Proceedings of the
IJCAI-05 Workshop on

AI and Autonomic Communications

Developing a research agenda for Self-Managing Networks and the Knowledge Plane

Edinburgh, Scotland
31st July 2005

Editors
Roy Sterritt, Simon Dobson, Mikhail Smirnov

Organising Committee

Roy Sterritt
University of Ulster, Northern Ireland

Simon Dobson
University College Dublin, Ireland

Mikhail Smirnov
Fraunhofer FOKUS, Germany

Programme Committee

David Clark, MIT, USA
Simon Dobson, University College Dublin, Ireland
Dave Lewis, Trinity College Dublin, Ireland
Maurice Mulvenna, University of Ulster, Northern Ireland
Guy Pujolle, l'Université Paris 6, France
Fabrice Saffre, British Telecom, UK
Mikhail Smirnov, Fraunhofer FOKUS, Germany
Roy Sterritt, University of Ulster, Northern Ireland

Support Acknowledgements

ACCA: Autonomic Communication: Coordination Action (AST-6475) funded through EU Open Strategic Objective: Future and Emerging Technologies (FET 2.3.4.1)

University of Ulster's Centre for Software Process Technologies (CSPT) funded by Invest NI through the Centres of Excellence Programme, under the EU Peace II initiative.

Contents

Efficient Overlay Networks for Autonomic Communication

Fabrice Saffre
British Telecommunications plc.
Pervasive ICT Research Centre
Adastral Park, Orion 1st floor pp 12, Martlesham Heath IP5 3RE, U.K.
fabrice.saffre@bt.com

Abstract

Autonomic Communication refers to the concept of a self-configuring, extensible and dynamic communication infrastructure. We argue that the realization of such a system will depend on the development of a decentralized overlay network, able to take account of locally available implicit knowledge of system state and usage patterns.

Almost by definition, autonomic (i.e. self-configuring, self-repairing…) communication will simultaneously require andpromote (via positive feedback and viral intake) mobility of content and software components. Indeed, successful implementation of the autonomic "vision" will likely involve continuous migration of huge numbers of various "modules" between the many devices attempting to respond to a dynamic demand in a context-sensitive fashion.

As soon as the mapping between one piece of equipment and a predefined set of functionalities disappears, so does the ability to define local requirements upfront, which in turn calls for software "agility" (i.e. mobile, self-contained modules that can be installed and discarded as needed).

Furthermore, the plasticity and unpredictability of the network environment (different protocols and capabilities, mobile and static devices, wireless etc.) dictate that service discovery and content delivery will likely be supported by P2P interactions, as opposed to the centralized, more manageable but far less adaptive "client-server" model.

Together, all these aspects suggest that meeting the needs of autonomic communication will require vast amounts of bandwidth. Yet ultimately, the success of the paradigm will depend on the reliability of the supported services: if the necessary network resources are not available or decentralized management proves incapable of discovering and mobilizing them, autonomic communication will simply fail to gain public trust and support.

There are however many indications that enough bandwidth is available (or soon will be, thanks to the on-going proliferation of high-bandwidth connection opportunities like, e.g., WiFi, Bluetooth or ZigBee). It will still be contended though, and it is a well-known fact that failing to consider network resource management issues when designing local clients can result in extreme waste of capacity and foster poor performance, even in an over-provisioned environment (as dramatically illustrated by P2P file sharing for example). So a major challenge is to create a "network-friendly" autonomic communication infrastructure, one that will avoid unnecessarily flooding information along already saturated links and be able to identify ways of reducing bandwidth consumption without compromising QoS. This is a complex problem, as the economical solution may alternate between extremes like finding the shortest path to the overprovisioned network core or, on the contrary, relying exclusively on local exchanges at the periphery.

Overlay networks currently look like one of the best candidate paradigms to meet this challenge. At first sight, it may seem paradoxical that a design philosophy pledged to ignore the detailed constraints of the supporting physical infrastructure may help reduce the waste of bandwidth. However, this apparent contradiction finds its origin in a deep misunderstanding of overlay networks' potential ability to organize themselves so as to map the web of successful interactions between nodes. This bears the possibility of taking into account implicit "knowledge" about system state and activity patterns, and of continuously adapting to the changing "landscape" of available network resources.

In a decentralized and dynamic environment, useful regularities tend to be unpredictable and large amounts of unnecessary traffic may be generated if local clients rely heavily on resource/service rediscovery every time they handle a new request. Admittedly, most existing protocols already try to address that problem in one way or the other (e.g. with "super-peers" or caches), but it can be argued that a more fundamental and principled approach to generating efficient co-operative overlay networks is needed. For example, member devices should be continuously questioning the value of their "virtual" links, from both a "typical availability" and "ability to meet my needs" point of view (much in the same way as we humans manage our network of contacts based on a combination of factors).

Because growing and maintaining self-organized overlay networks will have to rely on autonomous decision-making by individual nodes, we believe that AI techniques, both "well-established" (reinforcement learning, fuzzy logic, Bayesian networks etc.) and more "speculative" (e.g. collective or swarm intelligence), are likely to play an important role in designing autonomic communication elements.

Reflex Unified Fault Management Architecture – Lessons for the Knowledge Plane?

Roy Sterritt
School of Computing & Mathematics
University of Ulster at Jordanstown
Northern Ireland
r.sterritt@ulster.ac.uk

Dave Bustard
School of Computing & Information Engineering
University of Ulster at Coleraine
Northern Ireland
dw.bustard@ulster.ac.uk

Abstract

As the range of communication services and consumer expectations continue to grow, the demands on telecommunication organizations to find ways to make their systems more adaptable and flexible, while remaining dependable, becomes critical. Autonomic communications is an emerging strategic approach for addressing these needs. This paper discusses the unified fault management architecture and the proposal to add autonomicity through a 'reflex-healing' dual strategy. The autonomic reflex unified fault management architecture has goals in common with the perceived new construct of autonomic communications, the knowledge plane.

1 Introduction

Autonomic systems are essentially self-managing systems, based on the biological metaphor of the autonomic nervous system [Horn, 2001]. They are typically characterized by having one or more of four standard sub-properties: self-healing, self-protecting, self-optimizing and self-configuring. *Autonomic communications* focuses on the use of such ideas in the design of robust networks.

Extending the biological metaphor, an earlier paper introduced the concept of an autonomic *reflex reaction* mechanism [Sterritt, 2003b] as a two-stage approach to fault handling. In the first 'reaction' stage, the system responds quickly to protect itself from a perceived threat. In the second 'healing' stage, the system assesses the situation, considering any damage done, and initiating repair and recovery as necessary.

This reflex reaction concept may be realized in a telecommunications fault management system architecture to assist in achieving autonomicity.

This paper first discusses Autonomic Communications and the Knowledge Plane. It then describes the existing fault management architecture at BT [Brodrick, 2002], highlighting some of its complexities and the proposed autonomic extensions to the fault management architecture. The paper concludes with consideration of lessons from the unified fault management architecture for the perceived knowledge plane.

2 Autonomic Communications

The research discussed in this paper was motivated by a study undertaken in British Telecom (BT) in 2003 [Sterritt, 2003a] on *Autonomic Computing and Telecommunications*. Since then, *autonomic communications* has emerged as a branch of research in its own right, and was formally announced as a European Union funding initiative for 2005 and beyond [Sestini, 2004].

This initiative began when an European Union Future Emerging Technologies (EU FET) brainstorming workshop in July 2003 to discuss novel communication paradigms for 2020 identified 'autonomic communications' as one potential important area for future research and development [EU IST FET, 2003]. This was interpreted as further work on self-organizing networks, and which includes developments in ad-hoc, cooperative wireless networks and wireless sensors networks [EU IST FET, 2003] but was also undoubtedly a reflection of the growing influence of autonomic computing advocated by IBM [Horn, 2001]. In effect, autonomic communications has the same motivators as the autonomic computing concept with particular focus on the communications research and development community. Goals highlighted at this initial workshop were to understand how an autonomic network element's behaviours are learned, influenced or changed, and how in turn, these affect other elements, groups and networks. The ability to adapt the behavior of the elements was considered particularly important in relation to drastic changes in the environment, such as technical developments or new economic models [EU IST FET, 2003].

At the heart of autonomic communications are *selfware* principles and technologies that will create the autonomic network. They borrow largely from autonomous distributed systems research and non-conventional networking (ad hoc, sensor, peer-to-peer, group communications, active networks and so forth), among others [Smirnov and Popescu-Zeletin, 2003]. In addition, a new construct, a knowledge plane, has been identified as necessary to act as a pervasive system element within the network to build and

maintain high level models of the network. These indicate what the network is supposed to do to provide communication services and advice to other elements in the network [Clark *et al*, 2003]. It is generally considered that this knowledge plane will rely on the tools of AI and cognitive systems to meet the uncertainties and complexities of this goal, rather than traditional algorithmic approaches [Clark *et al*, 2003],[Agosta and Crosby, 2003]. It will also work in coordination with the management plane and data planes.

The second EU FET consultation meeting on 'novel communication paradigms for 2020' in March 2004 was now almost solely focused on the subject of autonomic communications [Smirnov, 2004b], [EU IST FET, 2004], [Sestini, 2004], [Smirnov, 2004a] and identified the following research challenges:

- Telecommunication strategy towards Autonomic Communications
- Zero-effort deployment ('spray' deployment)
- Programming of self-organisation including architectural programmability
- Self-Management in Autonomic Communication
- Autonomic Communication contribution to Network Information Theory
- Security and Protection
- Coordination and Intelligence in Service Provisioning for Autonomic Communication
- Behaviour knowledge and knowledge execution in Autonomic Communication

Specifically, in relation to the telecommunications strategy, the workshop highlighted the need for highly dynamic networks and communication services, with more intelligent support in the heart of the Internet, in addition to that currently available at its endpoints.

The research community has labeled 'Situated and Autonomic Communications' as a year 2020 paradigm which suggests the considerable amount of research and development that will be needed to achieve the grand vision of autonomicity.

3 The Knowledge Plane

The Internet is a hugh success which has become the pervasive system of today. Its success lies in its generality and heterogeneity, the combination of a simple transparent network (the data plane) with rich end-system functionality. Yet the down sides become apparent when something fails along with costs through high management overhead with large manual configuration, diagnosis and design [Clark *et al*, 2003].

The simple and transparent core with intelligence at the edges essentially means the network carries data without knowing what the data is or what its purpose is, as such when a combination of events occur that prevent the data from getting through the edge may recognize that there is a problem, but the core has no idea what should be happening [Clark *et al*, 2003].

As such it has been recognized that a new construct is required for next generation networks, a pervasive system within the network that builds and maintains high level models of what the networks is supposed to do in order to provide services and advice to other network elements [Clark *et al*, 2003].

It is perceived that the knowledge plane will be built on top of the transparent network as a global, decentralized network overlay that aggregates global information, observations, assertions, requirements, constraints and goals [Clark *et al*, 2003]. In terms of fault detection and isolation it would facilitate cross-correlation assessment with diagnoses traveling up to the KP and conclusions being passed down.

4 The Management Plane: Reflex Unified Fault Management

4.1 UFM Architecture

Wide-area national and global telecommunication systems, initially designed for voice traffic, provide the backbone bandwidth capabilities necessary for Internet traffic. To ensure adequate quality of service they are built with substantial management control systems and extensive redundancy, often based on a survivable network architecture (SNA) within the data plane, which essentially uses ring structures, with inbuilt protected capacity that is only used when part of the network fails. Components are protected individually, which achieves robustness in the presence of faults but makes fault detection more complex and difficult. An individual fault occurring in one component may affect the components with which it interacts, which can then in turn raise other alarms. The net result is often a cascade of alarm events, reported to an *element controller*.

The behaviour of the alarms is typically so complex that it appears non-deterministic [Bouloutas, 1994], [Sterritt, 2002]0. Consequently, it can be very difficult to isolate the root cause of the problem. Failures in the network are unavoidable but quick detection, identification and repair is essential to ensure an adequate service. Central to achieving this objective is the rapid analysis, or *correlation* of alarm events to identify their interdependencies. At one extreme, this might be entirely the operator's responsibility, achieved by performing an analysis on the full set of events reported. Ideally, however, the process should be as automated, or *self-managing*, as possible.

The situation is further complicated by the need for telecommunication systems to operate in a heterogeneous environment, supporting a wide range of communication services across a range of technologies, including SDH/SONET, PDH, ATM, ADSL and IP. An additional difficultly is that these technologies are inter-connected, in that, for example, SDH frames may be carrying ATM traffic.

A simplified view of the Unified Fault Management (UFM) system architecture that has evolved at BT to take account of these business realities is shown in Figure 1 [Brodrick, 2002].

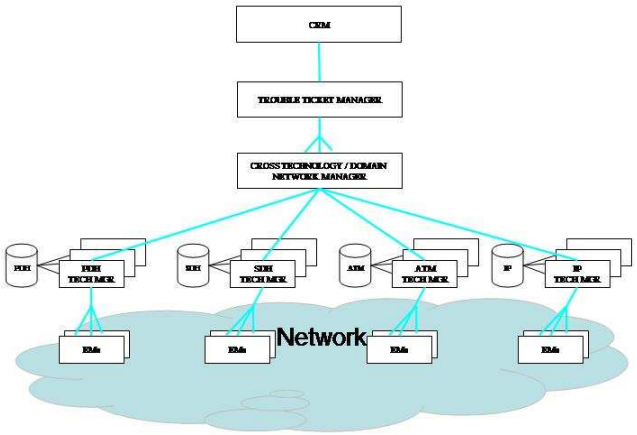


Figure 1 Simplified view of the Unified Fault Management System Architecture

The architecture elements in the bottom layer, the *physical network*, tend to be outside a telecommunication organization's design control, being supplied by third party vendors. Consequently, the potential to include autonomic functions in this layer is limited, due to the element specific interfaces. The obvious solution is for suppliers to recognize this issue and collaborate to agree communication standards that support autonomic behavior [Ganek, 2003], 0[Lightstone, 2003]. This would then support initiatives to refit autonomic computing into legacy systems [Kaiser *et al*, 2003a], 0[Kaiser *et al*, 2003b]. IBM and Cisco's agreement on problem determination [IBM, 2003], [IBM & Cisco, 2003] is a sign of encouraging progress in the right direction.

The next layer up in the UFM architecture reflects the variety of technologies a large telecommunications company is required to manage and the complexity that results. Each technology within the network has its own specific technology *fault manager* (also referred to as a domain fault manager). The individual element mangers within the network pass the alarms and event messages up to the relevant manager for their technology. Since SDH frames, for example, may be carrying ATM packets, which may even include IP traffic and so on, a fault in one technology, such as SDH, may affect the other dependent technology domains. An individual technology domain will not be able to determine a fault that extends across different domains as it has insufficient knowledge of their services and protocols. 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 is available 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 a 'trouble ticket' and 'task force' management for remedy.

By its very nature, root-cause analysis introduces delays at each management layer in the architecture in handling alarm and event correlation. This is because time has to be allowed for the inter-related alarms from different sources to arrive at the correlating manager before analysis can begin. There is then further delay in obtaining information from the components affected. When this process is aggregated across successive layers it may take as long as 10-15 minutes from the fault arising to it being reported at the service level (CRM) under extreme fault conditions. By that time, network users may already have reported the problem directly, which is clearly undesirable.

The timeline for the existing BT fault management architecture (Figure 2) indicates that root cause analysis has inherent inbuilt delays to allow effective alarm correlation to occur. Figure 2 indicates how, under fault conditions, there is a time gap between the impact of a fault on customers and knowledge of that fault arriving at the service level. As indicated earlier, in the BT network this may be in excess of 10 minutes under major network incident conditions.

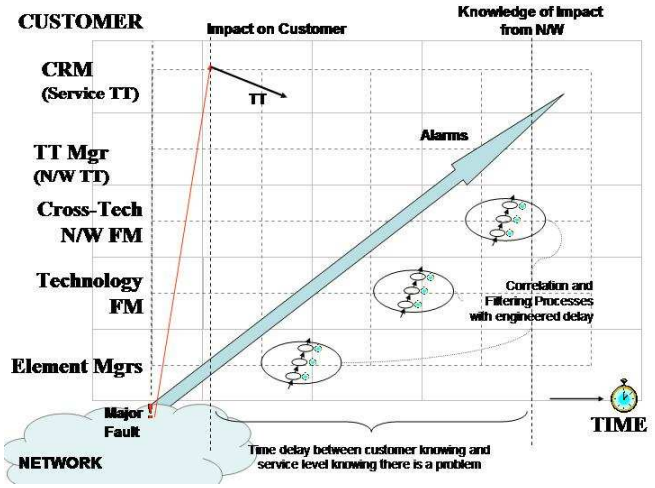


Figure 2 Single timeline - alarm progression through management hierarchy

Other, faster ways of communicating information need to be found. This applies to communication both from the network layer up to the customer (northbound) and from the customer down to the network layer (southbound). Seeing the network as an autonomic system can help identify a suitable solution.

4.2 Reflex UFM Architecture

A proposed extension to the Unified Fault Management (UFM) system architecture is to include autonomic behavior based on the reflex pulse monitoring concept is presented in Figure 3. Links are shown from the technology-specific fault managers to the cross technology fault manger, and on up to the managers for trouble tickets at the network and service layers (CRM).

This architecture is conceptually a simple extension of the current structure presented in Figure 1. In practice, the differences can be even smaller than is suggested because heartbeat monitors already exist between the management components, as a safeguard against their failure. The proposed extension is therefore to add health indicators to these existing heartbeats, to create 'pulses'.

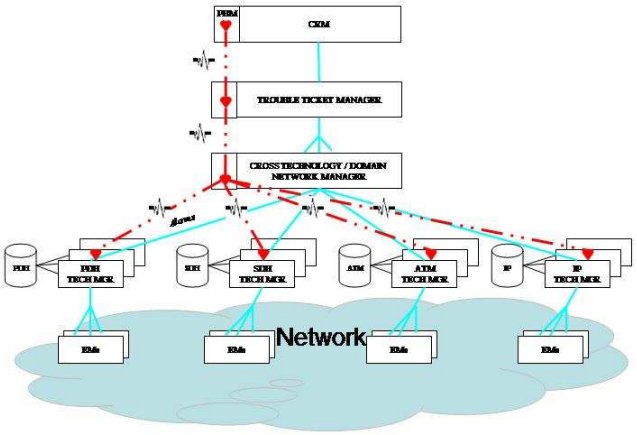


Figure 3 Autonomic Reflex UFM System Architecture

In practical terms, it is important to keep the health indicator information sufficiently sparse to ensure that the reflex reaction is not compromised. In particular, this means ensuring that information is communicated rapidly through the network, implying that the cost of processing such information must be low. This cost is a combination of the total volume of information transmitted and the associated time to compose and analyze the information across the interactions involved.

The pulse has two mechanisms to indicate health information: a *health indicator summary* contained within the heartbeat and an *urgency level*; this may also be contained within the heartbeat or indicated by the rate at which the beat is sent. The heartbeat monitor sends a beat at a constant interval and under normal conditions the pulse monitor would do likewise. On encountering circumstances affecting the network, however, such as a significant rise in alarms reported faults, the pulse rate can increase to warn of the problem.

This dynamic pulse rate is consistent with the biological metaphor, but it is also desirable to ensure that information is reported more frequently when operating conditions become difficult. To achieve the reflex reaction a signal should be sent immediately, implying a change in the pulse rate, which should then stay at a higher level, reporting state information, until the situation is resolved.

The pulse mechanism has been described as *conceptually* extending the heartbeat monitor since *physically* there is still a role for it and the two are likely to co-exist in the network. For instance, at the granularly level where the component guarded by a heartbeat monitor has limited low-level functionality it only requires a heartbeat to convey health information.

The pulse monitor is conceived for situations, such as within system monitors, where a heartbeat monitor is used but the component is also in a position to supply an assessment of health conditions to other parts of the network. This may be achieved by extending the heartbeat to a pulse or providing a separate communication link.

