

Exploring Autonomic Options in an Unified Fault Management Architecture through Reflex Reactions via Pulse Monitoring

Roy Sterritt,
School of Computing and Mathematics
Faculty of Engineering
University of Ulster
Northern Ireland
r.sterritt@ulster.ac.uk

Darren Gunning, Alan Meban, Phillip Henning
BT Exact
BT Riverside Tower
Belfast
Northern Ireland
{[darren.gunning](mailto:darren.gunning@bt.com), [alan.meban](mailto:alan.meban@bt.com), [phillip.henning](mailto:phillip.henning@bt.com)} @bt.com

Abstract

This paper investigates the potential of adding Autonomic capabilities to the telecommunications fault management architecture and highlights the importance of a reflex-healing dual strategy to facilitate this advanced automation. The reflex reaction is facilitated through the concept of a pulse monitor – essentially the extension of the fault tolerant heartbeat monitor mechanism to incorporate reflex urgency levels and health check summary information.

1. Introduction

Autonomic systems are essentially self-managing systems (SMS) based on the biological metaphor of the autonomic nervous system. The increasing number of SMS initiatives stemming since IBM's Autonomic Computing strategic refocus in 2001 [1] indicates the importance of reducing the complexity and cost in managing today's systems [2]. The latest related strategic research area to be highlighted has been autonomic communications [3].

At the system level with the increasing convergence of computing and telecommunications both the autonomic computing and autonomic communications initiatives will be vital in contributing to the achievement of autonomic systems.

This paper will discuss the concept of incorporating a reflex reaction mechanism [4] into telecommunications fault management system architecture to facilitate new options for engineering autonomic properties into the system such as self-healing, self-protecting, self-optimizing and self-configuring.

The paper will first briefly describe an existing system architecture (Unified Fault Management - UFM) [5] and some of its complexities, it then describes the simple yet innovative means to incorporate reflex signals and finally discusses the autonomic architecture.

2. UFM System Architecture

A simplified view of the UFM system architecture is depicted in Figure 1 [5]. In a typical Telco architecture elements in the bottom layer, the physical network, tend to be outside the Telco's design control being supplied by many third party vendors, consequently the ability to incorporate autonomic functions at this layer is difficult due to the element specific interfaces. Where there are encouraging signs of autonomic computing research that will benefit this layer—for example refitting autonomic computing into legacy systems [6][7] and the IBM and Cisco agreement [8][9] concerning problem determination—the amount of different vendor equipment in a Telco network highlights the need for autonomic standards [10][11].

The next layer up reflects the variety of technologies a large Telco is required to manage and the complexity in the system. Each technology within the network – SDH (SONET), PDH, ATM, IP and so on, has its own specific technology fault manager (also referred to as domain FM). The individual element managers within the network pass up the alarms and event messages to their relevant technology manager.

Complexity lies within the system in that SDH frames may be carrying ATM which may be carrying IP and so on. A fault in SDH may thus affect the ATM and other higher technologies domains.

