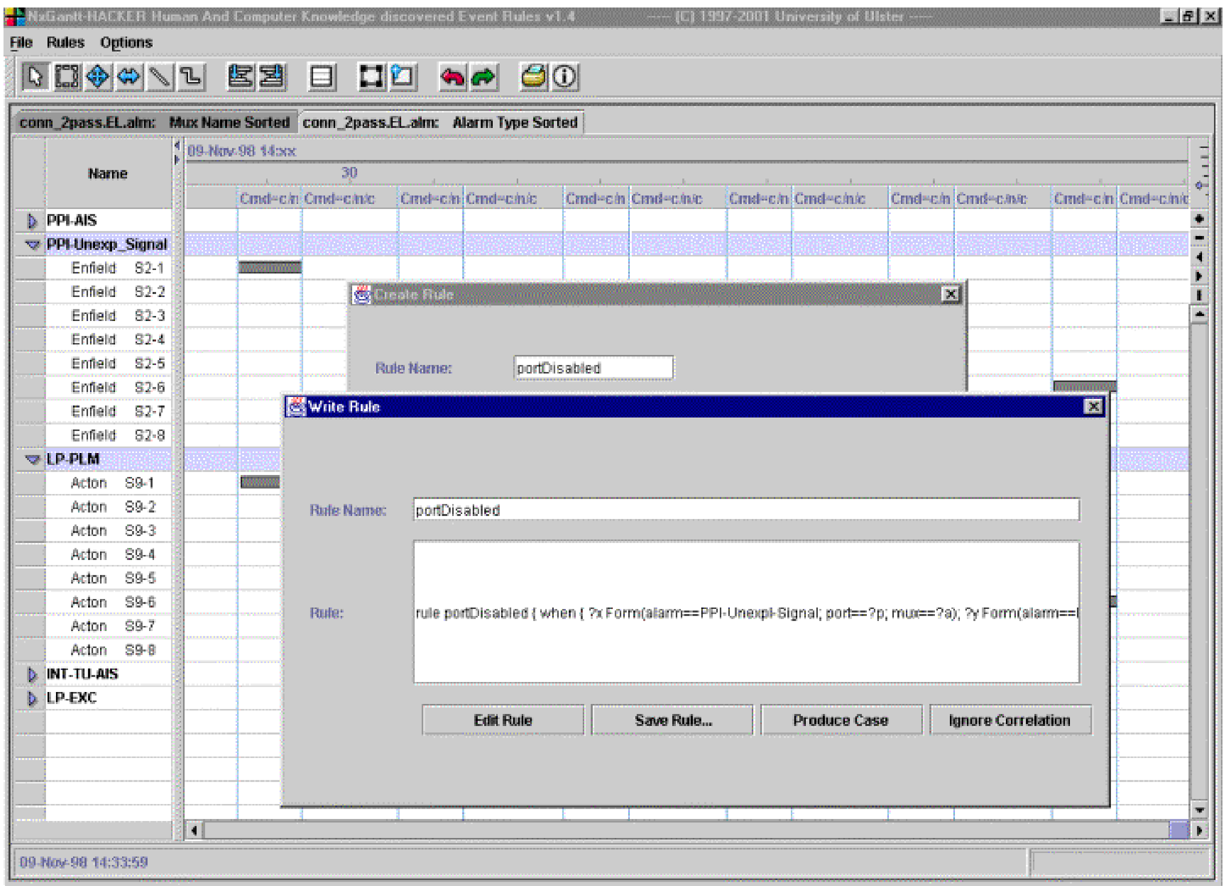

Figure 3 HACKER Screenshot
Human And Computer Knowledge discovery Event Rules.

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] E. Mainsah, "Autonomic computing: the next era of computing", IEE Electronics Communication Engineering Journal, Vol. 14, No. 1 (Feb), pp2-3, 2002
[3] J. Fitzgerald, A. Dennis, "Business data communications and networking, John Wiley, 5th edn, 1996.
[4] A. Wolfe, "IBM sets its sights on 'Autonomic Computing'", IEEE Spectrum, Jan., p18-19, 2002
[5] L.D. Paulson, "IBM Begins Autonomic Computing Project", IEEE Computer, Feb., p25, 2002
[6] B. Randell, "Facing up to faults", The Computer Journal, ACM Press, 43(2), 2000
[7] R. Sterritt, Discovering Rules for Fault Management, Proceedings of IEEE International Conference on the Engineering of Computer Based Systems (ECBS), Washington DC, USA, April 17-20, 190--196, 2001
[8] T. Oates. Fault identification in computer networks: A review and a new approach. Technical Report 95-113, University of Massachusetts at Amherst, Computer Science Department, 1995.
[9] C. Bournellis. Internet '95. Internet World, 6(11):47--52, 1995.
[10] M. Cheikhrouhou, P. Conti, J. Labetoulle, K. Marcus, Intelligent Agents for Network Management: Fault Detection Experiment. In Sixth IFIP/IEEE International Symposium on Integrated Network Management, Boston, USA, May 1999.
[11] R.J. Brachman, T. Anand, "The Process of Knowledge Discovery in Databases: A Human-Centered Approach"., Advances in Knowledge Discovery & Data Mining, AAAI Press & The MIT Press: California, 1996, pp37-57.
[12] 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.
[13] A. T. Bouloutas, S. Calo, and A. Finkel, "Alarm correlation and fault identification in communication networks", IEEE Trans. on Comms., vol. 42, no. 2/3/4, 1994.
[14] M. Klemettinen, "A knowledge discovery methodology for telecommunication network alarm databases", Ph.D. Thesis, University of Helsinki, Finland, 1999.
[15] G. Jackobson and M.D. Weissman, "Alarm correlation", IEEE Network, vol. 7, no. 6, pp. 52-59, Nov. 1993.
[16] ITU-T Recommendations, M.3010 principles for a telecommunications management network, Feb. 2000.
[17] The Globus Heartbeat Monitor Specification v1.0, http://www-fp.globus.org/hbm/heartbeat_spec.html
[18] R. Sterritt, D.W. Bustard, Practical Intelligent Support for Rule Discovery in Fault Management Systems, Cybernetics and Systems: An International Journal, 33 (6), Summer 2002
[19] R. Sterritt, E.P Curran, H, Song, HACKER: Human And Computer Knowledge discovered Event Rules for Telecommunications Fault Management, Proc. IEEE Int. Conf. Systems Man & Cybernetics, Oct. 2002
[20] 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)
[21] R. Sterritt, W. Liu, "Constructing Bayesian Belief Networks for Fault Management in Telecommunication Systems," First ENUITE Workshop on Computational Intelligence in Telecommunications and Multimedia at EUNITE 2001, pp 149-154, Dec. 2001
[22] 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
[23] J. Wyatt, R. Sherwood, M. Sue, J. Szijjarto, "Flight Validation of On-Demand Operations: The Deep Space One Beacon Monitor Operations Experiment", 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS '99), ESTEC, Noordwijk, The Netherlands, 1-3 June 1999.
[24] N. Muscettola, P. P. Nayak, B. Pell, and B. Williams, "Remote Agent: To Boldly Go Where No AI System Has Gone Before".. Artificial Intelligence 103(1-2):5-48, 1998.
[25] J.Wyatt, H. Hotz, R. Sherwood, J. Szijjaro, M. Sue, "Beacon Monitor Operations on the Deep Space One Mission", 5th Int. Sym. AI, Robotics and Automation in Space, Tokyo, Japan, 1998.