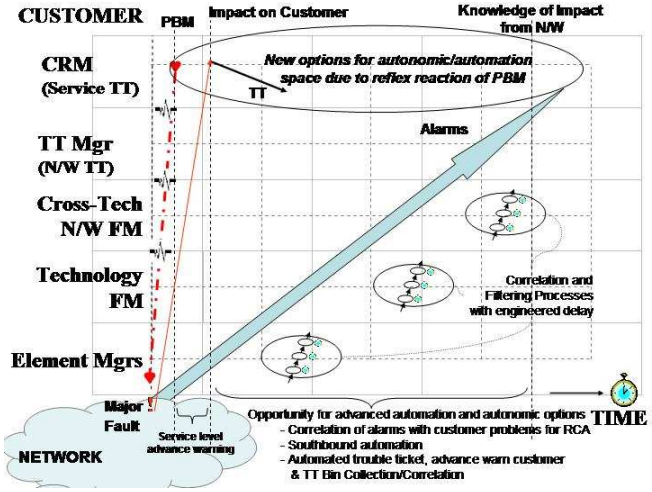


Figure 4 Dual timelines - introducing reflex reaction into architecture

Figure 4 shows the potential of a reflex reaction within the architecture. Firstly the reflex mechanism (pulse monitor) informs the service level of a major fault ahead of any direct customer reports. This then provides new options for southbound automation and autonomic behavior at this level: for instance through early warning of a major fault, any customer trouble tickets arriving subsequently can be

linked to that fault. One advantage of this approach is the potential to use the details on the trouble ticket as external *symptoms* and correlate these with the internal symptoms (the alarms) to assist with the diagnosis of the fault(s).

The addition of a reflex signal certainly benefits the top layer in the architecture but also adds value to lower levels. For example, in the case where the health indicator pulse changes because of a 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 depicts a sudden change in alarms due to a fault in the network. In the example the technology being affected is SDH, it highlights that this sudden burst in alarms may result in an overload in the process handling SDH.

Figure 5 also indicates that through the reflex mechanism (pulse monitor) when the domain fault manager (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.

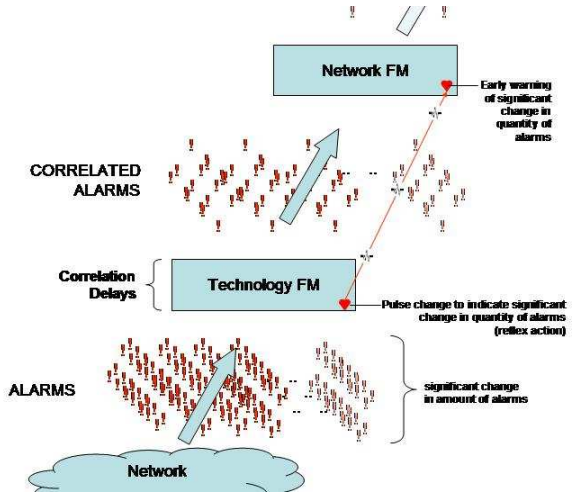


Figure 5 new autonomic options - reflex signal indicates sudden alarm burst

In some cases 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. It is obviously wasteful, however, to take preparatory actions anticipating an alarm flood, which subsequently does not appear. This implies a need for the fault managers at the domain and cross-technology levels to learn from such false-positives situations and either avoid escalating the perceived problem in future or indicate that there may be some doubt about its severity. In effect, this requires the pulse monitor to become 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 of the technology where the fault originated, for example a pulse urgency indication first registered from an SDH fault manager may result in the subsequent connected ATM flood of alarms having less priority since the root cause lies in the SDH domain.

5 Discussion and Lessons for the KP?

The knowledge plane is a proposed third abstraction in the emerging research area of autonomic communications, adding to the existing data and control/management planes. In their vision paper, the proponents of the knowledge plane discuss broadly how machine learning algorithms can be applied to garner knowledge and increase the self-awareness of the network. How the knowledge plane will be achieved is an open research area.

The knowledge plane sits in a different space than the data and management planes; it does not move data directly nor responsible for such management functions as accounts [Clark et al, 2003]. Yet it has been identified that one of the requirements from the KP is *Cross-Domain and Multi-Domain Reasoning* [Clark et al, 2003]. The cross-technology fault manager within the UFM attempts to provide a single overall view of the network and build knowledge (in terms of rules) that interlinks the behavior of the different technologies involved. From this perspective it shares part of the goals of the Knowledge Plane.

This approach to dealing with cross-domain technologies within large scale telecommunications was novel and a success [Brodrick, 2002]. Challenges identified from the large-scale operational approach of the UFM are dealing with the complexity and engineering/learning the knowledge for within the *cross domain fault manager* to deal with root cause analysis [Sterritt 2002], [Sterritt and Bustard, 2002a], [Sterritt and Bustard, 2002b].

Another challenge identified was the need for dynamic speeds through the architecture as discussed here and proposed through the reflex reaction implemented by the pulse monitors within the R-UFM, together with the extensions to management components to react to pulse changes, helps provide a base for the development of such services envisaged within autonomic communications [Sterritt 2003], [Sterritt *et al*, 2004], [Sterritt *et al*, 2005].

These challenges may highlight some lessons for the perceived knowledge plane.

In terms of retrofitting autonomicity into the UFM domains the pulse mechanism may be piggybacked on the existing heartbeat monitor. 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. The essential 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 to achieve the necessary level of self-management.

6 Conclusions

Autonomic communications is gaining ground as the paradigm for next generation networking. It aims to bring a new level of automation and communication services through self-managing properties, in common with Autonomic Computing, such as self-healing, self-optimizing, self-configuring and self-protecting.

With the emerging convergence of computing and telecommunications the engineering of autonomic systems, incorporating autonomic computing and autonomic communications, will become even more critical. The longer-term aspiration is to implement network management in a way that supports automatic decision-making based on the specification of high-level business policies. This will require additional research and development in a number of contributing areas. In particular, as well as architectural changes to support reflex mechanisms, there is a need for background reflection on the effect that such changes have on network behavior, possibly using machine learning strategies. An important aspect of this work is the provision of additional material to assist decision-making. There is still a need for alarm correlation but cross-checking that with reflex information should help confirm or rule out some of the options being considered.

This paper has discussed the unified fault management architecture (UFM) and the concept of incorporating a pulse monitor to provide this 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.

Acknowledgments

This work was supported through a British Telecom Short Term Research Fellowship (2003) – "xACT: Autonomic Computing and Telecommunications". The wider context and follow-up of the research is undertaken within the Computer Science Research Institute (CSRI) and 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

[Agosta and Crosby, 2003] JM Agosta, S Crosby, "Network integrity by inference in distributed systems", NIPS Workshop on Robust Communication Dynamics in Complex Networks, 2003

[AMS, 2003] Autonomic Computing Workshop, 5th Int. Workshop on Active Middleware Services (AMS 2003), Seattle, WA, USA, June 2003

[Bapty et al, 2003] Bapty, T., Neema, S., Nordstorm, S., Shetty, S., Vashishtha, D., Overdorf, J., Sheldon, P., "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.

[Bouloutas, 1994] Bouloutas, A. T., S. Calo, A. Finkel, 1994. Alarm Correlation and Fault Identification in Communication Networks, IEEE Transactions on Communication, Vol. 42, No 2/3/4.

[Brodrick, 2002] Brodrick, K., BT's Unified Fault Management Architecture, NIACT 2002.

[Clark et al, 2003] D Clark, C Partridge, JC Ramming, JT Wroclawski, "A Knowledge Plane for the Internet", Proc. Applications, technologies, architectures, and protocols for computer communication, Karlsruhe, ACM SIGCOMM 2003

[EASe, 2004] EASe 2004, Workshop on the Engineering of Autonomic Systems, IEEE ECBS 2004, Brno, Czech Rep., May 2004.

[EU IST FET, 2003] EU IST FET, "New Communication Paradigms for 2020", brain storming meeting July 2003, Brussels, Belgium, (report published Sept 2003)

[EU IST FET, 2004] EU IST FET, "New Communication Paradigms for 2020", Consultation meeting 3-4th March 2004, Brussels, Belgium.

[Ganek, 2003] Ganek, A., "Keynote: Autonomic Computing: Implementing the Vision", Autonomic Computing Workshop – IEEE Fifth Annual International Active Middleware Workshop, Seattle, USA, June 2003.

[Horn, 2001] 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, Scotsdale, AR. (available http://www.research.ibm.com/autonomic/), 2001

[IBM & Cisco, 2003] IBM & Cisco Systems, Adaptive Services Framework white paper, October 2003.

[IBM, 2003] IBM, Automating problem determination: A first step toward self-healing computing systems, white paper October 2003.

[IBM, 2004] IBM, alphaworks Autonomic Computing site, http://www.alphaworks.ibm.com/autonomic

[IJCAI AI+AComp, 2003] 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

[Kaiser et al, 2003a] Kaiser, G., Parekh, J., Gross, P., Valetto, G., "Retrofitting Autonomic Capabilities onto Legacy Systems", Columbia TR CUCS-026-03, 2003.

[Kaiser et al, 2003b] Kaiser, G., Parekh, J., Gross, P., Valetto, G., "Kinesthetics eXtreme: An External Infrastructure for Monitoring Distributed Legacy Systems." Autonomic Computing Workshop – IEEE Fifth Annual International Active Middleware Workshop, Seattle, USA, June 2003.

[Lightstone, 2003] Lightstone S., "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.

[Patterson et al, 2002] Patterson, D.A., Brown, A., Broadwell, P., Candea, G. , Chen, M., Cutler, J., Enriquez, P., Fox, A., Kiciman, E., Merzbacher, M., Oppenhiemer, D., Sastry, N., Tetzlaff, W., Traupman, J., Treuhaft, N., 2002, Recovery-Oriented Computing (ROC): Motivation, Definition, Techniques, and Case Studies, U.C. Berkeley Computer Science Technical Report, UCB//CSD-02-1175, University of California, Berkeley, March 15.

[Sestini, 2004] F Sestini, 'Situated and Autonomic Communications', EU IST FET New Communication Paradigms for 2020 Consultation meeting, Brussels, March 2004.

[Sestini, 2004] Sestini F., IST-FET proactive initiative on "Situated and Autonomic Communications" 4th IST call, deadline December 2004 - March 2005.

[Smirnov and Popescu-Zeletin, 2003] M Smirnov, R Popescu-Zeletin, "Autonomic Communication", presentation EU IST FET brainstorming meeting Communication Paradigms for 2020, Brussels, July 2003

[Smirnov, 2004a] M Smirnov, 'Area: Autonomic Communications', EU IST FET New Communication Paradigms for 2020 Consultation meeting, Brussels, Belgium. (ver. 02), March 2004.

[Smirnov, 2004b] M Smirnov, Managing Internet complexity in Autonomic Communication, presentation EU IST FET consultation meeting Communication Paradigms for 2020, Brussels, March 2004

[Stelling et al, 1998] Stelling, P., Foster, I., Kesselman, C., Lee, C., v. Laszewski, G., "A Fault Detection Service for Wide Area Distributed Computations", Proceedings of the 7 th IEEE Symposium on High Performance Distributed Computing, 1998

[Sterritt and Bustard, 2002a] Sterritt R, Bustard DW, "Fusing Hard and Soft Computing for Fault Management in Telecommunications Systems", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews , Vol. 32, No. 2, IEEE, Pages 92-98

[Sterritt and Bustard, 2002b] Sterritt R, Bustard DW, "Practical Intelligent Support for Rule Discovery in Fault Management Systems", Cybernetics & Systems: An International Journal, Vol. 33, No. 6, Taylor & Francis, ISSN 0196-9722 (Paper) 1087-6553 (Online), Pages 579-601

[Sterritt and Bustard, 2003] Sterritt, R., Bustard, D.W., "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.

[Sterritt et al, 2004] 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 2004) at the 11th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2004), Brno, Czech Republic, 24-27 May, Pages 449-455

[Sterritt et al, 2005] Sterritt R, Bustard DW, Gunning D, Henning P, "Autonomic communications and the reflex unified fault management architecture", Advanced Engineering Informatics, Elsevier, in press.

[Sterritt, 2002] Sterritt R, "Facing Fault Management as It Is, Aiming for What You Would Like It to Be", Proceedings of Soft-Ware 2002: Computing in an Imperfect World, Belfast, Northern Ireland, April 2002, in "LNCS 2311" (Edited by Bustard D., Liu W., Sterritt R.), Springer-Verlag (Berlin), ISBN 3-540-43481-X, Pages 31-45

[Sterritt, 2003a] Sterritt, R., “xACT: Autonomic Computing and Telecommunications”, BT Exact Research Fellowship, 2003.

[Sterritt, 2003b] Sterritt R, “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

[Sterritt, 2004] Sterritt R, "Autonomic Networks: Engineering the Self-Healing Property", Engineering Applications of Artificial Intelligence, Vol. 17, No. 7, Elsevier, ISSN 0952-1976, Pages 727-739

[Wyatt, 1998] Wyatt, J., Hotz, H., Sherwood, R., Szijjaro, J., Sue, M., “Beacon Monitor Operations on the Deep Space One Mission”, 5th Int. Sym. AI, Robotics and Automation in Space, Tokyo, Japan, 1998

Categorization and Modelling of Quality in Context Information

M.A. Razzaque, Simon Dobson and Paddy Nixon
Systems Research Group
School of Computer Science and Informatics
University College, Dublin IE
abdur.razzaque@ucd.ie, simon.dobson@ucd.ie, paddy.nixon@ucd.ie

Abstract

Pervasive Computing environments are dynamic and heterogeneous. They are required to be self-managing and autonomic, demanding minimal user's guidance. In pervasive computing, context-aware adaptation is a key concept to meet the varying requirements of different clients. In order to enable context-aware adaptation, context information must be gathered and eventually presented to the application performing the adaptation. It is clear that some form of context categorization will be required given the wide range of heterogeneous context information. Categorizations can be made from different viewpoints such as conceptual viewpoint, measurement viewpoint, temporal characteristics viewpoint and so on. To facilitate the programming of context-aware applications, modelling of contextual information is highly necessary. Most of the existing models fail both to represent dependency relations between the diverse context information, and to utilize these dependency relations. A number of them support narrow classes of context and applied to limited types of application, and most do not consider the issue of Quality of Contextual Information (QoCI). Along with a detailed context categorization, this paper will analyse existing context models and discuss their handling of dependency issues. It uses this analysis to derive a methodology for quality context information modelling in context aware computing.

1. Introduction

Pervasive Computing envisages a world with users interacting naturally with device-rich environments to perform a variety of tasks [Streitz and Nixon, 2005]. These environments are dynamic and heterogeneous. They are required to be self-managing and autonomic; demanding minimal user's guidance. In this heterogeneous environment of *Pervasive Computing*, context-aware [Coutaz et al., 2005] adaptation is a key concept to meet the varying requirements of different clients. In order to enable context-aware adaptation, context information must be gathered and eventually presented to the application performing the adaptation. It is clear that some form of *context categorization* will be required given the wide range of heterogeneous context information. Two important categorizations viewpoints are:

- *Conceptual viewpoint* – who, where, what occurs, when, what can be used, what can be obtained etc.
- *Measurement viewpoint* – what is the room temperature or network bandwidth or network latency etc?

To facilitate the programming of context-aware applications an infrastructure is necessary to gather, manage and disseminate context information to applications. And this infrastructure ultimately requires the *modelling of contextual information.* There are number of existing context descriptions based on one of the following methods:

- Set theory
- Directed Graph
- First-order Logic
- Preference and user Profiles

Most of these models fail to both represent *dependency relations* between the diverse context information and to utilize these dependency relations. A number of these support narrow classes of context and applied to limited types of application. Furthermore most of them do not consider the issue of *Quality of Contextual Information* (QoCI). This will

be a critical issue for the next generation pervasive computing; primarily because the quality of a given piece of contextual information will dramatically effect the decisions made by the autonomous application. Along with a detail context categorization this paper will analyse existing context models. Dependency relations, one of the missing issues in most of the existing context model are discussed. Further it presents a methodology for quality context information modelling in context aware computing.

The organization of the paper is as follows. Section 2 defines what we mean by context and context awareness. Context categorization and analysis of context models are presented in section 3 and section 4 respectively. Section 5 briefly describes the dependency relations in context information. A methodology of quality context information is presented in section 6, while section 7 concludes with some future directions.

2. What is context and context awareness?

It is quite unlikely that a single definition of context will be accepted by all researchers. From time to time, from application to application this definition varies. Historically [Winograd, 2001], "Context" has been adapted from linguistics, referring to the meaning that must be inferred from the adjacent text. In respect to computing world definitions of context varies with *computing environment* (available processors, devices accessible for user input and display, network capacity, connectivity, and costs of computing) *user environment* (location, collection of nearby people, and social situation) and *physical environment* (lighting, noise level etc). According to [Dey et al., 2000a] context is "*any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects*." Although this definition encompasses the definitions given by previous authors, it is sometimes too broad. [Winograd, 2001] has given a more specific and role based definition. According to him context "*is an operational term: something is context because of the way it is used in interpretation, not due to its inherent properties.*" Most recently [Coutaz et al., 2005] defined context "*is not simply the state of a predefined environment with a fixed set of interaction resources. It's part of a process of interacting with an ever-changing environment composed of reconfigurable, migratory, distributed, and multiscale resources.*"

Context awareness is a term from computer science, which is used for devices that have information about the circumstances under which they operate and can react accordingly. Context-aware computing involves application development that allows for collection of context and dynamic program behavior dictated by knowledge of this environment. Context-awareness is not unique to ubiquitous computing. For example, explicit user models used to predict the level of user expertise or mechanisms to provide context-sensitive help are good examples used in many desktop systems. With increased user mobility and increased sensing and signal processing capabilities, there is a wider variety of context available to tailor program behavior. Through context-awareness rapid personalization of computing services will be possible.

Today's computer systems are unaware of the user's context. They do not discern what the user is doing, where is the user, who is nearby and other information related to the user's environment. They just take the explicit input from the user, process it, and then output the result. Deemed as computing for the next generation, pervasive computing will greatly change the way today's computers behave. The basic idea is to instrument the physical world around us with various kinds of sensors, actuators, and tiny computers. The huge amount of information can then be collected and processed by computer systems, enabling computer systems to deduce the user's situation and act correspondingly with user's intervention [Nixon et al, 2002]. Active Badge System, Call Forwarding, Teleporting, PracTab system, Conference Assistant, Office Assistant, Classroom 2000, CyberDesk, etc are examples of present context aware Systems/Applications.

Category	Semantics	Examples
User context	Who?	*User's Profile:* identifications, relation with others, to do lists, etc
Physical context	Where?	*The Physical Environment:* humidity, temperature, noise level, etc
Network context	Where?	*Network Environment:* connectivity, bandwidth, protocol, etc
Activity context	What occurs, when?	*What occurs, at what time:* enter, go out, etc
Device context	What can be used?	*The Profile and activities of Devices:* identifications, location, battery lifetime, etc
Service context	What can be obtained?	*The information on functions which system can provide:* file format, display, etc

Table 1: Conceptual Categorization

3. Context Categorization

Context categorization will be required for the wide range of heterogeneous context information in next generation context aware computing. Context categorization helps application designer and developer to uncover the possible context and simplify the context manipulation. Classification context information can be helpful in providing quality context information. For example, conflicts can be resolved by favoring the classes of context that are most reliable (static followed by profiled) over those that are more often subject to error (sensed and derived).

Two possible broad categorizations viewpoints are:

- *Conceptual viewpoint* – who, where, what occurs, when, what can be used, what can be obtained etc.
- *Measurement viewpoint* – what is the room temperature or network bandwidth or network latency etc?

But most of the researchers did the categorization from conceptual viewpoint and some of them are following:

- [Gwizdka, 2000]
  - Internal Context: the state of the user
  - External context: the state of the environment
- [Petrelli et al., 2000]
  - Material Context: the location, device and available infrastructure
  - Social Context: social aspects and personal traits
- [Dey et al., 2000a]
  - Primary Context: location, time and activity
- [Schilit et al., 1994]
  - Primary Context: user environment, physical environment, computing environment

Although aforementioned categorizations are helpful but sometimes context information can't be clearly delimited and they are incomplete. Considering these issues this paper is aimed to provide a more comprehensive categorization from conceptual viewpoint as well as from measurement viewpoint.

Conceptual categorization:

The conceptual categorization of context (table 1) provides a description of the contextual space in terms of the actors, the actions and the relationships between them.