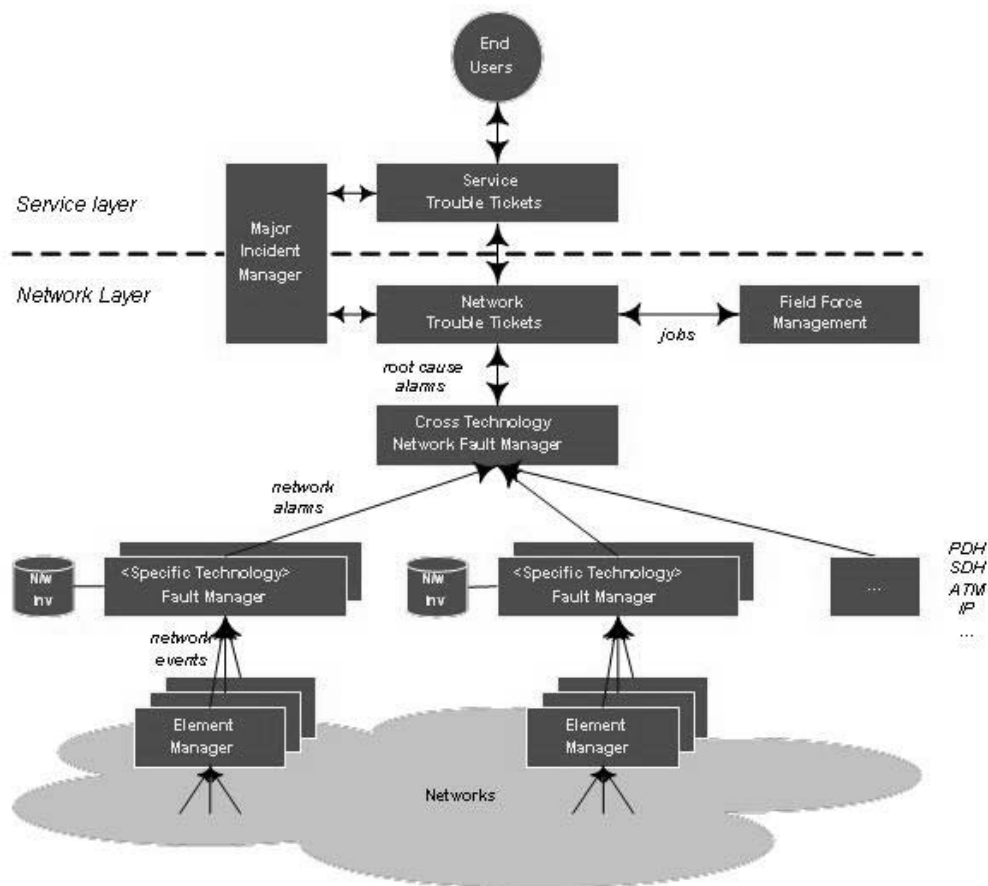


Figure 1 Simplified view of the Unified Fault Management System Architecture

It is only at the next layer, the cross technology network fault manager (xTech N/W FM), where a total view of all the different technologies occurs through cross-domain alarm and event correlation. Once the root cause has been determined either automatically or through operator assistance then the fault can be assigned to trouble ticket and task force management for remedy.

By its very nature root-cause analysis imposes delays at each management layer in the architecture for successful alarm and event correlation – essentially allowing time for the inter-related alarms from different sources to arrive at the correlating manager and the subsequent retrieval of the affect parts information. As such it is possible that through the combined inherent correlation timeframes of each layer in the architecture the customer may have informed the service layer of a major incident before the service layer is aware of it from the management architecture beneath.

This single speed of fault management information also limits the potential for further advanced northbound (network layer to customer) and southbound (customer to network layer) automation that may be considered autonomic behavior.

3. Reflex Monitoring Mechanisms

Reflexes and healing is a dual strategy approach concept inspired by biological systems [12]. Animals have a reflex system, where the nerve pathways enable rapid response to pain. Reflexes cause a rapid, involuntary motion, such as when a sharp object is touched. The effect is that the system reconfigures itself, moving away from the danger to keep the component functioning.

The body will heal itself on a much longer timescale. Resources from one part of the system are redirected to rebuild the injured body part, including repair of the

reflex response network. While this cannot help in the real-time response, directly after an event, it can prepare the system for the next event. In addition, it can readjust the system for operation with a reduced set of resources [12].

The typical approach for system management is based on events which are generated and sent under fault or problem conditions.

In the embedded system space the opposite is typically the case. A system management action occurs when something does not occur. An example is the fault tolerant mechanism of a heartbeat monitor (HBM), through a combination of the hardware (the timer) and software (the heartbeat generator) an 'I am alive' signal is generated periodically to indicate all is well [13]. The absence of this signal indicates a fault or problem. Some embedded processors have a hardware timer which, if not periodically reset by software, causes a reset/restart. This allows a particularly blunt, though effective, recovery from a software hang.

This approach offers the advantage that through continuous monitoring problem determination becomes a proactive rather than a reactive process.

In the fault management architecture the managers and their vital processes are guarded through the use of heartbeat monitors (HBM), for instance an SDH fault manager will send periodic heartbeats to the cross technology network fault manager. This fault tolerant approach is vital in safeguarding the fault management processes and ensuring the continued operation of the manager. The absence of this 'I am alive' signal enables the remote manager to take protective action such as switching to the back-up manager while investigating the absence of the beat.