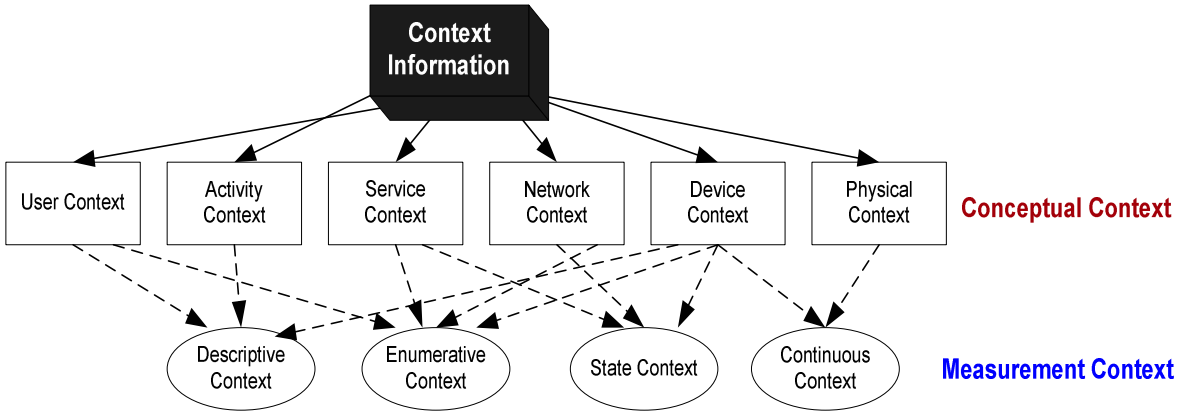


Figure 1: Hierarchical Categorization of Context information

Measurement Categorization:

- Continuous Context
- Enumerative Context
- State Context
- Descriptive Context

Continuous Context

In this category the value of context changes continuously. Continuous context component (ξ), is function of

-- current value of the context component,
-- lowest threshold value
-- highest threshold value
-- compare value
-- the metric of the value

and it uses function formula for the calculation.

Enumerative Context

Here the values of context are a set of discrete values and defined in a list or set. They are based on set operations. Like, enumerative context componentδ, **val** $(\delta) \in \Delta$, $\Delta = \{\delta_1 \ldots \delta_i \ldots \delta_n\}$

State Context

This category consists of two opposite values and they toggle between them. Like, state context component η **val** $(\eta) \in H$, $H=\{0,1\}$ and this is calculated in predicate calculus.

Descriptive Context

This is based on the description statement of the context and for this purpose it uses predicate calculus. For example;

location (CellPhone, loc_A)
location(laptop, loc_B)
location (obj1, loc1) ^ location (obj2, loc2) ^ (loc1^ loc) ^ (loc2 ^ loc) =>near(obj1,obj2)

Another context categorization could be done in terms of temporal properties of context information:

- **Static context:** Static context information describes those aspects of a pervasive system that are invariant, such as a person date of birth, social security number etc.
- **Dynamic context:** Pervasive systems are typically characterized by frequent changes; the majority of information is dynamic. The persistence of dynamic context information can be highly variable; for example, relationships between colleagues typically last for months or years, while a person's location and activity often change from one minute to the next.

Conceptual and measurement viewpoints contexts could be again classified as static or dynamic contexts. Above categorizations are not exhaustive for future's pervasive computing where context information will exhibit more diverse characteristics but these could be very helpful for application designer and developer in pervasive computing to manipulate context information efficiently.

4. Context Modeling

To facilitate the programming of context-aware applications an infrastructure is necessary to gather, manage and disseminate context information to applications. And this infrastructure ultimately requires the *modeling of contextual information.* Context modeling is highly important to capture:

- user requirements/profile, application requirements, device capabilities
- relationship between context

Context information is gathered, stored, and interpreted at different parts of the system. A representation of the context information should be applicable throughout the whole process of gathering, transferring, storing, and interpreting of context information. Most of the existing context models are based on one of the following methods:

- Set theory
- Directed Graph
- First-order Logic
- Preferences and user's Profiles (CC/PP and CSCP)

Set theory

- [Schmidt et al., 1999] used set theory for the context presentation. The context *T* is described by a set of two-dimensional vectors. Each vector *h* consists of a symbolic value *v* describing the situations and a number *p* indicating the certainty that the user (or the device) is currently in this situation.
- [Yau et al, 2001] also used set theory for the context and a *context-tuple* is defined as a tuple <ai, aj, ak, . .. an, tm> of size *n*, where *n* is the number of unique contextual-data sources present in the device. Each variable *ai* in the tuple represents a value, which is valid for the corresponding type of context. The variable *tm* represents the time of the tuple creation time.

Set theory describe context schematically and dependency relations are not embodied.

Directed Graph

[Henricksen et al., 2002] proposed an object-based context modeling in which context information is structured around a set of entities, each describing a physical or conceptual object such as person or communication channel. It uses the form of a directed graph for the diagrammatic representation of context, in which entity and attribute types form the nodes, and associations are modeled as arcs connecting these nodes. This is a comprehensive model which includes QoCI and dependency relations but fails to represent the dependency relation accurately.

First-order Logic

[Ranganathan et al., 2002] proposed a context model named *ConChat* and it is based on first-order predicate calculus and Boolean algebra. It covers the wide variety of available contexts and supports various operations, such as conjunction and disjunction of contexts and quantifiers on contexts. It allows the creation of complex first-order expressions involving context, so it is possible to write various rules, prove theorems, and evaluate queries. This modeling is consists of the four elements in the following ways:

- Context (<ContextType>, <Subject>, <Relater>, <Object>)

ContextType: the type of context,
Subject: person, place, or thing, with which the context is concerned,
Object: a value associated with the subject,
Relater: comparison operator, verb, or preposition
Examples:
context(people, Room 22,>=,3)
context(application, PowerPoint, Is, Running)
context(RoomActivity, 22, Is, Presentation)

This is a well defined modeling to specific field like electronic chat but in this model relation between continuous data cannot be described easily and even it is not dealing with QoCI.

Preferences and user Profiles

Composite Capability/Preference Profiles (CC/PP) [Klyne et al., 2001] is the W3C's proposal for a profile representation language and it is a framework based on the Resource Description Framework (RDF). CC/PP is intended to express both device capabilities and user preferences. Its specification defines a basic structure for **profiles.** A profile is basically constructed as a strict two-level- hierarchy: each profile having a number of **components**, and each component having a number of **attributes** (shown in figure 2). The particular components and attributes are not defined by the CC/PP specification. The definition of a specific vocabulary is up to other standardization bodies. Although CC/PP able to fulfill all the requirements except structural property of profile representation mentioned [Held et al., 2002] but vocabulary is not rich enough; it needs to be extended. Most importantly it can't represent the complex relationships and constraints. Even Component/Attribute model becomes clumsy if there are many layers.

Comprehensive Structured Context Profiles

Comprehensive Structured Context Profiles (CSCP) [Held et al., 2002] is based on the Resource Description Framework (RDF) and overcomes the shortcomings of the Composite Capability/Preference Profiles language (CC/PP) regarding structuring. Furthermore it extends the mechanisms to express user preferences. It can't represent the complex relationships and constraints. Component/Attribute model becomes clumsy if there are many layers.

From the above study it is quite clear that existing context models are suffering at certain extent which makes them not very suitable as a context model for future pervasive systems. Future's full fledged pervasive systems will require much more sophisticated context models in order to support seamless adaptation to changes in the computational environment. The context models will need to specify a range of characteristics/quality of context information including temporal characteristics (freshness and histories) accuracy resolution (granularity) confidence in correctness of context information, as well various types of dependencies among the different context information.

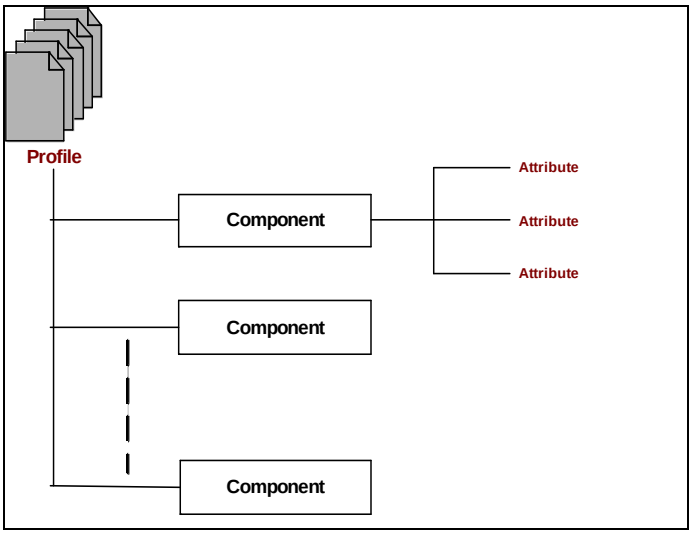


Figure 2: CC/PP

5. Dependency relations

Future pervasive and context aware systems will need to deal with heterogeneous services and contexts. It is very likely that these context information will be some how interrelated and dependent. According to [Henricksen et al., 2002], "*A dependency is a special type of relationship, common amongst context information, which exists not between entities and attributes, as in the case of associations, but between associations themselves*." Here associations are the unidirectional relationships between the entity and its attributes and a dependency shows the reliance of one association upon another. [Efstratiou et al., 2001] showed the importance of capturing dependencies in context aware applications. Without knowledge of such dependencies, inappropriate decisions can be made by context-aware applications that lead to instability and unwanted results. Moreover, knowledge of dependencies is important from a context management perspective, as it can assist in the detection of context information that has become out-of-date. Dependency relations will be critical in diverse context information and it can't be ignored most of the cases. Above analysis on the number of existing context models shows that they don't include these dependency relations and suffer for this issue. Hence future context models should include these dependency relations more comprehensively.

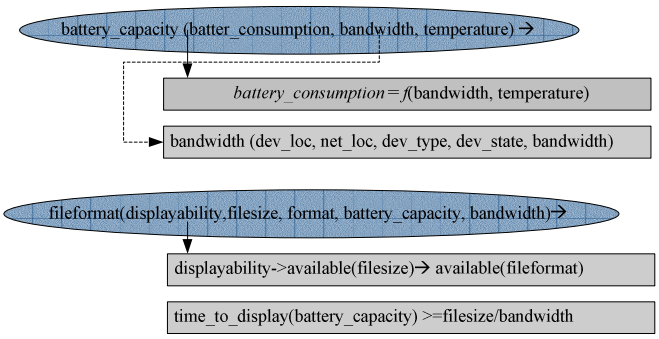


Figure 3: Dependency Description

Constraint Logic Programming Language [Marriott 1998] is a language, which allows the programmer simply to state relationships between objects and this, could be used for the description dependency relation. Constraint languages provide powerful, high-level descriptions for rule-based systems modelling which can operate on different types of (primary and derived) data. Consider, for example, displaying information in a smart phone like Nokia 6630. Figure 3 shows a sample scenario of the dependency description related to display information in a smart phone where two main concerns are battery power and file format.

6. Quality of Context Information (QoCI)

In context aware systems, errors in context information may arise as a result of errors in *gathering (sensing), interpretation* and *presentation* level. As context information is relied upon by applications to make decisions on the user's behalf, it is indispensable that applications have some means by which to judge the reliability of the information. *Quality of Contextual Information* or *data* is a judgment parameter or criteria for the contextual information or data. Most of the existing context models do not consider the issue of *Quality of Contextual Information* (QoCI). This will be a critical issue for the next generation pervasive computing; primarily because the quality of a given piece of contextual information will dramatically effect the decisions made by the autonomous application. Poor information or data quality can have severe impact on the overall effectiveness of the context aware system. Therefore inclusion of QoCI in the future context model is highly necessary.

Next generation pervasive and context aware systems will need to deal with heterogonous applications which will require diverse context information. Moreover these assorted applications will require various *Quality of Service (QoS).*To provide these QoS we need various *QoCI* to be incorporated in the context model.

Before analyzing or managing information or data quality, one must understand what information or data quality means. Information quality management requires understanding which dimensions of information quality are important to the user or application. According to [Wang et al., 1993] we can define *QoCI* in terms of information quality parameters and information quality indicators as below:

- ***An information quality parameter*** is a qualitative or subjective dimension by which a user evaluates context information quality. *Source credibility* and *timeliness* are examples.
- ***An information quality indicator*** is a context information dimension that provides objective information about the context. *Source, creation time, and collection method are examples.*
- ***An information quality attribute*** is a collective term including both quality parameters and quality indicators.
- ***An information quality indicator value*** is a measured characteristic of the gathered and stored data. The information quality indicator source may have an indicator value like from a sensor or user.

- ***An information quality parameter value*** is the value determined for a quality parameter (directly or indirectly) based on underlying quality indicator values. Application-defined functions may be used to map quality indicator values to quality parameter values. For example, because the *source is user himself for his date birth information*, so *credibility* is high.
- ***Information quality requirements*** specify the indicators required to be tagged, or otherwise documented for the information related to an application or group of applications. If a context model includes this then it is possible to make the context aware system more efficient and effective.

Necessity of the diverse quality of context information has been broadly recognized in number of research works, yet none of the existing work addresses the problem in an adequate or general way. [Dey et al., 2000b] suggests that ambiguity in information can be resolved by a mediation process involving the user. But in case of potentially large quantities of context information involved in pervasive computing environments and the rapid rate at which context can change, this approach places an unreasonable burden on the user. [Ebling et al., 2001] describe a context service that allows context information to be associated with quality metrics, such as freshness and confidence, but their model of context is incomplete and lacks formality. [Castro et al., 2001] defined notion of quality based on measures of accuracy and confidence, but their work limited to location information. Schmidt et al. associates each of their context values with a certainty measure that captures the likelihood that the value accurately reflects reality [Schmidt et al, 1999] .They are concerned only with sensed context information, and moreover take a rather narrow view of context quality. Gray and Salber include information quality as a type of meta-information in their context model, and describe six quality attributes: coverage, resolution, accuracy, repeatability, frequency and timeliness [Gray et al., 2001]. Finally [Henricksen 2002] included QoCI in their directed graph based context model but this could be limited to this sort of modelling. Most of their quality models are not formally defined, as they are intended to support requirements analysis and the exploration of design issues, rather than to support the development of a context model that can be populated with data and queried by applications.

Considering the above limitations in quality modelling our effort is to provide a generic approach of quality context information modelling based on [Wang et al., 1993]. Figure 4 shows the step by step methodology for quality contextual information modelling where initial input is *user's and corresponding application's* requirements and the final outcome of the modelling is the *quality schema*. Each step includes the *input*, *process* and *output*. Table 2 provides a brief description of each step:

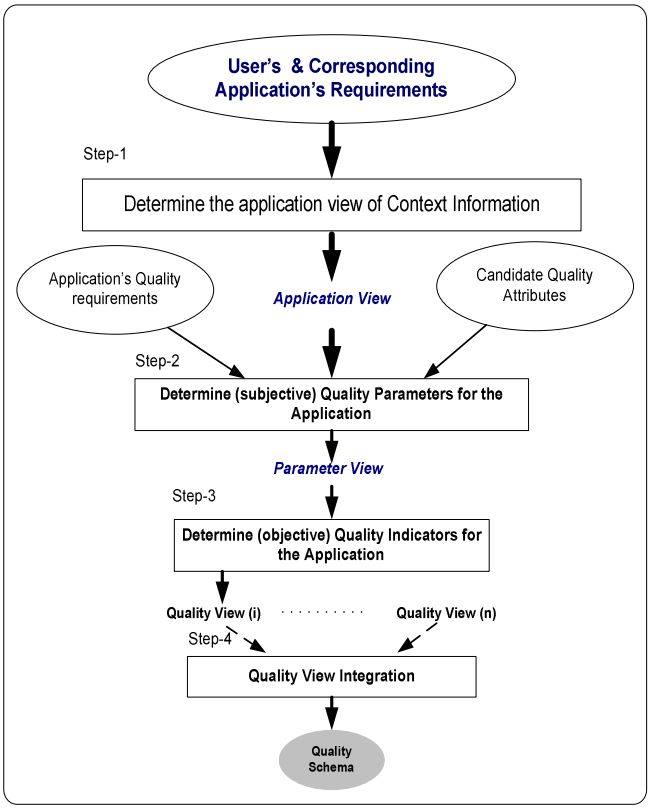


Figure 4: The process of quality contextual information modelling

Step No.	Input	Output	Process
Step-1	User's and Corresponding Application's requirements	Application view	It embodies traditional context information modelling and objective is to extract and document application requirements of context information.
Step-2	Application view, application quality requirement, candidate quality attributes	Parameter view	It determines the quality parameters (like timeliness, reliability etc) to support information quality requirements.
Step-3	Parameter view(application view included quality parameters)	Quality view	It converts the subjective quality parameters into measurable characteristics or quality indicators (like timeliness to date, etc)
Step-4	Quality view/views	Quality schema	This involves the integration of quality indicators.

Table 2: Brief description of the methodology for quality contextual information modelling

7. Conclusion

Next generation context aware systems have to deal with diverse context information. Categorization of this context information will be helpful for the context aware application designers and developers. To address this issue, this paper deals with categorizations and quality modeling in context information. Categorizations can me made from different viewpoints such as conceptual viewpoint, measurement viewpoint, temporal characteristics viewpoint. To facilitate the programming of context-aware applications, *modelling of contextual information* is highly necessary. An analysis of the number of existing models shows most of these models fail to both represent *dependency relations* between the diverse context information and to utilize these dependency relations. A number of these support narrow classes of context and applied to limited types of application. Moreover most of them do not consider the issue of *Quality of Contextual Information* (QoCI). A methodology for quality contextual information modelling in context aware computing is presented. The methodology is briefly described. Detail of quality modeling in contextual information with details of different application oriented quality dimensions can be extended in future work.

References

[Castro et al., 2001] Castro,P.,Chiu, P.,Kremenek, T.,Muntz,R. "A probabilistic room location service for wireless networked environments" UbiComp 2001 Conference, Atlanta (2001)

[Coutaz et al., 2005] Joëlle Coutaz, James Crowley, Simon Dobson, and David Garlan. "Context is key." Communications of the ACM, 48(3), March 2005

[Dey et al, 2000a] A.K.Dey, G. D.Abowd. "Towards a Better Understanding of Context and Context-Awareness."CHI2000 Workshop, 2000.

[Dey et al, 2000b] Dey,A., Manko.,J., Abowd,G. "Distributed mediation of imperfectly sensed context in aware environments."Technical Report GIT-GVU-00-14,Georgia Institute of Technology (2000)

[Ebling et al., 2001] Ebling,M.,Hunt,G.D.H.,Lei,H. "Issues for context services for pervasive com-

puting." Middleware 2001 Workshop on Middleware for Mobile Computing,Heidelberg (2001)

[Efstratiou et al., 2001] Efstratiou, C.,Cheverst,K.,Davies,N.,Friday,A. "An architecture for the effective support of adaptive context aware

applications.In:Mobile Data Management (MDM)"Hong Kong,China,Springer (2001)15-26.

[Gray et al., 2001] Gray,P.,Salber,D. "Modelling and using sensed context in the design of interactive applications." In 8th IFIP Conference on Engineering for Human-Computer Interaction, Toronto (2001)

[Gwizdka, 2000] J.Gwizdka. "What's in the Context."CHI2000 Workshop.

[Held et al.,2002] Held, A., Buchholz, S., Schill, A. "Modeling of Context Information for Pervasive Computing Applications" Proc. of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI2002), Orlando, FL, Jul 2002

[Henricksen et al., 2002] K.Henricksen, J.Indulska, A.Rakotonirainy. "Modeling Context Information in Pervasive Computing Systems."Proceedings Pervasive 2002 -Zurich August 2002.

[Indulska et al., 2003] Jadwiga Indulska, Ricky Robinson, Andry Rakotonirainy, Karen Henricksen. "Experiences in Using CC/PP in Context-Aware Systems."Proceeding of Mobile Data Management. Jan. 2003.

[Klyne et al., 2001]G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies", *W3C Working Draf*t, Mar 15, 2001.

[Marriott and Stuckey, 1998] Kim Marriott, Peter J. Stuckey. "Programming with Constraints: An Introduction."MIT Press. 1998.

[Streitz and Nixon, 2005] Norbert Streitz and Paddy Nixon, "The Disappearing Computer", Special issue of Communications of the ACM, 48(3), March 2005.

[Nixon et al., 2002] Paddy Nixon, Feng Wang, Sotirios Terzis and Simon Dobson. Engineering context-aware systems. In Proceedings of the International Workshop on Engineering Context-Aware Object-Oriented Systems and Environments. 2002.