The system monitoring of the network health by the managers through event messaging and the sub-system monitoring of the managers' health through HBM are vital activities currently carried out independently. Yet, the managers themselves are in a key position to give an indication of the health of the network as they see it at that moment in time. For instance, if a technology fault manager has noticed a sudden dramatic increase in the number of alarms arriving it may give instant indication to the cross-technology manager above.

The HBM and event correlation have distinct purposes yet they may be used together to offer new options. Since the managers are in a position to offer a view of the network health this may be piggybacked into the periodic heartbeat (a health value as a pulse) giving that indication to the other managers. The analogy is instead of measuring the presence or absence of the heartbeat (dead or alive) to measure the actual pulse as a health indicator. This provides the same heartbeat guard (presence or absence of pulse) for the manager along

with additional network health information that may be used for reflex reactions.

NASA has a similar construct the 'Beacon monitor' [14]. The spacecraft sends a signal to the ground that indicates how urgent it is to track the spacecraft for telemetry (Nominal, Interesting, Important, Urgent, No Tone). This concept involved a paradigm shift for NASA from routine telemetry downlink and ground analysis to onboard health determination and autonomous data summarization. In high-level concept terms, the beacon monitor is similar to the heartbeat monitor, with the addition of a tone to indicate the degree of urgency involved.

The logical difference between the pulse signal and general event messages is essentially that the pulse provides the mechanism for a reflex reaction whereas the general event messages under fault conditions form part of the slower healing process—root cause analysis from the event stream.

4. UFM Autonomic Reflex System Architecture

The UFM Autonomic Reflex System Architecture based around reflex pulse monitoring is depicted in Figure 2. Several important concepts concerning the use of this architecture are key to its success. Firstly, the HBMs already exist between the managers within the existing architecture providing a safe-guard against their failure. As such the retrofitting of autonomic reflex capabilities into the architecture is not as extensive an effort as may first seem since communication of the health indicator can be contained with the heartbeat (the pulse).

Secondly, an important point to note from the HBM, and also from the Beacon Monitor, is the minimization of data sent – essentially only a 'signal' is transmitted. Any move towards sending more information must not compromise this reflex reaction. As such the beat must contain within it the urgency level and health summarization only, or in the absence of bandwidth concerns information that can be acted upon quickly and not incur processing delays.

From this perspective the pulse has two mechanisms to indicate health information; health indicator summarization contained within the heartbeat and the urgency level also contained within the heartbeat or indicated by the rate at which the beat is sent. The HBM sends a beat at a constant interval, under normal conditions the pulse monitor would do likewise but upon encountering changing circumstances, such as a fault in the system, this regular interval will decrease, i.e. the pulse rate increases.