[Petrelli et al, 2000] D. Petrelli, E. Not, C. Strapparava, O. Stock, M. Zancanaro. "Modeling Context is Like Taking Pictures."CHI2000 Workshop, 2000.

[Ranganathan et al., 2002] Anand Ranganathan, Roy H. Campbell,Arathi Ravi,Anupama Mahajan. "ConChat: A Context-Aware Chat Program."IEEE Pervasive Computing. Vol.1,Iss.3, July-Sept. 2002. p51 –57.

[Schilit et al., 1994] B. Schilit, N. Adams, R. Want. "Context-aware computing applications."Procsof IEEE workshop on Mobile Computing Systems and Applications. 1994. p85-90.

[Schmidt et al, 1999] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, et.al. "Advanced Interaction in Context."1stInternational Symposium on Handheld and Ubiquitous Computing (HUC'99).

[Wang et al., 1993] Wang, R.Y.; Kon, H.B.; Madnick, "Data Quality Requirements Analysis and Modeling" Data Engineering, 1993. Proceedings. Ninth International Conference on 19-23 April 1993 Page(s):670 - 677

[Winograd, 2001] T. Winograd, "Architecture for Context", Human Computer Interaction , Vol. 16,pp401-419,2001 .

[Yau et al, 2001] Stephen S. Yau, Fariaz Karim. "Context-Sensitive Middleware for Real-time Software in Ubiquitous Computing Environments."Proceedings. 4th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC2001). May 2001. p163 –170.

[Zaslavsky, 2002] Arkady Zaslavsky."Adaptability and Interfaces: Key to Efficient Pervasive Computing."NSF workshop series on Context-Aware Mobile Database Management.Jan.2002.

Forgetting the Local Knowledge Model – A Fundamental Problem for Autonomic Communications in Future Generation Networks

Peter Duxbury-Smith
Intelligent Solutions Ltd
York
United Kingdom
pete@intelligent-solutions.ltd.uk

John G Gammack
Institute for Integrated and
Intelligent Systems
Griffith University
Australia

Abstract

Distributed management and recompilation of knowledge in a future autonomic communications network will need means to 'forget' irrelevancies when reconciling circumstance to situation. Methods currently available are fragmented. The richest source of pertinent thought and possible solution appears in Cognitive Science and studies of human memory. An important component of a future practical solution will be effective initial encoding, because of its immediacy.

1. The "Knowledge Plane" Problem

Local knowledge represented and used in a situated communications network element needs to be periodically updated. It must adapt to changes in the surrounding context of consequence beyond assumptions made when the local knowledge was initially modelled. For example addition of new neighbouring elements, or a fault in one, may demand changes in behaviour that depends upon a model of knowledge about neighbouring, or even distant, elements. As network complexity increases, intelligence is required to manage and effectively analyse operational level activity. [Oppenheimer, 2003].

A practical and general example, focused on in this paper, is generation of a transaction data record (TDR). Transactions across multiple, geographically distributed elements in a communications network need to be represented in TDRs for many different purposes. An obvious one is for billing a voice call, or identifying electronic payments as in existing communications networks. Future generation networks are likely to provide for more complex transactions involving multiple sorts of service provider making billing an even more complex process of collecting together records about each transaction from many distributed elements [Telcordia 2004]. But TDRs are needed for other purposes besides billing, being the general way in which performance and operation of a communications network can be recorded for analysis. Applications include Fraud Management, Network Performance Management, Customer Profiling and Relationship Management and many more.

The records need to be collected from individual elements where local operations have been conducted when controlling a transaction then combined to represent an appropriate TDR. Not all transactions require the same information in TDRs. TDRs have to be constructed according to what services have been employed and what has been done using each service, in every instance of a transaction, and for each different type of TDR needed.

One way to cope with this requirement is to forward the minutiae of all transaction records to one central location where the extraction of TDRs is generated. But this treats peripheral network elements as dumb, and even in existing communications networks such centralised TDR extraction is not common practice. Instead intelligence is typically applied at elements and successive sub-managers to abstract information before being passed to a central manager. This arrangement lowers the demand on bandwidth required to centralise records. But it can also result in increased bandwidth for interactions amongst elements and sub-managers determining what information needs to be abstracted by each, because of dependencies across elements.

Furthermore, relational dependencies amongst element interactions can present a problem for tractability. Essentially the same problem is faced for distributed

knowledge representation in the Semantic Web. Which has led to pragmatic adoption of a Resource Description Framework (RDF) [Swartz 2002, O'Regan 2004] so that web resources themselves do not have to be searched to find meaning or potential value in them. Knowledge can be used proactively, in RDF, to limit potentially intractable search. Detail is reduced in RDF, but to make it effective requires careful modeling of how things happen with each resource. Where and when this knowledge is established is open. It is in keeping with the idea of Autonomic Networks that such ontologies should derive from local context, with each resource/element being responsible to "forget" what no longer applies.

With the philosophy of increasingly moving intelligence out to the periphery of the network in future generations we can see occurrence of the classic problem of interaction of autonomous element operations with the "Knowledge Plane": the parallel level of adaptive cognitive modelling about the underlying network [Clark *et al*, 2003]. Cognitive techniques, rather than algorithms, have been proposed for designing this functionality, but if the system is to be truly autonomic, the intelligent activity must be integral and effectively distributed. Generation of a TDR will require peripheral elements somehow "knowing" what information needs to be abstracted from local transaction data and forwarded to a central TDR producer. This knowledge will, in turn, be dependent on knowing what is happening simultaneously at different parts of the network of geographically distributed elements. Continuous context and situational awareness, allowing adaptive behaviour at the periphery must be accommodated.

2. The Business Case

If we consider existing Global Systems for Mobile communication (GSM) networks the error occurrence in billing runs at 8-9%. Some of this error is due to mediation of information in TDRs. Often the greater part is due to centralised processing during billing, in particular where incorrect tariff models are applied in rating calls. The operational maintenance of rating engines has historically been a cause of much billing error. For medium sized GSM operators who typically pay 50% of their operational costs to other network operators in billing reconciliation, having 8-9% error in billing has a substantial direct effect on bottom line.

In Future Generation Networks, "intelligence", such as that currently represented in centralised rating engines, will be distributed to peripheral elements. The maintenance of the knowledge used at the periphery will be even more of a challenge than is faced with centralised billing processes today. There are other business-critical network operations dependent on TDRs and processing intelligence that will be distributed, for example customer data integration [Wahedra, 2005] and enterprise information integration generally.

In general, with very high transaction volumes, storage, and fast access to stores becomes a business issue. Although storage may be relatively cheap, and there are audit and other requirements to retain full detail on transactions, capacity and access speeds are not unlimited, and thus models and data abstractions for intelligent business activity are required. For particular types of application, perhaps run by intermediary organisations or agents, a model that had forgotten (or had never known in the first place) specific details is also desirable from a business perspective, when the need to know is not required to be so comprehensive.

3. Managing and Distributing a Central Model of Network Knowledge

An alternative to either a fully-distributed or fully-centralised model of network knowledge is where local, situated models of knowledge are maintained by a central manager, which periodically updates and re-distributes these. Each local, situated model is harmonised with others by the central manager during a re-compilation process so that each is able to make reasonable predictions about what is occurring at other parts of the network and include these predictions in reasoning about local situations.

This recompilation process is equivalent to "forgetting" about knowledge that no longer applies due to changes in the network. It involves periodic re-learning of knowledge used to make predictions of what happens in the network. It may employ the testing of localised models of network context by individual elements during routine operations, and use the results of these tests as part of the recompilation process. The regression software testing approach described by [Rothermel & Harrold, 1996] which is suited to changing systems offers example. If systems have been designed using conceptual components that allow modular reasoning, as many recent systems have, defects can be reduced without requiring knowledge of source code of other components, not incurring cost traditionally associated with regression testing [Weide, 2001]. Again, the Codebook Correlation Technology used in SMARTS InCharge [Hasan, Sugla & Ramesh, 1999] is a means to compile and distribute localised "smart" models for network event correlation under SNMP.

We are drawing a parallel between update or recompilation and the cognitive processes of memory and forgetting. There is a considerable literature in cognitive science that details models for these processes, and we now briefly sketch, with considerable simplification, some key ideas from this that can play a role in designing specific methods for knowledge update.

Cognitive theories of memory and forgetting have often used the metaphor of a store, augmented by active, working components that "retrieve" information and process it in some context. Often used information is reinforced; little used information decays, (or slowly obsolesces); processes sensitive to context determine relevance and selection, and

outputs are constructed from models of past experiences, modified by circumstances, but largely in a reliable way. Items that are multiply associated, or occur in different contexts are easier to remember, and harder to forget. For different forms of output reporting information granularity is contextually determined, and abstractions or integrations are made to cover many instances, whose particular details become lost or fuzzy as the general pattern is established. This pattern often becomes used instead as a template, and details may be filled by imaginative processes. Case based reasoning is a technology in tune with this.

Often patterns can become automated and apply subconsciously. These can be resistant to change. As time goes on, mobile humans may typically first forget the landline number of a previous house they once lived in, later the house number and street but rarely the city. Such familiar phenomena may be variously explained using a range of widely accepted theories, cast in terms of associations, levels, usage, decay, context, working memory, long term/short term stores, learning, dynamic construction, and invoking specialised forms of cognitive constructs for specific phenomena, at individual and social levels of description. By maintaining some awareness of what information can be sourced externally, the need to remember details is reduced. Having a model of what types of information are available elsewhere in the network allows this to occur during a recompilation process, when information available elsewhere is made redundant.

Cognitive science has largely focused on individual processes of memorisation, although there is also some work on social and ecological structures of memory. In management science, at organisational levels there is a separate literature on organisational memory, closely aligned to the idea of knowledge management. This has also tried to symbolize and externally model the content of knowledge held corporately, or by employees, and has evolved from the early ideal of expert systems (a knowledge base of facts and rules with a rational inference process) through machine learning, neural nets, data mining and knowledge discovery, to metadata schemes and indexed repositories. This literature, though fulsome, has lagged cognitive science in the sophistication of its models, and only recently have notions such as strategic forgetting and testing knowledge models against changing contexts become more prominent. In the next section we explore some approaches that try to identify heuristics and methods that can be used to "forget" local situated knowledge that no longer applies when making predictions about what is happening in the network at large.

These heuristics and methods would be used to maintain localised network knowledge in Self-Managing networks of distributed "smart" elements by harmonising it with the central "Knowledge Plane".

4. Some Heuristics and Methods for Forgetting Local Models

In studies of human memory and forgetting, a long tradition of work in psychology and neural modelling has identified a range of cognitive constructs relevant to designing algorithms that replicate or extend the behaviour of intelligent agents. These ideas hold true beyond the limitations of laboratory work and artificial worlds, to the idea that real world influences shape what is represented and remembered.

Key ideas in this tradition include consolidation of trace, interference of various kinds, and effectiveness of initial encoding, with sophisticated theories around the basic ideas, for which there is much evidence, and as such, the ideas have never gone away. Each explains data on some phenomena well but not others, and computational models of the processes provide a starting point for more general forgetting models using these themes [Meeter & Murre, 2004]. It is likely however that effective theories and models of forgetting will entail more than one of these mechanisms, and semantic and knowledge processes are likely to play a role in operating these. Indeed it is fair to say that if identifying complete mechanisms for forgetting eludes current psychological research, computational modelling promises solutions not bound by human limitations.

Several methods that suggest partial mechanisms have been mooted in the literature. We review some of these briefly then look prospectively at the shape of future theory.

4.1. Projective Visualisation

With Projective Visualisation [Goodman 1994] a memory of what has happened before is used to predict what will happen next, and at successive stages following that, given a current situated context. If prediction of what happens next proves accurate, then predictions of successive stages are maintained (and even further look ahead may be carried out on their basis). If there proves to be a variation from the memory-based prediction, then the predication is discarded/ "forgotten" so that it can be replaced by one worked out afresh, again by matching memory. In a subsequent, "off-line" recompilation of memory, the variation from prediction may be noted and used to update the memory by identifying what distinguishes the situation in hand from what was remembered, and, where appropriate, adding this newly-learned situation to renewed memory. "Forgetting" here is a process of testing sets of feature values that are found to be predictors of following feature values. And where prediction fails, either discarding predictors or finding something that distinguishes the current set of feature values from similar sets held in memory.

This is an inductive and rational solution strategy, incorporating evolutionary and learning elements, but applicable in situations of bounded rationality. A general description and model of this has been outlined by [Arthur,

2001] which allows for intelligent activity at agent level in a wider adaptive complex system that replaces ineffective patterns or "belief models".

4.2. Visualising partial network topology

Trailblazer is a Model-Based network event correlation system developed by Duxbury-Smith for a large GSM operator. Trailblazer is unlike most commercially available Network Management Systems where rigid hierarchical management structures permit only primitive, local event correlation, Trailblazer deals with vast quantities of network information from the entire network. This has the advantage of freedom in knowledge-based interpretation of events, allowing relationships to be found amongst events from anywhere in the network. But Trailblazer has also to be able to "forget" irrelevant information and differentiate amongst simultaneous local situations.

Trailblazer was based on intensive knowledge elicitation with and observations of human network operators performing the same diagnostic task. It uses a version of the Connected Components algorithm to construct a visualisation of relevant network topology from event information. Then it uses this visualisation to guide Model-Based correlation. It orients to parts of network topology indicated to be of active interest because current events are generated there.

But this technique relies upon being able to collect all network information to a central point and operate on it there.

4.3. Forgetting by long-term memory trace decay

[Nachev & Ganchev, 2003] argue that there are parallels between human forgetting and classic Adaptive Resonance Theory (ART2 neural networks) conceiving of "forgetting" as the release of atrophied or unused resources: They identify four factors involved in forgetting from cognitive psychology: Trace decay, interference, physical damage and emotion. They restrict their focus to the first two, particularly trace decay. In autonomic systems, if appropriate, emotion may best be equated with policy priorities, and modeled using weightings. Trace decay and interference are however established notions with considerable research detailing their operation. Traces of encoded memories may fade, and lie dormant but stable: They may or may not take part in reconfigurations as new information (memories) are integrated.

4.4 Representation of forgetting as retrieval failure

Memory traces or other encodings in autonomic systems need never be lost, but access to them can be compromised. [Cox & Ram, 1992] take a negative view of forgetting, as a failure to retrieve knowledge at the appropriate time.

> "If a system's knowledge is not indexed or organised correctly, it may make an error, not because it does not have either the general capability or specific knowledge to solve a problem, but rather because it does not have the knowledge sufficiently organized so that appropriate knowledge structures are brought to bear on the problem at the appropriate time".

They identify four types of forgetting: Absent Memory, Absent Index, Absent Retrieval Goal, Absent Feedback. In addition to these retrieval oriented mechanisms, focused on the models of data organization, the use of the term "appropriate" implies that means of identifying relevance also exist – A contextual awareness that guides organisation and dynamic recompilations.

5. Effective initial encoding

If cognitive models are to apply to guide the forgetting process, initial encoding is also worth a closer look. This (original) notion suggests that items are immediately coded for relevance and weighted accordingly. A process of resource availability allocation decision determines whether an item is notified for future contribution to pattern identification, or considered immediately as negligible. This eliminates noise from the system at an early stage, but depends on the relative availability of storage, and the ability to determine relevance sufficiently early. Clearly the latter is a matter of experience, where relevance is adjudged against (modified) history, or in the absence of a specific model, under a principle derived with awareness of context [Sperber and Wilson, 1986]. This requires a more central intervention but also effective abstraction at intermediate levels. Our question here is how such abstraction might occur at a conceptual level, when storage limitations, and encoding itself are not at issue.

Theories of cognitive architecture [Anderson, 1983] suggest that it is more adaptive to forget trivial details than attempt to store everything, but this supposes a means to identify what is trivial. Activation, rehearsal and reinforcement processes dynamically weight associative links to form patterns in such designs. In an autonomic system identifying and matching particular recurrent patterns provides means to recognise something may be worth memorising, and this can feed into a weighting algorithm, which is periodically assessed for retention of important structuring information. This is unlikely to involve a single mechanism.

Mechanisms of suppression, repression and inhibition have been identified and operationalised in cognitive science for handling these phenomena (e.g. Minerva [Hintzman, 1986]). In neural net research, the Governor Architecture [Stober, Meeden & Blank, 2004] is one method for avoiding catastrophic forgetting which can occur from interference as new patterns become trained. It is particularly useful in dynamic online environments whose characteristics are not known in advance, and the method

can determine the patterns representing "key events that merit rehearsal".

In psychology, [McNamara and McDaniel, 2004] propose a knowledge based model which draws upon [Kintsch, 1988]'s Construction Integration model of comprehension. It explains data across various experimental situations, addressing the assimilation of new knowledge at expense of extant structure. Their work suggests forgetting is complex, and that single mechanisms, such as theories of inhibition alone, are insufficient for a full model. Work in neuroscience highlights consolidation of new memory traces and how forgetting is influenced by processes interfering with that. Reactivated memories may be vulnerable to similar processes [Wixted, 2004]: Effective memory systems do not just wipe old memories but can function with partially degraded ones and allow the possibility of reconstruction. Note that all of this is not noticed consciously – The functions are autonomic, hiding their complexity from their hosts. Autonomic systems, because the underlying data plane is logically independent of knowledge based models that can access it, can work at this level of abstraction without loss. Instead a focus on more optimized access to knowledge relevant in larger configurations will be required. Self managed "forgetting", through relevance assessment and recompilation, or constructive reintegration, addresses the issue of integrated systemic intelligence. The latter is akin to a "middle management" awareness of global (knowledge plane) context, but with access to detailed data below. The mechanisms to be designed will thus be less algorithmic, and more based around reconfiguring and fitting quasi-cognitive patterns into holistic and parameterized knowledge constructions.

6. Forgetting in Autonomic Communications Networks

We have mentioned but a few possible kinds of "forgetting" heuristic that might be employed in a Future Generation Autonomic Communications Network to maintain salience of situated intelligence at peripheral elements. In order to see how they might be used let us consider the use of TDRs:

We may reasonably expect processes for finding out about services, what they cost or even bidding for their price, combining them and making payments will become so sophisticated that they can be conducted instantaneously, in real-time with ease and creativity on the part of users. Users will be supported by automated intelligence to facilitate the construction, configuration and management of services which we strain to conceive of today.

Communications networks will not be like the switched telecoms networks of today, instead involving a larger variety of services and systems. Some "elements" of the network may include intelligent agents. Nevertheless the abstraction and correlation of network information (into some forms of TDRs) will remain key network functions.

Within sight of today's technology we can imagine the case of real-time rating and discounting for a new service that has been designed, configured and activated "on the fly" by self-organising elements in the Autonomic Network: Some of the elements know each other already through a history of transactions. Others are total newcomers to that particular functional group. They are owned by various service providers each with their own policies for such things as security, and each has their own tariff models for rating and discounting their services. These models are sophisticated beyond anything around today, able to take into account many factors and ranges of data from operational and business contexts. There may even be negotiating automated Agents which can setup services and arrange tariffs "on the fly".

In this fast-moving and complex scenario, the ability to rapidly adapt to new situations and to extract the right TDRs has survival value. Forgetting heuristics derived from Case Based Reasoning may be useful where familiar sets of elements and situations exist that are readily matched to known cases or where there is rich contextual information. Trace decay forgetting could apply where key knowledge is dispersed across elements and has to be retrieved and expanded for particular situated contexts. Visualising partial topologies to add meaning and reduce uncertainty, or using "look ahead" for testing out predictions and also reduce uncertainty are methods to support forgetting of unhelpful detail in TDRs. By the same token, they (amongst other methods) could reinforce immediate choice of detail included in TDRs.

Given the importance of extraction of information into TDRs to network management it may be more useful to think about how local information is thrown away (ie. "forgotten") in Autonomic Communications Networks than to try to filter or control what is remembered at arbitrary levels of managing network element. The immediacy of effective initial encoding characterizes what we would expect of good future solutions and theory.

7. Conclusion

When communications networks become autonomic we have to think not only of adding knowledge and intelligence to situated elements, but also of how knowledge can be taken away i.e. "forgotten".

There is considerable risk to a communications network business due to irrelevant or misleading knowledge that is hard to track down and manage. Risk is increased for autonomic networks due to complex distribution of such knowledge and responsibilities for maintaining it.

While the knowledge plane offers a model for managing a dumb data plane, this conflicts with the requirement to move intelligence and autonomy to the periphery, raising a classic problem. We propose an integrated focus on forgetting local knowledge (details) through periodic recompilations of models that are aware of their relevance in

the global context. The knowledge plane allows weightings and policy settings to guide which recompilations or reintegrations apply, and strategic forgetting in data networks is enabled. Effective initial encodings, can be enabled by reference to the values set higher up, setting initial weightings, and continuously monitoring those as the data network expands.

Cognitive theories of memory and forgetting offer richer territory for solutions to distributed management and recompilation of knowledge than do current techniques of network monitoring and management.

Fragmented methods and heuristics for "forgetting" situated knowledge from existing technology are indications of the current primitive stage of work from which we approach Autonomic Communications Networks. Future theory and solutions should more elegantly explain and implement effective initial encoding.

References

[Anderson, 1983] Anderson JR, The Architecture of Cognition, Harvard UP Cambridge MA.

[Arthur, 2001] Arthur WB, Inductive Reasoning and Bounded Rationality www.santafe.edu/arthur/Papers/El_Farol.html Accessed March 12th 2005

[Clark *et al*, 2003] Clark, D. D., Partridge C. J., Ramming C. & Wroclawski J.T., A Knowledge Plane for the Internet, *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Karlsruhe 2003, 3-10

[Cox & Ram, 1992] Cox Michael T. & Ram A., An Explicit Representation of Forgetting In: J. W. Brahan and G. E. Lasker (eds.), *Proceedings of the Sixth International Conference on Systems Research, Informtics and Cybernetics*, Baden-Baden, Germany August, 1992, 115-120.

[Goodman, 1994] Goodman, M., Results on Controlling Action with Projective Visualization., *Proc. AAAI 1994*, 1245-1250

[Hasan, Sugla, Ramesh, 2005] Hasan M., Sugla B., Ramesh V., A Conceptual Framework for Network Management Event Correlation, Bell Laboratories, www.belllabs.com/user/rv/external/publications/im99.ps Accessed 12 March 2005

[Hintzman, 1986] Hintzman D.L., Schema Abstraction in a Multiple Trace Memory Model, *Psychological Review* 93, 411-428

[Kintsch, 1988] Kintsch, W., The Use of Knowledge in Discourse Processing: A Construction-Integration Model, *Psychological Review* 95, 163–182.

[McNamara & McDaniel, 2004] McNamara Danielle S. & McDaniel Mark A., Suppressing Irrelevant Information: Knowledge Activation or Inhibition? *Journal of Experimental Psychology* Vol. 30, No. 2, 465–482

[Meeter & Murre, 2004] Meeter M. & Murre J.M., Consolidation of Long-Term Memory: Evidence and Alternatives, *Psychological Bulletin* 130(6), 843-857

[Nachev & Ganchev 2003] Nachev A & Ganchev I., Learning with Forgetting: An Approach to Achieve Adaptive Neural Networks, Accessed 12 March 2005 www.interdisciplinary.net/ci/AI/s5.htm

[O'Regan, 2004] O'Regan T., RDF and the Semantic Web linuxgazette.net/105/oregan.html Accessed March 13 2005

[Oppenheimer,2003] Oppenheimer D., The Importance of Understanding Distributed System Configuration. *CHI 2003* (Conference on Human Factors in Computing Systems) workshop, April 2003, roc.cs.berkeley.edu/papers/dsconfig.pdf Accessed March 12th 2005

[Rothermel & Harrold, 1996] Rothermel G. & Harrold, M. J., Analyzing regression test selection techniques, *IEEE Transactions on Software Engineering*, Vol. 22, no.8, August, 529-551, Accessed March 12th 2005 csce.unl.edu/~grother/papers/tse96-2.pdf

[Sperber & Wilson, 1986] Sperber D. & Wilson D., *Relevance: Communication and Cognition*, Blackwell, Oxford.

[Stober, Meeden & Blank, 2004] Stober J., Meeden L. & Blank D., The Governor Architecture: Avoiding Catastrophic Forgetting in Robot Learning under review mightymouse.brynmawr.edu/~dblank/papers/sab04.pdf Accessed March12th 2005

[Swartz, 2002] Swartz A., The Semantic Web in Breadth, logicerror.com/semanticWeb-long Accessed March 13th 2005

[Telcordia 2004] Revenue Assurance – a single solution to multiple problems Issues brief September www.telcordia.com/collateral/issues_briefs/rev_assurance.pdf Accessed March 12th 2005

[Wahedra, 2005] Wahedra A., Would you call a plumber to build your house, *Business Intelligence*, www.businessintelligence.com/ex/asp/code.142/xe/article.htm Accessed March 12th 2005

[Weide, 2001] Weide B.W., Modular Regression Testing: Connections to Component-Based Software *Proceedings Fourth ICSE Workshop on Component-Based Software Engineering*, IEEE, May 2001, 47-51. www.sei.cmu.edu/pacc/CBSE4_papers/Weide-CBSE4-7.pdf Accessed March 12th 2005

[Wixted, 2004] Wixted J., The Psychology and Neuroscience of Forgetting *Annual Review of Psychology* Vol. 55, 235-49

Dynamic Bayesian Networks: a contribution to Autonomic Communications and the Knowledge Plane?

Roy Sterritt [1] and Adele Marshall [2]
[1]University of Ulster and [2]Queen's University, Belfast
Northern Ireland
r.sterritt@ulster.ac.uk, a.h.marshall@qub.ac.uk

Abstract

Systems that are subject to uncertainty in their behaviour are often modelled by Bayesian Networks (BNs). These are probabilistic models of the system in which the independence relations between the variables of interest are represented explicitly. A directed graph is used, in which two nodes are connected by an edge if one is a 'direct cause' of the other. However the Bayesian paradigm does not provide any direct means for modelling dynamic systems. There has been a considerable amount of research effort in recent years to address this. This paper reviews these approaches and proposes a new dynamic extension to the BN. This paper proceeds to discuss fault management of complex telecommunications and how the dynamic Bayesian models can assist in the prediction of faults.

1 Introduction

Systems that are subject to uncertainty in their behaviour are often modelled by Bayesian Networks (BNs). These are probabilistic models of the system in which the independence relations between the variables of interest are represented explicitly. A directed graph is used, in which two nodes are connected by an edge if one is a 'direct cause' of the other.

However BNs provide no direct mechanism for representing temporal dependencies [Aliferis and Cooper, 1983], [Allen, 1983], [Young and Santos, 1996]. In certain domains such as medicine, planning and control, and industrial environments, the incorporation of a temporal aspect into the model is crucial if the model is to achieve an effective and accurate representation of the system in question. The time that symptoms appear and their duration, the time that observations/measurements are made and the time that faults are induced can significantly affect the formulation of hypotheses used. The model must be able to update the system given that observations and evidence can be made over time, that is capture the evolution of the system as it changes over time.

This paper is organised as follows. Section 2 highlights the motivation - Section 3 introduces the fault management domain. Section 4 describes Bayesian Networks. Section 5 summarises the published research into adapting BNs with a dynamic or temporal dimension. Section 6 proposes an alternative approach combining BNs with survival analysis. Section 7 explores how this can be used for fault management and section 8 finally ends the paper with a conclusion and future work.

2 Cognitive Networking: The Knowledge Plane

The need for a knowledge plane has been identified as necessary in next generation networks to act as a pervasive system element within the network to build and maintain high level models of the network. These indicate what the network is supposed to do to provide communication services and advice to other elements in the network [Clark *et al*, 2003]. It is generally considered that this knowledge plane will rely on the tools of AI and cognitive systems to meet the uncertainties and complexities of this goal, rather than traditional algorithmic approaches [Clark *et al*, 2003],[Agosta and Crosby, 2003].

In terms of creating the knowledge plane possible building blocks that have been highlighted include epidemic algorithms (for distributing data), Bayesian networks (for learning), and so on [Clark *et al*, 2003], [KP Resources, 2004]. At the same time one of the potential roles identified for the knowledge plane is fault diagnosis and mitigation [Clark *et al*, 2003].

In previous work we have investigated using Bayesian networks (formerly Bayesian Belief Networks) for telecommunication fault management systems [Sterritt *et al*, 1997], [Sterritt and Liu, 2001], [Sterritt, 2001, 2002]. We also proposed to extended this to include a time component were by the Bayesian network represents a developing situation over time (Dynamic Bayesian networks) [Sterritt et al, 2000b]. Essentially a Bayesian network is learnt from alarm event data to create (along with human assistance) a BN that diagnosis the fault from the evidence presented in terms of alarms (or at least correlates further the alarm events). The challenge experienced with this research (apart from the down turn in the telecommunications market at that time!) was that the approach although always receiving positive feedback from telecommunications partners (due to

the BNs transparency with its visual aspect and probability basis) was difficult to implement sensibly due to the large amount of alarm events possible and the number of fault conditions they may represent. In the end we had moved towards rules and pattern matching to provide initial masking and correlation with the BN at a higher level. This has certain resonances with the knowledge plane, as such the motivation of this paper is to briefly review BNs and DBNs once more with a view they may be useful for the knowledge plane.

3 The Management Plane & Fault Diagnosis

High-speed broadband telecommunication systems are built with extensive redundancy and complex management systems to ensure robustness. The presence of a fault may not only be detected by the offending component and its parent but the consequence of that fault discovered by other components. This often results in a nett effect of a large number of alarm events being raised and cascaded to the element controller.

The behaviour of the alarms is so complex it appears non-deterministic [Bouloutas et al, 1994]. It is very difficult to isolate the true cause of the fault. Failures in the network are unavoidable but quick detection and identification of the fault is essential to ensure robustness. To this end the ability to correlate alarm events becomes very important.

The major telecommunication equipment manufacturers deal with alarm correlation through alarm monitoring, filtering and masking as specified by ITU-T [1988] and other international standard bodies, with rule-based type systems for assistance to the operator. Yet often it is left to the operator's expertise to determine the actual fault or multiple-faults from the filtered set of alarms reported.

At the heart of alarm event correlation is the determination of the cause. The alarms represent the symptoms and as such, in the global scheme, are not of general interest once the failure is determined [Harrison, 1994]. There are two real world concerns: (1) the sheer volume of alarm event traffic when a fault occurs; (2) the cause not the symptoms.

Alarm monitoring, filtering and masking meets criterion (1), which is vital. They focus on reducing the volume of alarms but do not necessarily meet criterion (2) to determine the actual cause - this is left to the operator to resolve from the reduced set of higher priority alarms. Ideally, a technique that can tackle both these concerns would be best.

4 Bayesian Networks (BNs)

Bayesian Networks (BNs) offer a potential solution. BNs consist of a set of propositional variables represented by nodes in a directed acyclic graph. Each variable can assume an arbitrary number of mutually exclusive and exhaustive values. Directed arcs (arrows) between nodes represent the probabilistic relationships between nodes. The absence of a link between two variables indicates independence between them given that the values of their parents are known. In addition to the network topology, the prior probability of each state of a root node is required. It is also necessary, in the case of non-root nodes, to know the conditional probabilities of each possible value given the states of parent nodes or direct causes. A good illustration of a BN and its related joint probability distribution is contained within Lauritzen and Spiegelhalter's paper [Lauritzen and Spiegelhalter, 1988] where they consider an example based on doctors diagnoses of patients suffering from shortness of breathe (dyspnoea), Figure 1.

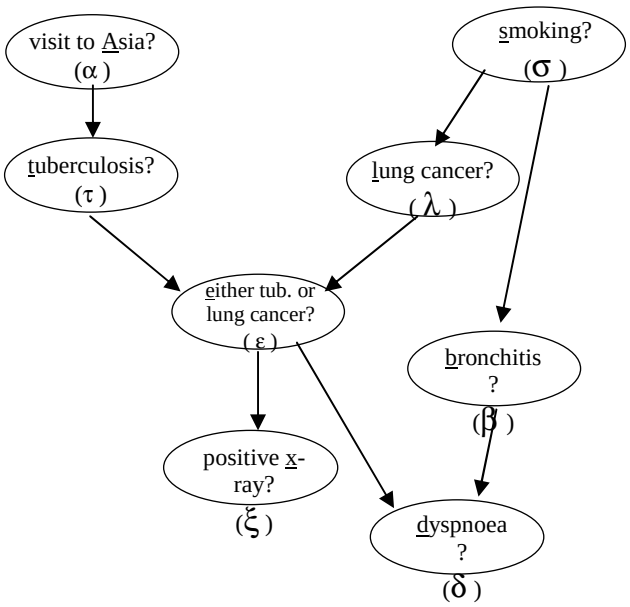


Figure 1: Lauritzen and Spiegelhalter's illustration of a BN

The graph can be considered as representing the joint probability distribution for all the variables. In the above example this is $P(\alpha,\tau,\varepsilon,\delta,\lambda,\beta,\sigma)$. The chain rule can re-express this joint probability as the product of the conditional probabilities which need to be specified for each variable or node.

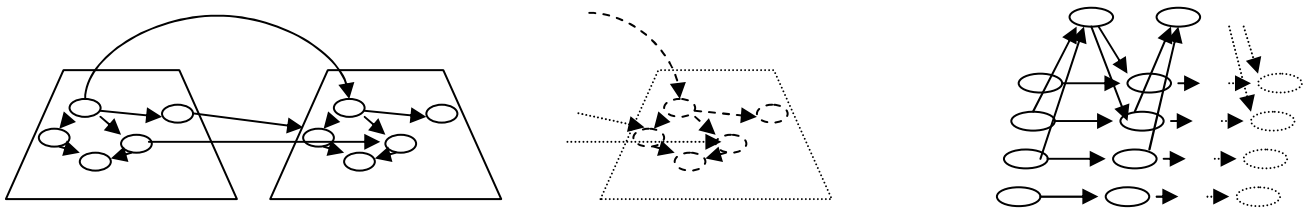
$$P(\alpha,\tau,\varepsilon,\delta,\lambda,\beta,\sigma) = P(\alpha)P(\tau\mid\alpha)P(\xi\mid\varepsilon)P(\varepsilon\mid\tau,\lambda)P(\delta\mid\varepsilon,\beta)P(\lambda\mid\sigma)P(\beta\mid\sigma)P(\sigma) \quad (1)$$

The chain rule is given below:

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_i \mid pa(X_i)) \quad (2)$$

where $pa(X_I)$ is the parent set of X_i.

Each node has associated with it a conditional probability table that quantifies the effects that the parents have on the node. Taking the graph as a whole, the conditional probabilities and the structure can be used to determine the marginal probability or likelihood of each node holding one of its states.



(a) (b)

Figure 2: Time slices, adapted from [Hanks et al, 1995].

The power of the BN comes to light whenever we change one of these marginal probabilities. The effects of the observation are propagated throughout the network and the other probabilities updated. In simple networks the marginal probabilities or likelihood of each state can be calculated from the knowledge of the joint distribution, shown earlier, using the product rule and Bayes' theorem. This simply means that the DAG is singly connected; each link is a bridge where the removal of one leads to a disconnected network.

However, cycles often occur and the calculation is much more complex. Algorithms have been devised to cope with the complication of such cycles. Some calculate the marginal probabilities exactly but by doing so introduce calculations which are NP-hard [Cooper, 1990]. Therefore many researchers have developed algorithms which approximate the marginal probabilities. This may lead to the compromise of accuracy over a lower computational overhead.

The BN can be used for deduction in the fault management domain. Given alarm data it will determine the most probable cause(s) of the supplied alarms, thus enabling the system to act as an expert system.

In previous work [Sterritt et al, 1998] have developed an exact algorithm approach to deduce the marginal probabilities for their BN application based on that developed by Lauritzen and Spiegelhalter [1988].

5 Dynamic Belief Networks (DBNs)

A problem with the standard theory of belief networks is that there is no natural mechanism for representing time [Aliferis and Cooper, 1983], [Allen, 1983], [Young and Santos, 1996]. There have been various efforts to extend the theory to allow time to be modelled. For example, where probability of movement from one state to another has a temporal dependency, survival analysis [Marshall et al, 1999] can be used. Constraints on the behaviour of the system can be expressed using the formalism of temporal logic.

Dynamic Belief Networks (DBN) model a system that is dynamically changing or evolving over time [Faddy, 1994], [Kanazawa et al, 1995], [Kjarulff, 1992]. This model will enable the user to monitor and update the system as time proceeds.

Bayesian Networks were not designed to model temporal relationships explicitly; they are a static model. The prediction and deduction made do not vary depending on when the observations or predictions are made.

This standard theory of belief networks has been further developed by researchers to incorporate a temporal feature or time element into the model. This has been approached in various different ways. Aliferis and Cooper [1983]0 summarise just some of the extensions of belief networks for time modeling presented over the last few years. These include temporal influence diagrams [Provan, 1993b], Dynamic Belief Networks (DBNs) [Dagum et al, 1992], temporal models of endogenous change [Hanks et al, 1995], Temporal Bayesian Networks (TBNs) [Young and Santos, 1996], Temporal Nodes Bayesian Network (TNBNs) [Arroyo-Figuerao et al, 1998], embedded Markov processes [Berzuini et al, 1989], logic and time nets [Kanazawa, 1991], [Kanazawa, 1992], Modifiable Temporal Belief Networks (MTBNs) [Aliferis and Cooper, 1983], as well as specific applications [Berzuini et al, 1989], [Nicholson and Brady, 1994].

An obvious way of classifying the literature is to use the authors' individual terms (as above) to describe the various approaches. However, in most cases these terms, mainly dynamic and temporal, are interchangeable. For instance, if the time slices of a temporal model were applied so that the movement between the slices was based on a change in state instead of time, we could then classify them as belonging to a dynamic model. Likewise if Markov-chain approaches, dynamic models, were implemented that each state was a point in time, we could classify the applications as temporal models. Therefore we can say that the difference is primarily dependent on the application of the model.

Another approach of classifying the literature is to consider how the methods actually model the time/temporal element. This classification has been used by Palmer et al. [2000] who divide the temporal approaches into two main categories of time representation, namely those models which represent time (1) as points or instances or (2) as time intervals.

Within category (1) the models, based on points in time, require that events occur instantaneously where each event considered occurs at an instant in time. These are basically the time slice models and temporal reasoning models reviewed by Hanks *et al*. [1995] and illustrated in Figure 2.

Figure 2(a) represents an approach where a time slice is used to represent a snapshot of the evolving temporal process [Kanazawa et al, 1995]. The belief network consists of a sequence of sub-models each representing the system at a particular point or interval in time (time slice) and which are interconnected by temporal relations. Kjaerulff [1992], Dagum et al. [1992], [and Galper, 1993], Provan [1993a], Berzuini [*et al*, 1989], Lekuona [et al, 1995] are just some of the researchers currently using the time slice approach.

Figure 2(b) represent models where the network is composed of sub-models and duplicated over time slices, as before. However links between state variables within a time slice are disallowed. Dean and Kanazawa [1989] and Kanazawa [Kanazawa, 1991], [Kanazawa, 1992] use this approach in their research.

Hanks et al. [1995] proposed a modification to the time slice approach where they take into account the system as it changes over time, both due to exogenous and endogenous influences.

Category (2) of the classification approach in Palmer et al. [2000] considers interval representations of time. Allen's interval algebra and its 13 relations were used to provide the temporal basis for the model [Allen, 1983]. This may be more broadly thought of as a dynamic model as an interval in time represents an event or process during which a property (either true or false) holds uniformly throughout. Examples of work in this area are the Temporal Abduction Problem (TAP) [Santos, 1996] and the Probabilistic Temporal Network (PTN) [Young and Santos, 1996].

An alternative to the above approaches is introduced in the next section.

6 A Dynamic Bayesian Belief Network Approach (DBBNs)

A new approach currently being researched and applied to geriatric patient management [Marshall et al, 2000] is that of combining BNs and Survival Analysis to create a Dynamic Bayesian Belief Network (DBBN).

DBBNs are described as generalising the concept of BNs to include a time dimension. The approach represents a stochastic (or probabilistic) process along with causal information [Dean and Kanazawa, 1989], [Russell et al, 1995]. Heckerman et al. [1997] has also introduced a temporal component to BNs by providing a temporal definition of causal dependence where he associates a set of variables indexed by time with each cause and with an effect.