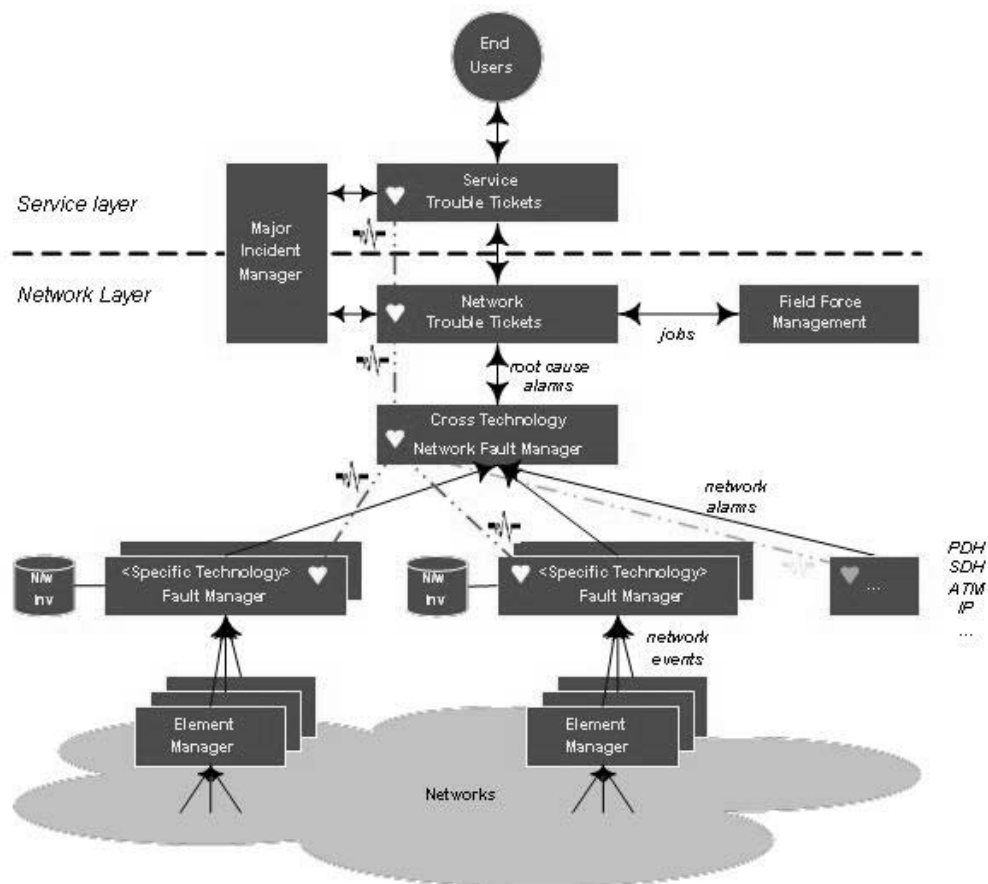


Figure 2 UFM Autonomic Reflex System Architecture

This dynamic rate apart from fitting with the biological metaphor, is necessary since if you take the situation where an event occurs that effects the system health this information would be delayed on average half the heartbeat timeframe. To facilitate the reflex reaction the heartbeat would be sent immediately and the pulse rate increase remain until the situation was resolved.

The pulse mechanism has been described as *conceptually* extending the HBM since *physically* there is still a role for the HBM and the two may co-exist in a system, for instance, granularly level where the component being guarded by a HBM has limited specific low-level functionality.

The pulse monitor is conceived for the situation such as a system monitor, which is vital and as such safeguarded by a HBM but it is also in a key position to provide an assessment of health conditions and provide a reflex mechanism. Since inherently the pulse monitor contains the 'I am alive' mechanism this use of a HBM may physically be replaced with a pulse monitor, as is

the case between managers within the UFM. In other situations, the pulse monitor will be in addition to the HBM, for instance HBM will exist within fault managers to guard internal vital processes.

The existing timeline within the Telco fault management (FM) architecture (Figure 3) highlights that root cause analysis (RCA) has inherent inbuilt delays to allow effective alarm correlation to take place. Figure 3 highlights how under fault conditions there exists a time gap between impact on customer and the knowledge of the fault at the service level. Within the BT network this may be in excess of 10 minutes.

Figure 4 highlights the potential of a reflex reaction within the architecture. Firstly the reflex mechanism (pulse monitor) informs the service level before the customer of a major fault. This then provides new options for southbound automation and autonomic behavior at this level: for instance through the knowledge of a major fault, the forthcoming customer's trouble tickets (TT) can be assigned to that fault.