In statistical theory, Markov models are often used to represent stochastic processes. Structured phase-type (Ph) distributions [Neuts, 1989] characterise a type of latent Markov model which provide an intuitive and robust way of describing probabilistic processes. Such models describe duration until an event occurs in terms of a process consisting of a sequence of latent phases - the states of a latent Markov model. For example, duration of stay in hospital can be thought of as a series of transitions through phases such as: acute illness, intervention, recovery or discharge. The representation of such a process in terms of latent phases is realistic, as that is how a domain expert conceptualises the process. It is also mathematically suitable since we can prove that any such statistical distribution may be represented arbitrarily closely by one of phase-type form [Faddy, 1994].

In this approach we combine the advantages of BNs in incorporating prior knowledge and causation into the model with the elegant and intuitive process representation of phase-type distributions (Figure 3).

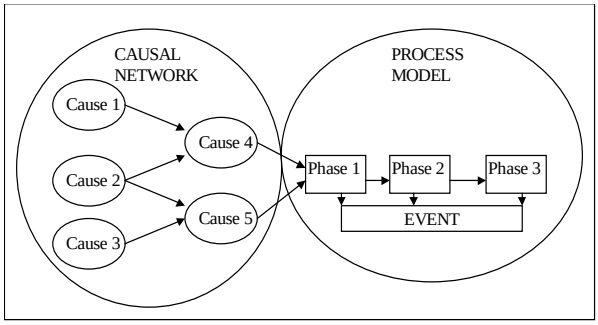


Figure 3: The underlying representation describes a DBBN in terms of a number of interrelated causal nodes which temporally precede and predetermine (in a probabilistic sense) the effect node(s) which constitute the process. The effect node(s) here are characterised by a continuous positive random variable(s), the duration, described by a phase-type distribution

The Causal Network is modelled as a BN. The Process Model may be defined in a manner similar to that of [Dean and Kanazawa, 1989], [Hanks et al, 1995] where we consider an event <E> which initiates a process P at time zero and <P, t> indicates that the process P is active at time t. Then prob<P, t> is the probability that the process is still active at time t. In statistical terminology, prob<P, t> is known as the survivor function, denoted by F(t) and, for a continuous time representation, its derivative f(t) is the probability density function (p.d.f.) of the time for which the process is active. Here we define f(t) by

f(t)□t = prob(process terminates in (t, t+□t) | process is still active at t).

We thus assume that the model includes variables, some of which are qualitative (the causal variables) and some quantitative (the survival variables). Some previous work has been done on data of this sort, mainly involving the introduction of conditional Gaussian (CG) distributions [Lauritzen and Wermuth, 1989], [Friedman, 1998]. We here introduce the idea of Conditional Phase-type (C-Ph) distributions which are more appropriate for process data.

7 A Discussion of Potential Fault Management Applications

Downtime in a network not only results in loss of revenue but can lead to serious financial contractual penalties. It is therefore not surprising that network operators are extremely keen to remedy faults as quickly as possible. To

this end not only is identification of the fault critical but an estimation of a fault's likely life span would greatly assist in managing and assessing maintenance strategies.

Fault management is an important but difficult area of telecommunications network management. Networks produce large amounts of alarm information that must be analysed and interrupted before the faults can be located [Klemettinen, 1999]. As has been stated earlier alarm correlation is the central technique in fault identification [Jackobson and Weissman, 1993].

The instance of a fault can cause numerous alarm events to be raised from an individual network element (NE), this means that the alarms are often interrelated. Also a fault may trigger numerous similar and different alarms to be generated in different NE's up or down stream on the network. For example, the Comms fail alarm, an alarm raised by the management system if it cannot maintain a communications channel to the indicated NE, may cause other alarms such as RS-LOS, RS-LOF, Qecc-Comms_fail, MS-EXC or even laser alarms depending on the fault and configuration.

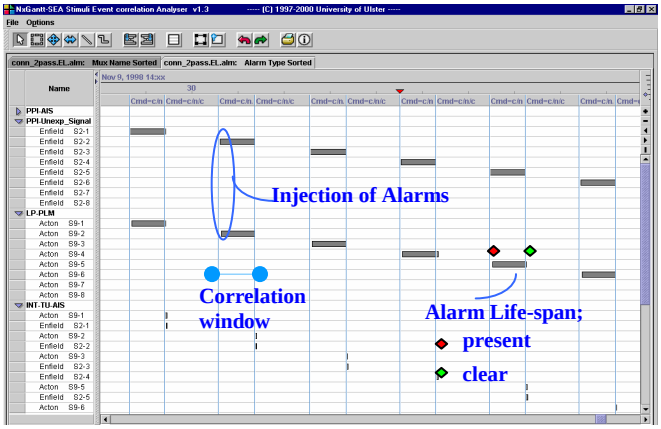


Figure 4: SDH Alarm data viewed over time. Screenshot of NxGantt [Sterritt et al, 2000a] with comments displaying an alarm's lifespan (horizontal Gantt bars), how close the injection of 2 alarms may occur in time and the correlation window.

Correlation serves to diminish the number of alarms presented to the operator, yet ideally the approach should be able to facilitate fault prediction;

- Fault identification/diagnosis - prediction of the fault(s) that have occurred from the alarms present
- Behaviour prediction – warn the operator before hand of severe faults from the alarms that are presenting themselves.

The Bayesian paradigm and its extensions offer the machinery to achieve these ideals. Although methods such as Artificial Neural Networks (ANNs) have been proven to obtain good predictive performance, they do not meet one important goal; that of comprehensibility. Telecommunication companies do not wish to install 'black boxes' into their fault management systems therefore ruling out ANNs [Hatonen et al, 1996]. BN's graphical structure more than meet the need for 'readability'.

When an alarm occurs in a network it is "present" until its accompanying "clear" arrives thus implying a temporal life span (Figure 4) and a correlation window.

BNs, DBNs and DBBNs can all be applied to fault management of telecommunications. Below, we discuss how BNs may be developed, refined by DBNs and further enhanced by DBBNs.

The inducing of this alarm data into a static BN (Section 3) then provides the 'guts' of an expert system, for answering "if then" questions exploring the effects of changing variable values. For example, if Alarm type LP-PLM is observed, this alters the probability (among others) that alarm PPI-Unexpl_Signal will be observed. [Shapcott et al, 1999] and [Sterritt et al, 1998] describes an architecture that induces a BN from this data inferring from it the likely alarm behaviour (Figure 5).

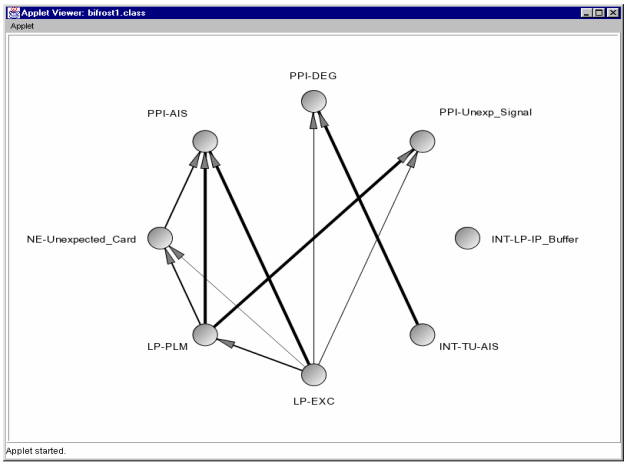


Figure 5: Alarm BN for Fault Management

TBNs or DBNs (Section 4) offer an opportunity to be more precise when predicting the fault by adding a temporal dimension to the model, since the alarms have a lifespan (Figure 4) and the network changes in state under fault conditions. The alarms that are correlated to produce a higher priority alarm may be correlated with other alarms in a later slice – narrowing to a prediction of likely faults.

The DBBN (Section 5) could offer the previously mentioned expert system in the form of the BN with additional benefits of extra predictions of how long until the fault occurs (in the case of behaviour prediction) or how long until the fault is repaired (in the case of fault identification). These additional predictions come from the inclusion of survival analysis into the model.

Once the phase-type distribution has been modelled from suitable available data it may be possible to adapt the DBBN model for more precise fault prediction. The incorporation of phase-type variables within the BN could contribute to a more realistic network where identification of the phase containing the evolving fault episode, would strengthen/weaken the time based prediction options.

8 Conclusion

Bayesian networks have been identified as possible building blocks for creating the knowledge plane. As such this paper has deliberated the Bayesian paradigm and reviewed the literature on its dynamic extensions. It has proposed a new dynamic approach by incorporating survival analysis as part of the model.

Included is a brief discussion on the potential applications of these models for intelligent fault diagnosis in complex telecommunications systems.

The paper has demonstrated the potential power of the dynamic approach for fault identification and behaviour prediction, for instance the ability to determine the likelihood of an alarm being set off at a particular point in time due to a fault occurring at a precise moment in the past.

In fault management there are two real world concerns: (1) the sheer volume of alarm event traffic when a fault occurs; and (2) the cause not the symptoms. The rule-based type systems (monitoring, filtering and masking) used in telecommunication systems address the first. The approaches discussed in this paper would address both concerns paving the way to true intelligent fault management.

References

[Agosta and Crosby, 2003] JM Agosta, S Crosby, “Network integrity by inference in distributed systems”, NIPS Workshop on Robust Communication Dynamics in Complex Networks, 2003

[Aliferis and Cooper, 1983] C. F. Aliferis, G. F. Cooper, “A Structurally and Temporally Extended Bayesian Belief Network Model: Definitions, Properties, and Modeling Techniques”, Proc. 12th Conf Uncertainty in Artificial Intelligence, pp.28-39, 1996.

[Allen, 1983] J.F. Allen, "Maintaining Knowledge About Temporal Intervals", Comms of the ACM, Vol. 26(11), pp.832-884, 1983.

[Arroyo-Figuerao et al, 1998] G. Arroyo-Figuerao, L. E. Sucar, A. Villavicencio, “Probabilistic Temporal Reasoning and its Application to Fossil Power Plant Operation”, Expert Systems with Applications, Vol. 15, pp.317-324, 1998.

[Berzuini et al, 1989] C. Berzuini, R. Bellazzi, S. Quaglini, "Temporal Reasoning with Probabilities", Proc. of Workshop on Uncertainty in Artificial Intelligence, pp.14-21, 1989.

[Bouloutas et al, 1994] A. T. Bouloutas, S. Calo, A. Finkel, “Alarm Correlation and Fault Identification in Communication Networks”, IEEE Transactions on Communication, Vol. 42, No 2/3/4, 1994.

[Clark et al, 2003] D Clark, C Partridge, JC Ramming, JT Wroclawski, “A Knowledge Plane for the Internet”, Proc. Applications, technologies, architectures, and protocols for computer communication, Karlsruhe, ACM SIGCOMM 2003

[Cooper, 1990] G. F. Cooper, "The Computational Complexity of Probabilistic Inference Using Belief Networks", Artificial Intelligence, Vol. 42, pp.393-405, 1990.

[Dagum and Galper, 1993] D. Dagum, A. Galper, "Forecasting Sleep Apnea with Dynamic Network Models", Proc. of Uncertainty in Artificial Intelligence, pp.64-71, 1993.

[Dagum et al, 1992] P. Dagum, A. Galper, E. Horvitz. "Dynamic Network Models for Forecasting", Proc. of the 8th Workshop on Uncertainty in Artificial Intelligence, pp.41-48, 1992.

[Dean and Kanazawa, 1989] T. Dean, K. Kanazawa, "A Model for Reasoning about Persistence and Causation", Computational Intelligence, Vol. 5(3), pp.142-150, 1989.

[Faddy, 1994] M. Faddy, "Examples of Fitting Structured Phase-Type Distributions", Applied Stochastic Models and Data Analysis Vol. 10, pp.247-255, 1994.

[Friedman, 1998] N. Friedman, K. Murphy, S. Russell, “Learning the Structure of Dynamic Probabilistic Networks”, Proc. of the Conf. on Uncertainty in Artificial Intelligence, 1998.

[Hanks et al, 1995] S. Hanks, D. Madigan, J. Gavrin, "Probabilistic Temporal Reasoning with Endogenous Change", Proc. of the 11th Conf. on Uncertainty in Artificial Intelligence, pp.245-254, 1995.

[Harrison, 1994] K. Harrison, "A Novel Approach to Event Correlation", Intelligent Networked Computing Lab, HP Labs, HP-94-68, pp. 1-10, 1994.

[Hatonen et al, 1996] K. Hatonen, M. Klemettinen, H. Mannila, P. Ronkainen, H.Toivonen, “Knowledge Discovery from Telecommunication Network Alarm Databases”, Proc. 12th Int. Conf. on Data Engineering (ICDE’96), pp.115-122, 1996.

[Heckerman et al, 1997] D. Heckerman, J. S. Breese, "A New Look at Causal Independence", Proc. of the 10th Conf. on Uncertainty in Artificial Intelligence, pp.286-292, 1997.

[ITU-T, 1988] ITU-T Recommendations M.3030 "Principles for a Telecommunications Management Network", 1988.

[Jackobson and Weissman, 1993] G. Jackobson, M.D. Weissman, “Alarm correlation”, IEEE Network, 7(6), pp52-59, November 1993.

[Kanazawa et al, 1995] K. Kanazawa, D. Koller, S. Russell, “Stochastic Simulation Algorithms for Dynamic Probabilistic Networks”, Proc. of the 11th Annual Conference on Uncertainty and Artificial Intelligence, 1995.

[Kanazawa, 1991] K. Kanazawa, "A Logic and Time Nets for Probabilistic Inference". Proc. of the 10th National Conference on Artificial Intelligence, AAAI pp.360-365, 1991.

[Kanazawa, 1992]K. Kanazawa, "Reasoning about Time and Probability" Thesis, Brown University, May 1992.

[Kjarulff, 1992] U. Kjaerulff, "A Computational Scheme for Reasoning in Dynamic Probabilistic Networks", Proc. Conf Uncertainty in Artificial Intelligence, pp. 121-129, 1992.

[Klemettinen, 1999] M. Klemettinen, "A Knowledge Discovery Methodology for Telecommunication Network Alarm Databases", PhD Thesis, University of Helsinki, Finland, 1999

[KP Resources, 2004] MIT, 2004 http://www.ana.lcs.mit.edu/peyman/KP/

[Lauritzen and Spiegelhalter, 1988] S. L. Lauritzen, D. J. Spiegelhalter, "Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems", J. R. Statist. Soc. B, Vol. 50(2), pp.157-224, 1988.

[Lauritzen and Wermuth, 1989] S. L. Lauritzen, N. Wermuth, "Graphical Models For Associations Between Variables, Some of Which are Qualitative and Some Quantitative", Annals of Statistics, Vol. 17, pp.31-57, 1989.

[Lekuona et al, 1995] A. Lekuona, B. LaCruz, P. Lasala, "On Graphical Models for Dynamic Systems", Proc., AI and Statistics, pp.317-323, 1995.

[Marshall et al, 1999] A. H. Marshall, S. I. McClean, C. M. Shapcott, P.H. Millard, "Using Bayesian Belief Networks to Predict the Survival of Stroke Patients", Proc. of the IX International Symposium on Applied Stochastic Models and Data Analysis, pp.112-117, 1999.

[Marshall et al, 2000] A. H. Marshall, S. I. McClean, C. M. Shapcott, P.H. Millard, "Learning Dynamic Bayesian Belief Networks using Conditional Phase-Type Distributions", Conference on Principles and Practice of Knowledge Discovery in Databases - PKDD 2000.

[Neuts, 1989] M. Neuts, "Structured Stochastic Matrices of M/G/1 Type and Their Application", Marcel Dekker, New York, 1989.

[Nicholson and Brady, 1994] A. E. Nicholson, J. M. Brady, "Dynamic Belief Networks for Discrete Monitoring". IEEE Trans. on Systems, Man and Cybernetics, Vol. 34(11), pp.1593-1610, 1994.

[Palmer et al, 2000] F. L. Palmer, R. Sterritt, C. M. Shapcott, E. P. Curran, K. Adamson, "Exploring Dynamic Belief Network Visualisation", Conference on Artificial Intelligence and Soft Computing – ASC 2000.

[Provan, 1993a] G. M. Provan, "Model Selection for Diagnosis and Treatment using Temporal Influence Diagrams" Proc International Workshop on AI and Statistics, pp.469-480, 1993.

[Provan, 1993b] G. M. Provan, "Tradeoffs in Constructing and Evaluating Temporal Influence Diagrams", Proc. of the 9th Conference on Uncertainty in Artificial Intelligence, pp. 40-47, 1993.

[Russell et al, 1995] S. Russell, J. Binder, D. Koller, K. Kanazawa, "Local Learning in Probabilistic Networks with Hidden Variables", Proc. 14th Int. Joint Conf. on AI, pp.1146-1152, 1995.

[Santos, 1996] E.(Jr.) Santos, "Unifying Time and Uncertainty for Diagnosis", J. Experimental and Theoretical Artificial Intelligence, Vol. 8, pp.75-94, 1996.

[Shapcott et al, 1999] M. Shapcott, R. Sterritt, K. Adamson, E. P. Curran, "NETEXTRACT - Extracting Belief Networks in Telecommunications Data", Proceedings of the ERUDIT Workshop on Application of Computational Intelligence Techniques in Telecommunication, pp63-71, 1999

[Sterritt and Liu, 2001] 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, Dec. 2001

[Sterritt et al, 1997] R. Sterritt, M. Daly, K. Adamson, C.M. Shapcott, D.A. Bell, F. McErlean, Netextract: An Architecture For The Extraction Of Cause And Effect Networks From Complex Systems, Proc. IASTED Int. Conf. Applied Informatics, Austria, pp 55-57, Feb. 1997.

[Sterritt et al, 1998] R. Sterritt, K. Adamson, C. M. Shapcott, D. A. Bell, "An Architecture for Knowledge Discovery in Complex Telecommunication Systems", (eds) Adey R.A., Rzevski G., Nolan P., Applications of Artificial Intelligence in Engineering XIII, CMP, pp.627-640, 1998.

[Sterritt et al, 2000a] R. Sterritt, E. P. Curran, K. Adamson, C. M. Shapcott, "Visualisation for Data Mining Telecommunications Network data", Data Mining II, (eds.) .F.F. Ebecken, C.A. Brebbia, A. Weigend, WIT Press, Southampton UK, (2000).

[Sterritt et al, 2000b] R. Sterritt, A.H. Marshall, C.M. Shapcott, S.I. McClean, Exploring Dynamic Belief Networks for Telecommunications Fault Management, IEEE International Conference on Systems, Man and Cybernetics, Nashville, Tennessee, USA, 8-11 October 2000, (Volume V) pp 3646-3652

[Sterritt, 2001] R. Sterritt, Discovering Rules for Fault Management, 8th Annual IEEE International Conference on the Engineering of Computer Based Systems (ECBS), Washington DC, USA, April 17-20, 2001

[Sterritt, 2002] 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, Springer-Verlag, 2002

[Young and Santos, 1996] J. D. Young, E. Santos, "Introduction to Temporal Bayesian Networks", Presented at the 7th Midwest AI and Cognitive Science Conf., 1996.

Ontology-based Semantics for Composable Autonomic Elements

John Keeney, Kevin Carey, David Lewis, Declan O'Sullivan, Vincent Wade
Trinity College Dublin
Knowledge and Data Engineering Group
Computer Science Department, College Green, Dublin 2, Ireland
{John.Keeney|Kevin.Carey|Dave.Lews|Declan.OSullivan|Vincent.Wade}@cs.tcd.ie

Abstract

The complexity of modern communication networks requires an autonomic approach, where elements exhibit a degree of self-management which when combined provide a level of self-management for the network as a whole. The heterogeneity of elements however prompts a knowledge driven approach to their definition, composition and management in order to address problems of semantic interoperability. This paper proposes a semantic service based approach to the definition of elements in an autonomic network in order to enable ontological reasoning in support of composable self-management functions.

1 Introduction

The management of computing and communications systems has traditionally been a skilled human task, so 'self-management' is only appropriate if it is overseen or governed in a manner understandable to a human controller. Autonomic communications systems are adaptive networks, the adaptive behaviour of which is governed by human-specified goals and constraints on how the services provided by the network should behave.