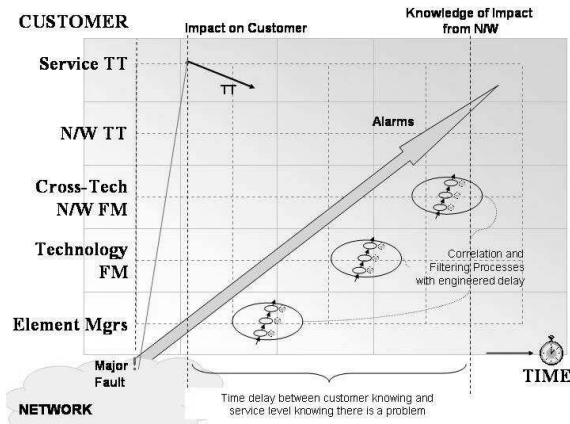


Figure 3 Single timeline - alarm progression through management hierarchy

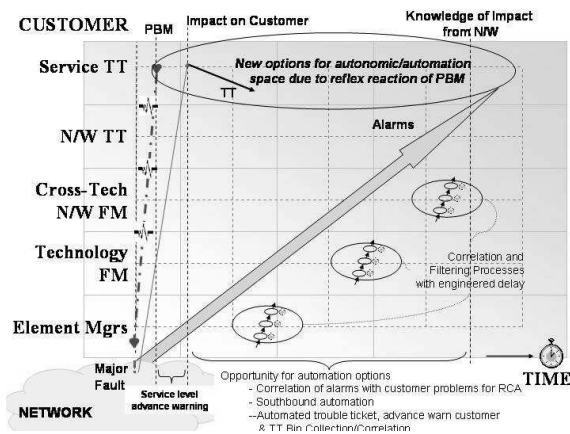
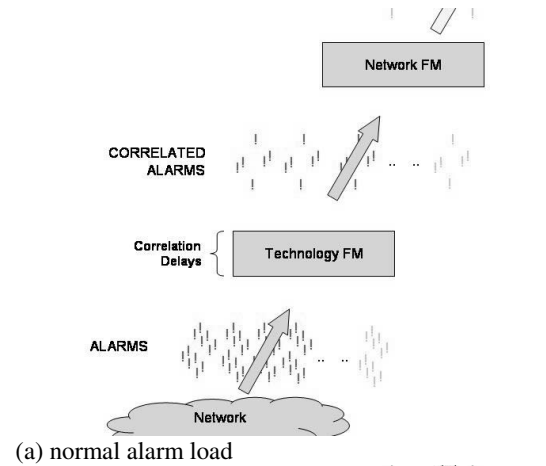


Figure 4 Dual timelines - introducing reflex reaction into architecture

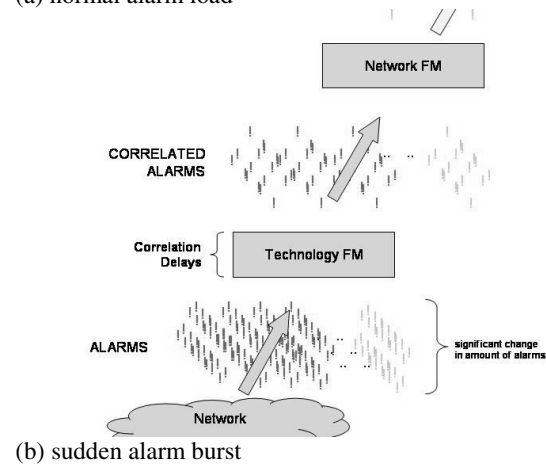
One advantage of this is the potential to utilize the details on the TT as external *symptoms* and correlating these with the internal symptoms (the alarms) to assist with the diagnosis of the fault(s).

The addition of a reflex signal does not only benefit the top layer in the architecture. In the scenario that the health indicator pulse is changed based on the sudden change in the number or rate of alarms arriving with a specific technology fault manager, the cross-technology network fault manager will be alerted via the pulse signal almost immediately and have the correlation delay time to prepare for the likely oncoming alarm burst (e.g. self-configure by dynamically allocating resources from less active processes to the relevant technology process).

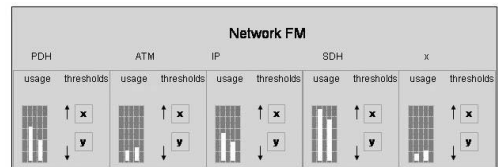
Figure 5 (a) & (b) depicts a sudden change in alarms due to a fault in the network. In the example the technology being affected is SDH. Figure 5 (c) highlights that this sudden burst in alarms may result in an overload in the process handling SDH.



(a) normal alarm load



(b) sudden alarm burst



(c) process handling risk overload

Figure 5 alarm burst

Figure 6 (a) highlights that through reflex mechanism (pulse monitor) at the instant the domain FM becomes aware of an alarm flood it alerts the xTech N/W FM. In effect the xTech N/W FM has advance warning (the tech FM correlation delay timeframe) to poll for spare capacity and self-configure to avoid overload (Figure 6 (b)). It may be the case that in some instances the domain FM may 'correlate away' the flood of alarms so that the burst is not seen at the xTech N/W FM. A change in the pulse signal would notify the xTech N/W FM that the danger had passed and it may reestablish its standard configuration.