The self-management of network elements requires dynamic mapping of human management goals to enforceable policies across a system, with the adaptive network elements reacting to changing context. However, this adaptivity must operate within constraints set by human-specified policies. Accurately mapping these high level policies or governance directives, down to low level adaptation and control policies for individual heterogeneous functional elements poses a challenge network administration and one for which automated solutions remain elusive. It is further complicated by the mapping typically having to occur in the context of a specific service chain or flow within more richly connected network of managed components

This paper introduces a Service-Oriented approach that presents a model of constrainable adaptivity for heterogeneous network management functions. In this model, resources are managed as composable services, called Adaptive Service Elements (ASE), containing inbuilt application-specific adaptivity in its use of sub-services and its subscription to relevant context information streams. In this manner, services can be composed into value chains and workflows while also exposing an elemental resource management view which can form part of an end-to-end resource management activity.

This paper also introduces how ontology-based semantics help address conceptual heterogeneity between services and context and provide a reasoning framework for policy refinement.

2 Semantic Services

Ontology-based semantics [berners-lee], proposed by the Semantic Web initiative, help solve some of the problems of heterogeneity and runtime discovery of service capabilities. Web Service Definition Language (WSDL), a standardised service description language, describes the functional aspects of services and so enables the definition of service operations along with their input and output parameters. However, a richer semantic language is needed in order to reason about services that must be discovered, composed or invoked dynamically. The OWL-based Web Service Ontology (OWL-S) [owls02] uses ontology-based semantics to enhance such web service descriptions. It uses description logic based ontologies, specified in the Web Ontology Language (OWL) [owl], and emerging semantic rule languages to define the Inputs, Outputs, Preconditions and Effects of a service (often abbreviated to IOPE), and in addition describes the resources used by that service. OWL-S provides an unambiguous, computer-interpretable semantic description of a service by providing rich definitions of the IOPEs of a service's operations, as well as a rich set of control specifications for linking constituent services. Through this semantic approach, inference engines (e.g., AI planners and matchmakers) are enabled to automate the discovery, composition, invocation, and monitoring of services [mclraith] despite the use of separately authored ontology models for describing IOPEs and resources.

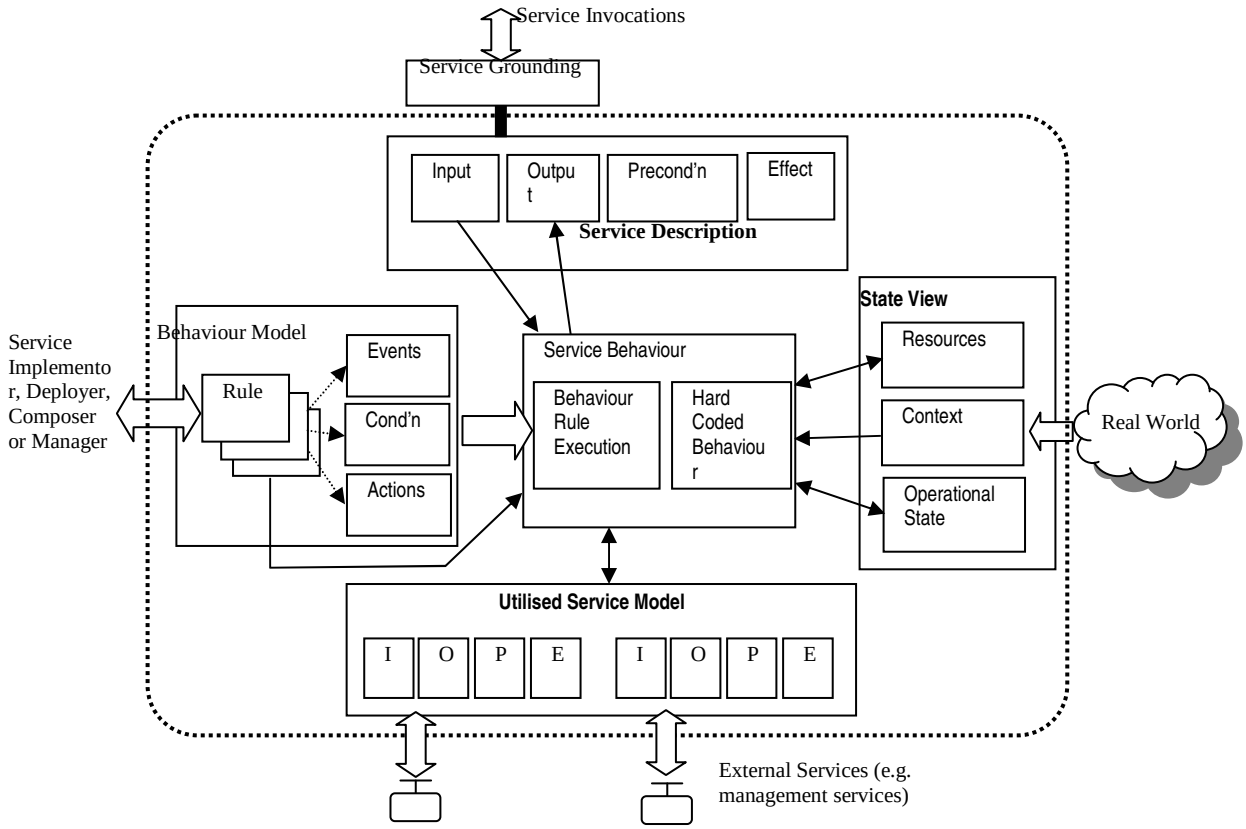


Figure 1: Adaptive Service Element Reference Architecture

3. Semantics of Autonomic Elements

Using a service-oriented approach to the management of autonomic network elements, a service interface is used to provide access to a specific set of resources, where the resources are controlled by the implementation of the service, either solely or shared with other service implementations. We model a service and its behaviour using the abstract concept of an Adaptive Service Element [lewis04] (See Figure 1). This offers a specific service, the behaviour of which is aware both of its local operational context and the characteristics of the network of which it is a part. It is aware of and controls a specific set of resources, which may be modelled as a further set of services.

The adaptive behaviour of an ASE will need to be managed to reflect the goals and requirements of the service users, those people or agents responsible for the service's resources, and the managers who oversee the operation of the network being managed. This management is performed by providing behavioural rules to the adaptive service element. These rules dictate the element's behaviour within the constraints provided by the element's developers, either human designers or automated agents that generate service compositions. Adaptation policy rules, specified as part of the service behaviour model, can be set by the service administrator to adapt how the particular service makes use of particular resources. In this way, the management of such a service-oriented system is achieved by policies local to service, rather than by policies that relate generally to the underlying resources. Such policies can be specified as action, goal or, utility rules. Overall, the responsibility falls on the service developer to expose, via the semantic service specification and behaviour model, all the adaptable interactions between the service and the resources it uses and manages.

Based on this, an ASE is characterised by: a service description; a model of the state observable by the ASE; a description of the services of which it makes use; and a rule-based model for describing and restricting its behaviour component's managed behaviour can be seen as a rule-based automaton. Each service element will also require a OWL-S grounding for each target platform technology that will use it.

Policy refinement is the decomposition of policies relevant to a composite system into a set of policies that are executed in its constituent parts, thereby implementing the behaviour intended by the overall system level policy. In order for even semi-automated policy refinement to be successful, it

is necessary to have access to semantic information about both the high level policy and the service being managed. To enable adaptive systems to process such semantics automatically we adopt ontology based semantics as a means of describing constraints on an adaptive service element's behavioural rules in a machine intelligible form. The expression of these constraint semantics is eased by having the semantics of services and the operational context also expressed in an ontological format.

4 Semantics for Autonomic OSS

Within the Semantic Web initiative it has been widely observed that ontological reasoning techniques will only become beneficial once a sufficiently large number of available services have been semantically marked-up. Similarly in the context of autonomic management, ontology-driven policy refinement will only be of use for autonomic systems once services and networks possess ontological representations.

To arrive at a situation where ontology-based semantics can be fruitfully employed in network operations, we must first move from the current state of the art in communications management technology used in Operational Support Systems (OSS). The predominant paradigm in network management has been the manager-agent model. This models management interface functionality in a fine-grained object-oriented manner, where management functionality is provided by get and set operations on object attributes, and depending on the model used is supplemented by object-level actions and notifications. Functional interface models are defined in terms of Management Information Base (MIB) specifications. Here, the OSI Management and Internet Management represent the two main standards bodies, using the GDMO and SMI languages respectively. Both of these languages, though being potentially generic profiles of ASN.1, were shaped in their usage by the features of the protocols that accompanied them, CMIP and SNMP respectively.

In the 1990s the Distributed Management Task Force defined the Common Information Model schema which was a principled attempt to define management information models for the manager-agent paradigm, but in a way that was independent from the protocol used. This proved successful, quickly becoming a focus for management information modelling standardisation effort, especially in the enterprise management sphere, with support added for a number of protocol bindings including DCE, XML/HTTP and LDAP. The modelling approach was highly object-oriented, yet also incorporated a number of ontological modelling concepts, such as making associations first class concepts with domain and range bindings to classes and allowing class and instance definitions to be freely mixed.

More recently Jorge de Vergara and Victor Villagra [deVergara] have show directly the value of modelling management information models in OWL, and how this can be used to ease the interoperation between models originally conceived in different MIB languages, i.e., GMDO, SMI, CIM.

In parallel, the engineering of service and business layer OSSs for the telecommunication market began to adopt the service-oriented and n-tier component architectures that had come to dominate enterprise computing. At the forefront of attempts to reach industry agreement on modelling such architectures for communications management was the TeleManagement Forum's NGOSS initiative [fleck]. This is attempting to stimulate an open market in telecoms business software component by forming agreements on management information exchanged between business processes and service definitions, via which inter-process invocations can be made. The former encompasses network and element level MIB information as well as service and business level information typically captured in corporate databases. Such business objects also increasingly become the subject of business-to-business e-commerce agreements, e.g. ebXML. This has a natural synergy with the enterprise management model of the DMTF, and the two organisations are now collaborating closely on information modelling. The models for inter-process invocation, termed contracts, are defined in a native XML binding [tmf053] that includes the usual input and outputs as well as preconditions and effects and other service component lifecycle information, e.g. vendor data, deployment setting etc.

It can be seen therefore that the emerging understanding of how semantic web ontology languages may assist in the semantic interoperability of management information models should be naturally reflected in the application of OWL-S to the definition of business application services for the OSS domain. In particular, the technology neutral approach taken in the NGOSS initiative would seem ripe for an ontological approach, provided suitable methodologies and tools emerge [duke]. For this reason our current investigations are moving in this direction, whereby we attempt to re-model existing OSS service components with OWL-S in an attempt to better understand the specific benefits of semantic interoperability and ontology-based conceptual reuse in the OSS software engineering domain.

5 Semantic Reasoning for Autonomics

Though the Adaptive Service Element reference model represents our target architecture for future autonomic communications networks, we acknowledge the need to take a number of exploratory steps in reaching it. This section outlines a number of specific directions currently being explored, and gives initial results where available.

5.1 Dynamic Service Composition

Artificial Intelligence (AI) planning is one technique that is receiving increased attention as a solution for automated service composition and automated adaptivity control. AI planning techniques can automatically generate composite

service plans consisting of simple sequence of actions. Each action can be supported by service invocations, given a set of required goals, a set of possible actions and a description of the initial state of the system. AI planning seeks to represent a relevant part of the world in terms of various states and possible changes that can be made to those states. For example, one branch of planning known as Situational Calculus classifies the functional properties of a service as Inputs, Outputs (states of knowledge of the user) and Preconditions and Effects (world states), which may be available in a semantic description of the service. This rigid approach to world representation allows the usual suite of AI techniques to be applied to a huge range of problems, including automatic service composition and automatic service adaptation. In a further approach, used only for adaptive service composition in [higel03], an analysis of the durative characteristics of services is used to compose services in an intelligent manner. The ASE model will focus on the use of more sophisticated approaches to driving the AI planning mechanisms used, not just to compose services, but to manage the adaptive behaviours of network elements, handled in a service-oriented manner.

5.2 Semantic Management Services

Clearly before we can make good use of AI planning techniques for autonomic communications we need a sufficiently rich set of services from which to compose new services. The ASE promotes a service oriented approach to developing new application components with a policy management interface that make them suitable for use in an autonomic framework. This requires a development approach, where service and management features are closely coupled at design time. As this approach is not currently widespread, we envisage a long period before such a sufficient large population of such components will have been developed to make AI planning viable as an effective adaptive technique.

```
<CLASS SUPERCLASS="CIM_EnabledLogicalElement" NAME="CIM_LogicalDevice">
…
<METHOD CLASSORIGIN="CIM_LogicalDevice" NAME="Reset" TYPE="uint32">
<QUALIFIER TRANSLATABLE="true" NAME="Description" TYPE="string"> <VALUE> Requests a reset … </VALUE> </QUALIFIER>
</METHOD>
…
</CLASS>

<CLASS SUPERCLASS="CIM_LogicalDevice" NAME="CIM_Printer">
…
<PROPERTY CLASSORIGIN="CIM_Printer" NAME="MaxCopies" TYPE="uint32">
   <QUALIFIER TRANSLATABLE="true" NAME="Description" TYPE="string"> <VALUE>The maximum ……</VALUE> </QUALIFIER>
</PROPERTY>
<PROPERTY CLASSORIGIN="CIM_Printer" NAME="PrinterStatus" TYPE="uint16">
   <QUALIFIER TRANSLATABLE="true" NAME="Description" TYPE="string"> <VALUE>Status information for a Printer …. </VALUE></QUALIFIER>
   <QUALIFIER NAME="ValueMap" TYPE="string"> … <VALUE>1</VALUE><VALUE>2</VALUE> … </QUALIFIER>
   <QUALIFIER TRANSLATABLE="true" NAME="Values" TYPE="string">…<VALUE>Idle</VALUE><VALUE>Printing</VALUE></QUALIFIER>
   <QUALIFIER NAME="MappingStrings" TYPE="string"> … <VALUE>MIB.IETF|Printer-MIB.hrPrinterStatus</VALUE> </QUALIFIER>
</PROPERTY>
…
</CLASS>
```

Figure 2: CIM printer data and methods in XML format

```
…
<owl:FunctionalProperty rdf:ID="CIM_Printer_MaxCopies">
   <rdfs:domain rdf:resource="#CIM_Printer"/>   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">"The maximum …"</rdfs:comment>
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>   <rdfs:label>CIM_Printer:MaxCopies</rdfs:label>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="CIM_Printer_PrinterStatus">
   <rdfs:domain rdf:resource="#CIM_Printer"/>  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">"Status information for a Printer …"</rdfs:comment>
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>  <rdfs:label>CIM_Printer:PrinterStatus</rdfs:label>
</owl:FunctionalProperty>
…
```

Figure 3: CIM printer data in OWL format

However, to initiate exploration of this possibility and to study in detail the means by which existing network semantics can be captured and used, we examine the use an ASE management interface that resembles more a semantic version of current manager-agent oriented interfaces, rather than the target policy-oriented interfaces. The approach taken is to use the algorithms described in [deVergara] to extract the class and property information contained in existing information models and then to integrate them with management service models based on OWL-S.

Here we take, as an example, a segment from the DMTF CIM information model for a printer, figure 2, and map to OWL, figure 3. From the information available by mapping the CIM management interface to an ontological format, any access to this data can be reasoned about, with the possibility of the management interface being automatically created. For example, from the excerpts from the Printer Device MOF above, and taking into account the default values of qualifiers not shown, a number of conclusions can be inferred from the ontology, e.g.:

- An operation to read the properties PrinterStatus and MaxCopies are required, but operations to set them are not required since "readonly" is the default for properties.
- Operations to return a descriptive string for each property is required.
- The method reset() is required for the CIM_Printer management interface, since methods propagate to subclasses by default.

This knowledge refers to how one interacts with the model, rather than the semantics of its informational aspects. This can thus be better expressed in OWL-S format, allowing the dynamic creation of such a semantic management interface. A full OWL-S description of this information model segment is outlined in figure 4. It would include a core definition of the process (figure 5) to allow an ontological reasoner, e.g. an AI planner, to reason about the inputs and outputs of such operations, the preconditions and effects of the operations, the types of operations allowed, or how the operations can be composed. A service profile would allow this service to be advertised, e.g. using UDDI, for use in a semantically driven service discovery process. The OWL-S grounding model could then be used in automated invocation of the management service. The example in figure 7 indicated a WSDL grounding, but a grounding to the specific XML and HTTP bindings defined by the DMTF Web Based Enterprise Management standards could be developed and used here equally.

```
<service:Service rdf:ID="CIM_Printer_Service">
   <service:describedBy rdf:resource="http://…/ …  #_Process"/>
   <service:presents rdf:resource="http://…/ … #_Profile"/>
   <service:supports rdf:resource="http://…/ … #_Grounding"/>
</service:Service>
…
```

Figure 4: Top level OWL-S definition for the CIM_Printer Service

```
<process:AtomicProcess rdf:ID="CIM_Printer_Class_getPrinterStatus">
 <process:hasResult>
  <process:Output rdf:ID=" CIM_Printer_Class_getPrinterStatusReturn_OUT ">
     <process:parameterType>"#CIM_Printer_PrinterStatus" </process:parameterType>
  </process:Output>
 </process:hasResult>
</process:AtomicProcess>
…
<process:AtomicProcess rdf:ID="CIM_Printer_Class_getMaxCopies">
  <process:hasResult>
    <process:Output rdf:ID="CIM_Printer_Class_getMaxCopiesReturn_OUT ">
      <process:parameterType>"# CIM_Printer_MaxCopies" </process:parameterType>
    </process:Output>
  </process:hasResult>
</process:AtomicProcess>
…
```

Figure 5: The OWLS Processes that make up the CIM_Printer Service

```
<profile:Profile rdf:ID="_Profile">
  <profile:hasOutput rdf:resource="http://…/_ProcessModel# CIM_Printer_Class_getPrinterStatusReturn_OUT "/>
  <profile:hasOutput rdf:resource="http://…/_ProcessModel# CIM_Printer_Class_getMaxCopiesReturn_OUT "/>
  <profile:hasOutput rdf:resource="http://…/_ProcessModel# CIM_Printer_Class_reset_OUT "/>
…
</profile:Profile>
```

Figure 6: The OWLS outputs profile of the CIM_Printer Service

```
<grounding:WsdlGrounding rdf:ID="_Grounding">
  <service:supportedBy rdf:resource="http://asdfasdf/_Service#_Service"/>
  <grounding:hasAtomicProcessGrounding rdf:resource="#WSDLGrounding__getPrinterStatus"/>
  <grounding:hasAtomicProcessGrounding rdf:resource="#WSDLGrounding__getMaxCopiesReturn"/>
  <grounding:hasAtomicProcessGrounding rdf:resource="#WSDLGrounding__reset"/>
...
</grounding:WsdlGrounding>
```

Figure 7: How the CIM_Printer Service would be grounded by some service described by WSDL

5.3 Policy-based Management for Composite Services

When adaptive services are composed, inevitably the behaviour rule sets grow and become unmanageable. In the policy refinement approach discussed in [carey04] adaptive behaviour rules (high level policies) can be automatically described for a composite service element, specified as finite state machine transitions, which are automatically refined into state transitions for the sub-finite state machines describing the ASE's adaptive behaviour. Here, the use of component behaviour ontologies based on finite state machines can be used to expose just a selected subset of behaviour for policy-based management purposes. We have conducted some preliminary prototyping of a tool for modelling finite state machines using behavioural concept expressed in OWL, e.g. the CIM Printer MIB used in the previous section as depicted in figure 8.

However, as can be seen from the management of complex adaptive systems such as network management systems [murray05], such a discrete state-based model is not sufficient. For an autonomic system to manage a network of adaptive network elements, a more expressive approach is needed to ensure that adaptivity is constrained in a manner where the network operates within an envelope of acceptable behaviour within a certain behaviour space [dobson04], rather than the fixed and restrictive manner described.

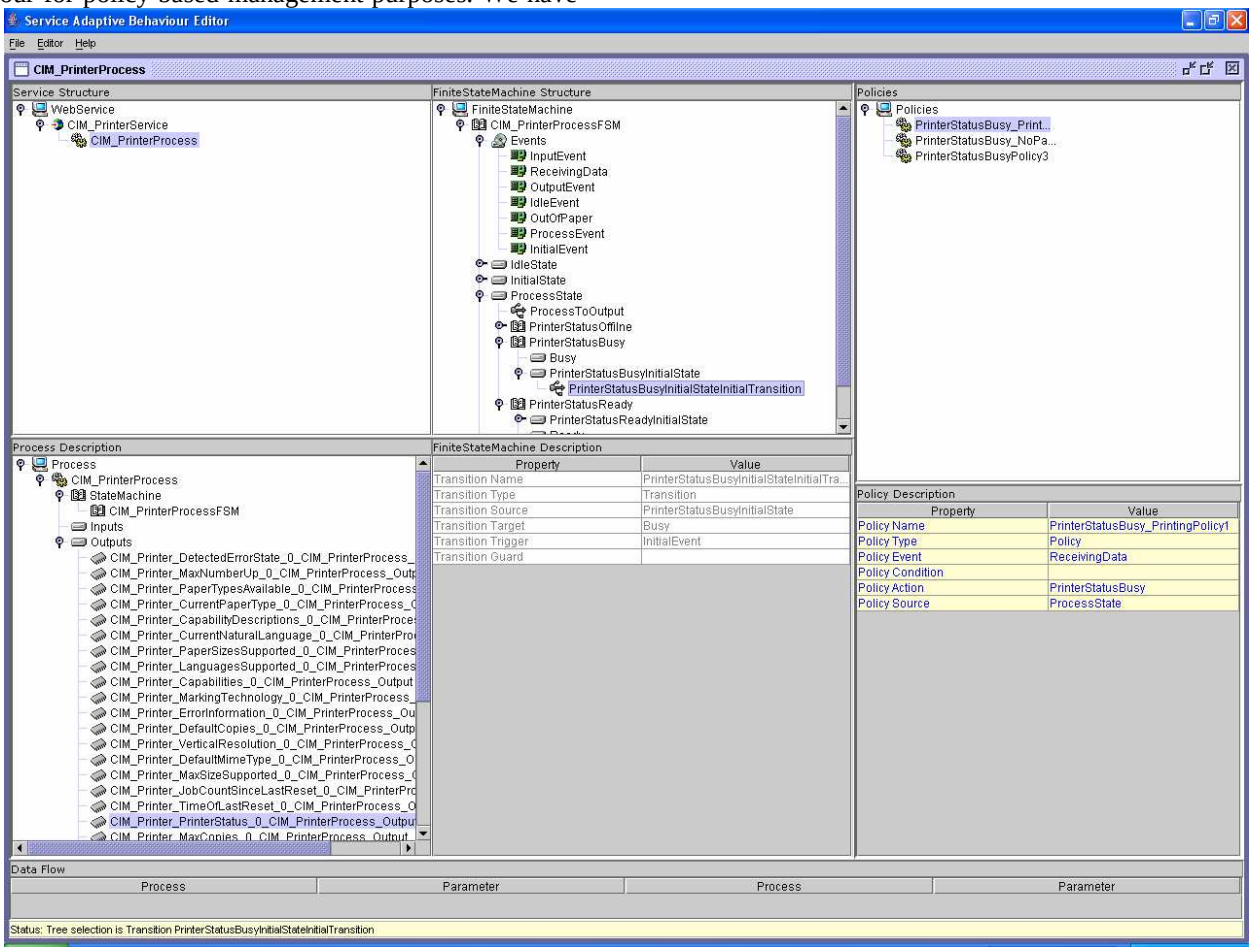


Figure 8: Automatic creation of management policies based on OWLS specifications of services

5.4 Semantic Interoperability

These proposed service-oriented ASEs allow interaction through well defined service interfaces, thereby allowing an adaptive communications framework to be constructed from elements sourced from any number of developers. There will, however, be a need for semantic interoperability through the resolution of semantic mismatches which will be inevitable due to the nature of differing vendor developments. It is expected however that resolution of these conflicts will be supported through the use of tools and processes which support the creation/discovery of mappings between ontologies. Substantial research has been ongoing into the area of semi-automatic techniques for mappings [osullivan03] but very little research has been undertaken into its applicability to autonomic systems. In particular we will study the extent to which semi-automated mapping approaches to semantic interoperability are sufficient in fulfilling the needs of autonomic systems and we will develop appropriate solutions.

5.5 Knowledge Delivery Network

In an autonomic computing environment, ASE's may be adaptable, but they their adaptation must be driven by both local context and network context. However, difficulties arise when heterogeneous elements must provide possibly complex end-to-end service provision chains over an adapting network topology. This heterogeneity leads to increased human costs to manage connections for information exchange. This can be alleviated by the provision of an active Knowledge Delivery Network to replace the standard passive information retrieval model. A Semantic Query Based Network is described in [lewis04] that uses a publish/subscribe paradigm from Content Based Networking to support the dissemination of ontologically defined knowledge. Such a model can be further expanded to act as a suitable Knowledge Delivery Network, with the semantic interoperability effort invested in the delivery network for use not just for managing the network but for other applications using the autonomic communications framework. Such an approach raises several issues, including ensuring suitable access control in multi-organisational setting, and where semantic interoperability functions may best be located.

6. Conclusions and Further Work

To conclude, this paper proposes a semantic service based approach to the definition of elements in an autonomic network in order to enable ontological reasoning in support of self-management functions. We are currently working on examining the use of semantics for various parts of the ASE reference mode. Specifically we are using OWL classes and properties derived from existing management information models as the core concepts for defining ASE state, both its resources and context, which is then also used in defining finite state machine definitions for the service adaptive behaviour. This adaptive behaviour model can then be used as the basis for defining run-time policies constraining the behaviour of the ASE. We are also using this MIB derived OWL model as the basis for input and output of OWL-S services providing management capabilities. This offers management capability in the more traditional manager-agent mode of interaction, but also offers the possibility of auto-generating semantic management services from MIB definition in a form that can be fed into an AI planner, so that composite management operations can be generated dynamically.

In addition we are using OWL-based management semantics as the basis for examining run-time semantic interoperability both for semantic service invocations and for content-routed semantic notifications in the knowledge delivery service.

Acknowledgments

This work has been partly sponsored by Science Foundation Ireland through the Centre for Telecommunications Value Chain Research, and partly by the Irish Higher Education Authority through the M-Zones programme.

References

[abelson *et al*., 1985] Harold Abelson, Gerald Jay Sussman, and Julie Sussman. *Structure and Interpretation of Computer Programs.* MIT Press, Cambridge, Massachusetts, 1985.

[berners-lee] Berners-Lee, T., Hendler, K., Lassila, O. (2001), *The Semantic Web*, Scientific American, pp 35-43, Issue 284 (3), 17th May 2001

[carey04] Carey, K., Lewis, D., Higel, S., Wade, V., *Adaptive Composite Service Plans for Ubiquitous Computing*, 2nd International Workshop on Managing Ubiquitous Communications and Services (MUCS2004), Dublin, Ireland, November 2004.

[deVergara] de Vergara, J.E.L., Villagrá, V.A., Berrocal, J., *Applying the Web Ontology Language to management information definitions*, IEEE Communications Magazine, Vol. 42, Issue 7, July 2004, pp. 68-74. ISSN 0163-6804

[dobson04] Dobson, S. and Nixon, P., *More principled design of pervasive computing systems*. In Proc Engineering for Human-Computer Interaction and Design, Specification and Verification of Interactive Systems.Hamburg, Germany, July 2004.

[duke] Duke, A., Davies, J., Richardson, M., Kings, N., *A Semantic Service Oriented Architecture for the Telecommunications Industry*, in proc of Intelligence in Communications Systems, (INTELLICOM 2004), Bankok, Thailand, Nov 2004, LNCS 3283, pp 236-245

[fleck] Fleck, J, *An Overview of the NGOSS Arhcitecture*, TeleManagmeent Forum Whitepaper, May 2003

[higel05] S. Higel, T. O'Donnell, D. Lewis, V. Wade, *Towards an Intuitive Interface for Tailored Service Compositions*, Proc. Distributed Applications and Interoperable Systems, 4th IFIP WG6.1 International Conference, DAIS 2003, Paris, France, November 17-21, 2003, LNCS 2893. pp 266-273

[lewis04] D. Lewis, T. O'Donnell, K. Feeney, A. Brady, V. Wade, *Managing User-centric Adaptive Services for Pervasive Computing*, Proc. IEEE International Conference on Autonomic Computing (ICAC'04), May 17th-18th 2004, New York, USA.

[mclraith] McIlraith, S.A., Son, T.C., Honglei Zeng, H. (2001), *Semantic Web Services*, IEEE Intelligent Systems, 16(2), March/April 2001

[murray05] K. Murray, D. Pesch, *Revenue Optimisation and User Prioritisation using Call Admission Control Strategies in Multi-service 4G Cellular Wireless Networks*, in Proc. European Wireless 2005, Cyprus, April 2005

[osullivan03] D. O'Sullivan, D. Lewis, *Semantically Driven Service Interoperability for Pervasive Computing*, Proc. 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access, San Diego, USA, 19 Sept 2003, ACM Press.

[owl] World Wide Web Consortium (W3C), *Web Ontology Language (OWL)*, www.w3.org/2004/OWL/, Visited Mar 2005

[owls02] The DAML Service Coalition, *OWL-S: Semantic Markup for Web Services*. http://www.daml.org/services/, October 2002

[tmf053] TeleManagement Forum, *NGOSS Technology Neutral Architecture Release 4.0*, January 2004,

Hybridising events and knowledge in an infrastructure for context-adaptive systems

Simon Dobson
Systems Research Group
School of Computer Science and Informatics
University College, Dublin IE
simon.dobson@ucd.ie

Abstract

Event-based systems are a popular substrate for distributing information derived from sensors to be used in driving adaptive behaviour. We argue that event systems only provide a poor model of context, and that a hybrid approach that uses events to populate and maintain a knowledge base provides a more stable solution. The inherent uncertainties imply that traditional knowledge-based system techniques are extended to deal with more uncertain reasoning. We discuss our plans for additional work in analysing and programming autonomic behaviours with this architecture.

1. Introduction

Autonomic systems are intended to adapt to their environment in a way that optimises performance, robustness and other features without requiring extensive human intervention. The challenges arise from the need to deal with complex and uncertain information about the environment, and to match this to appropriate changes in system behaviour.

In this position paper we describe the motivations for our current work within SRG on infrastructures and languages for adaptive systems. We argue for hybrid approaches, using an event-based infrastructure to drive and maintain a knowledge base. The resulting system may be programmed in both event-based and knowledge-based terms, allowing a range of approaches to adaptation to be explored.

Section two describes current approaches to building adaptive systems, highlights some deficiencies and argues for a hybrid model that combines event- and knowledge-based approaches. Section three briefly discusses some issues in programming such hybrids, while section four concludes with some directions for future work.

2. Context, events and knowledge

Designs for autonomic systems draw their inspiration from a number of sources. Prominent among these are biologically-inspired systems built around stygmergy or swarm intelligence, where simple individual responses to stimuli are aggregated to produces a global result [Bonabeau99]. At the other extreme are attempts to model adaptive systems in a closed-form way that allows more precise characterization of their behavioural envelopes [Dobson04]. The former relies on ideas from control theory, while the latter draws more on pervasive computing, continuous mathematics and AI.

The context of a system captures the environment in which it operates, including all “additional” or “non-functional” aspects that, while not being “core” to the system’s behaviour, nevertheless affect the way in which that behaviour should be optimised. Pervasive computing systems are good representatives of the

class of adaptive systems whose adaptations are constrained by their surrounding environment.

The Context Toolkit [Salber99] is the canonical example of programming pervasive applications based on events. Such systems consist of a number of adapters or *contextors* [Coutaz02], each capturing some aspect of the environment such as the temperature, or the reading from a location sensor. The advantage of such systems is that it is straightforward to construct both the infrastructure and the adapters; the disadvantage is that they place a large load on the developer to build a sufficiently flexible decision-support system to drive adaptative behaviour.

Why events and adaptation don't match

To understand the problem of using events directly, consider the following scenario. Suppose we have two people, `A` and `B`, together with a room `R`. Two events are defined, `enter(a,b)` and `leave(a,b)`, indicating that entity `a` has entered (or left, respectively) place `b`. These events are to be used to drive a system that will adapt its behaviour when `A` and `B` and in `R`. We use angular brackets to denote event traces: given events `e1`, `e2`, and `e3` we use `<e1,e2,e3>` to denote the sequence of events occurring in the order given and `<e1,…,e2>` to denote `e2` occurring after `e1` with zero or more events in between.

In the simplest model there are two possible event traces that can bring the desired situation about: `<enter(A,R),…,enter(B,R)>` or `<enter(B,R),…,enter(A,R)>`. On observing either of these event traces the system may adapt.

The problem, however, is that this approach is only stable given three key assumptions. The first is that events cannot be "counteracted" by other events. Suppose we observe the event trace `<enter(A,R),…,enter(A,S),…,enter(B,R)>`. Does `A` entering `S` mean that `A` is no longer in `R`? – in other words, are `R` and `S` disjoint spaces? This cannot be definitively answered without an understanding of the spatial relationships involved.

Furthermore in a open system we might introduce new events which interact with existing events in unforeseen ways. Introducing an event `leave(a,b)` (with the obvious intention) means that an event trace such as `<enter(A,R),…,leave(A,R),…, enter(B,R)>` is also not a valid trigger for adaptation.

The second problem concerns triggers that rely on a correspondence between events. Suppose we see the event trace `<enter(A,R),…,enter(B,R), …,enter(B,R)>` – what do we conclude? Should the second `enter(B,R)` event be considered a duplicate, an error, or the start of another trigger for which we should wait for a corresponding `enter(A,R)` event?

This leads directly to the third problem. Event systems were developed from process algebra which in turn describes processes that might be termed *exact*: the events that occur are assumed *actually to have occurred*. The problem with many pervasive (and other) systems that have a close connection to the real world, for example by way of sensors, is that the processes they are engaged in are *inexact*: the events may be noise.

It seems intuitively likely, absent any intervening `enter()` or `leave()` events, that the second `enter(B,R)` event is a duplicate. However, knowing this implies an enormous amount of knowledge about the structure of the real world and the external semantics of events. Moreover, *encoding* this knowledge in a way that will be suitable for triggering an adaptation seems likely to be inordinately complicated for any realistic case.

Although simple, these cases would defeat most event-algebra systems (for example the one described in [Hayton96]). We hypothesise – without any formal justification – that the twin problems of openness and noise render such algebraic systems intractable.

The conclusion we may draw is that, while event systems may be scalable from a systems perspective, they are decidedly *not* scalable from a programmer's perspective.

The problem is that events are being used to two disjoint purposes. On the one hand, events are used to indicate that "something happened" (albeit with some uncertainty); on the other hand, event traces are being used as the system's model of the outside world. The former is a system-level issue that is handled well by events; the latter is a semantic-level issue that is not. If we decouple the two, we may develop a hybrid system having the disadvantages of neither.

A more knowledge-driven approach

We may observe that many adaptive systems decisions are phrased in logical terms: "when A and B are in the room then…". We might therefore import techniques from knowledge-based systems to drive adaptations when particular conditions are true.

This approach has given rise to other contextual systems, for example [Wang04] using RDF to represent knowledge. Several programming techniques are then possible, including the use of truth-maintenance techniques to execute adaptation code when a predicate changes truth-state.

Such techniques face twin problems of uncertainty and noise. Most information derived from sensors is inherently error-prone, and sensors give rise to incorrect observations. To take one example, several authors have used RFID sensors to observe tags attached to people or artifacts. However, RFID sensors will fail to spot some tags, perhaps because it is moving too slowly to activate. They will also sometimes mis-identify tags because of interference. This means that a sensor-derived event may be incorrect or may be missed. It is easy to see why event traces are such an inadequate source of modeling.

However, it is possible to use a knowledge base as a stabiliser on the context model. Events must be treated as evidence for a fact, rather than as true Boolean values. We may then use techniques such as Bayesian probability, fuzzy logic or Dempster-Schaffer evidence theory to combine individual pieces of evidence into a more confidently-held view of the environment, which can then in turn be used to drive adaptation. This helps combat the danger of a system changing state dramatically as a result of a single, erroneous event, since other already-accepted evidence can act as a counterweight. Adding more knowledge of about the system (such as the behaviour of people in space) further increases this stabilisation effect.

3. Programming hybrids

This leads to a hybrid model in which an event infrastructure is used to collect and distribute evidence for the state of the system's environment, with the evidence being used to populate a knowledge base that maintains levels of confidence (or uncertainty) about that environment.

What sort of applications can be built on such a system? This is one topic of our current research. However, the nature of the available information provides some constraints.

The first observation is that all decisions are necessarily tentative. Uncertain reasoning approaches may allow a system to maintain an on-going level of confidence about its environment. Having a confidence interval makes such systems sensitive to small changes: a small change in evidence may cause the decision-making process to "tip". It remains the case, however, that many adaptation decisions are "crisp", so that the uncertain reasoning collapses to Boolean logic when the decision is made. This uncertainty means that we need to maintain one or more recovery strategies for any adaptation or decision the system makes, since each may need to be undone for at least two reasons: because circumstances change to cause a new adaptation, or because the additional evidence shows the initial adaptation to have been mistaken.

A second observation is that adaptations are not arbitrary: systems do not change from one behaviour to another, completely unrelated behaviour, but rather change within an envelope according to environmental changes. A core task for engineering autonomic systems is to ensure that all adaptations do indeed remain within the

design envelope and do not take the system to unacceptable parts of the behavioural space.

Finally, while autonomic systems of this type can make use of significant bodies of existing AI research, the levels of noise and uncertainty, coupled with the degree of unsupervised operation required, do seem to pose genuinely novel challenges. We believe that there are several foundational innovations to be made in the logics and reasoning approaches used to describe autonomic adaptation, as well as in the way this reasoning is used to select adaptive behaviour. In particular, we are becoming convinced that approaches that account for the entire system behaviour at once may have advantages over those which try to coalesce a number of individual independent adaptations. In a sense this is the difference between set theory and topology: we believe that topological approaches may prove useful both the analyzing and programming adaptive systems.

4. Future work

We believe that a combination of event-handling and knowledge management – distributed systems combined with AI techniques – offers a useful hybrid approach to modeling the context of adaptive systems. The knowledge base provides an important gain in the expressive power of the system in the face of erroneous events. The partial and tentative nature of all such knowledge means that programming techniques must make extensive use of uncertain reasoning and other AI-derived techniques. We further believe that it is important to move away from one-adaptation-at-a-time engineering to adopt a more holistic, closed-form approach to describing, analyzing and programming adaptive behaviours.

Our work in this area is following three complementary strands. From the systems perspective, we are developing a hybrid event and knowledge context system and evaluating different strategies for distributing and maintaining knowledge. From a programming perspective, we are exploring a range of programming models using combinations of events and knowledge. Another area of interest is whether we can use failure and noise constructively to drive computation[1]. Underpinning these activities is work on the semantics of adaptive systems: what exactly does it *mean* to be adaptive, and how can we capture the "shape" of that behaviour.

Acknowledgements

A number of members of SRG are actively and invaluably contributing to pursuing the ideas presented here, especially Graeme Stevenson, Sergey Tsvetkov, Lorcan Coyle, Steve Neely, Abdur Razzaque, Joe Kiniry and Paddy Nixon.

References

[Bonabeau99] Eric Bonabeau, Marco Dorigo and Guy Theraulaz. Swarm intelligence: from natural to artificial systems. Oxford University Press. 1999.

[Coutaz02] Joëlle Coutaz and Gaëtan Rey, Foundations for a theory of contextors. In Computer-aided design of user interfaces 3, pp. 13–34. Christophe Kolski and Jean Vanderdonckt (eds). Kluwer. 2002.

[Dobson04] Simon Dobson and Paddy Nixon. More principled design of pervasive computing systems. Proceedings of Engineering for Human-Computer Interaction and Design, Specification and Verification of interactive Systems (EHCI-DSVIS'04). Springer-Verlag. 2004. To appear.

[Hayton96] Richard Hayton, Jean Bacon, John Bates and Ken Moody. Using events to build large-scale distributed applications. Proceedings of the 7th ACM SIGOPS European workshop: Systems support for worldwide applications, pp. 9–16. ACM Press. 1996.

[Salber99] Daniel Salber, Anind Dey and Gregory Abowd. The Context Toolkit: aiding the development of context-enabled applications. Proceedings of CHI'99, pp. 434–441. 1999.

[Wang04] Xiaohang Wang, Jin Song Dong, Chung Yau Chin, Sanka Ravipriya Hettiarachchi, Daqing Zhang. Semantic Space: an infrastructure for smart spaces. IEEE Pervasive Computing 3(3), July–September 2004. pp. 32–39.

[1] This work is due to Jake Beal of MIT.