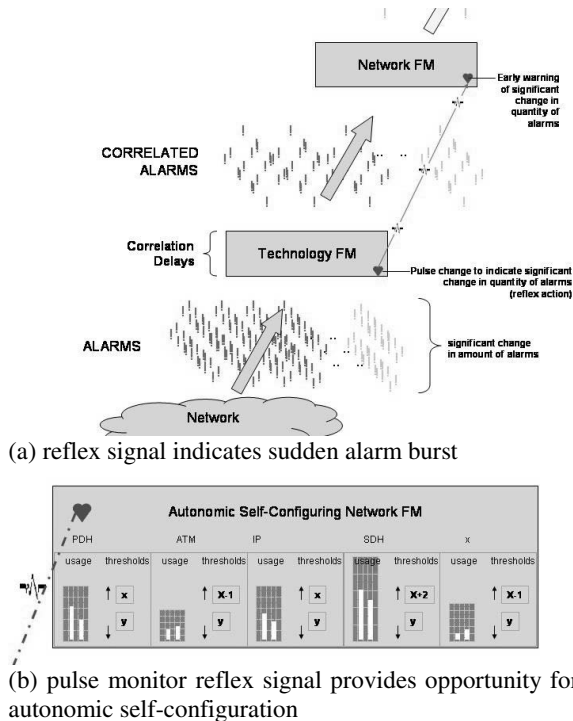


Figure 6 new autonomic options

To address this unhelpful and even counterproductive measure in certain circumstances would require that over time key characteristic differences between alarm floods that determine what will result in a flood or not a flood at the xTech N/W FM will be identified and used to determine any change in the pulse – i.e. the pulse monitor in the longer term will need to be self-adaptive.

Another aspect of this dual approach is that it may assist with cross domain alarm correlation since the pulse may give an immediate indication as to the technology where the fault originated.

5. Related Work

In order for the Autonomic Computing initiative to succeed it will need to draw on and collaborate with and within many domains such as systems and software engineering [15], Artificial Intelligence [16], dependable systems [17], fault-tolerant computing, active middle-ware [18] and so on.

As with most new initiatives the majority of early results tend to focus on ways to architect new software and systems [6]. Research is being carried out on retrofitting autonomic capabilities into legacy systems externally without any need to understand, modify or recompile the code [6][7].

Recently IBM and CISCO released a white paper on a problem determination framework [9] which fits with many of the emerging tools from IBM [19] such as the Log and Trace Tool that correlates event logs from legacy systems to identify patterns. These can be used to facilitate automation or help in debugging. The Tivoli Autonomic Monitoring Engine essentially provides server level correlation of multiple IT systems to assist with root cause analysis and automated corrective action. While the ABLE rules engine can be used for more complex analysis, 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.

6. Conclusion

Autonomic computing is gaining ground as a viable holistic approach to computer-based system development that aims to bring a new level of automation and dependability to systems through self-healing, self-optimizing, self-configuring and self-protection functions.

An EU brainstorming workshop discussing novel communication paradigms for 2020 identified "Autonomic Communications" as an area of long-term research, evolution of what is currently considered under the name of self-organising networks, and which includes developments in ad-hoc, cooperative wireless networks and wireless sensors networks [3].

The term Autonomic Communications is an analogy to IBM's Autonomic Computing initiative with specific focus from the communications research and development community.

With the emerging convergence of computing and telecommunications the Engineering of Autonomic Systems, incorporating autonomic computing and autonomic communications, will become even more critical.

This paper has discussed the concept of incorporating a pulse monitor to provide a reflex reaction for indicating the 'health' of the network as seen by the monitoring manager, giving advance warning to northbound managers and thus opening new options for engineering autonomic capabilities into the fault managing architecture.

In the UFM domains the pulse mechanism may be piggybacked on the existing HBM to assist in retrofitting autonomic capabilities into the system. Yet this is only a secondary point and a compromise to assist with cost effectiveness in retrofitting, which may not offer the full advantages of the concept. For instance, the heartbeat

timeframe is static and if it may not be changed, instead of an immediate reflex alert from the pulse, the indicator is on average delayed by half heartbeat timeframe to be sent. The primary point being made in this paper is the need for dynamics within autonomic responses and multiple loops of control; some *slow* and *precise*, others *fast* and possibly *imprecise*

Although the changes necessary to move from reflex to autonomic system are not necessarily huge, they will require significant effort on the part of Telcos, in association with the industry, to intellectually decide to do it, and then to invest the time, resource and money to changing existing OSS.

Acknowledgements

This work was supported through a BT Exact Short Term Research Fellowship (2003) – “xACT: Autonomic Computing and Telecommunications”. The wider context of the research is undertaken within the Centre for Software Process Technologies, which is supported by the EU Programme for Peace and Reconciliation in Northern Ireland and the Border Region of Ireland (PEACE II).

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] Y. Tohma, "Fault Tolerance in Autonomic Computing Environment", Proceedings of the 2002 Pacific Rim International Symposium on Dependable Computing (PRDC'02), IEEE Computer Society, 2002.
- [3] EU IST FET, New Communication Paradigms for 2020, July & Sept 2003.
- [4] Sterritt R, (Aug 2003) "Pulse Monitoring: Extending the Health-check for the Autonomic GRID", Proceedings of the IEEE Workshop on Autonomic Computing Principles and Architectures (AUCOPA' 2003) at IEEE Int. Conf. Industrial Informatics (INDIN 2003), Banff, Alberta, Canada, 22-23 August 2003
- [5] Keith Brodrick, BT's Unified Fault Management Architecture, NIACT 2002.
- [6] G. Kaiser, J. Parekh, P. Gross, G. Valetto, "Retrofitting Autonomic Capabilities onto Legacy Systems", *Columbia TR UCUS-026-03*, 2003.
- [7] G. Kaiser, J. Parekh, P. Gross, G. Valetto, "Kinesthetics eXtreme: An External Infrastructure for Monitoring Distributed Legacy Systems." *Autonomic Computing Workshop – IEEE Fifth Annual International Active Middleware Workshop*, Seattle, USA, June 2003.
- [8] IBM, Automating problem determination: A first step toward self-healing computing systems, white paper October 2003.
- [9] IBM & Cisco Systems, Adaptive Services Framework white paper, October 2003.
- [10] A. Ganek, "Keynote: Autonomic Computing: Implementing the Vision", *Autonomic Computing Workshop – IEEE Fifth Annual International Active Middleware Workshop*, Seattle, USA, June 2003.
- [11] Sam Lightstone "Keynote: Towards Benchmarking - Autonomic Computing Maturity", Workshop on Autonomic Computing Principles and Architectures (AUCOPA' 2003), at INDIN 2003 - First IEEE Conference on Industrial Informatics, Banff Canada, August 2003.
- [12] T. Bapty, S. Neema, S. Nordstorm, S. Shetty, D. Vashishtha, J. Overdorf, P. Sheldon, "Modeling and Generation Tools for Large-Scale, Real-Time Embedded Systems", 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, Huntsville, Alabama, USA, 7-10th April 2003, pp11-16.
- [13] P. Stelling, I. Foster, C. Kesselman, C. Lee, G. v. Laszewski, "A Fault Detection Service for Wide Area Distributed Computations", Proceedings of the 7th IEEE Symposium on High Performance Distributed Computing, 1998
- [14] 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
- [15] EASe 2004, Workshop on the Engineering of Autonomic Systems, IEEE ECBS 2004, Brno, Czech Rep., May 2004.
- [16] IJCAI Workshop, "AI and Autonomic Computing: Developing a Research Agenda for Self-Managing Computer Systems, Acapulco, Mexico, August 10, 2003, <http://www.research.ibm.com/ACworkshop>
- [17] R Sterritt, DW Bustard, "Autonomic Computing-a Means of Achieving Dependability?", Proceedings of IEEE International Conference on the Engineering of Computer Based Systems (ECBS'03), Huntsville, Alabama, USA, April 7-11 2003, pp 247-251.
- [18] Autonomic Computing Workshop, 5th Int. Workshop on Active Middleware Services (AMS 2003), Seattle, WA, USA, June 2003
- [19] IBM, alphaworks Autonomic Computing site, <http://www.alphaworks.ibm.com/autonomic>