

University of St Andrews
School of Computer Science

Distinguished Lecture Series 2003/04

*Wednesday 14th April 2004
John Honey Building
North Haugh St Andrews*

**Autonomic Computing as a Unifying
Framework for Self-Managing Systems**

by

*Ric Telford
Director of Autonomic Computing
Architecture and Standards
IBM*

Computing technology has progressed rapidly over the last several decades with implementations and applications that were unthinkable a decade ago now commonplace. The rate of progress, however, has brought its own cost. As large IT infrastructures grow more complex the cost of managing these systems has increased rapidly. As a result a greater percentage of the IT budget is going toward maintenance of the infrastructure rather than improving its benefit to the business. The complexity of such a computing infrastructure requires that the environment become more “autonomic” -- that is, self-managing.

Developing self-managing computing resources is not a new problem for computer scientists. For decades system components and software have been evolving to deal with the increased complexity of system control, resource sharing, and operational management. The advent of the Internet and dramatically increased price performance of information technology in the last few years has led to a huge growth in the scale and complexity of computing systems. Autonomic computing is the next logical evolution of these past trends to address the increasingly complex and distributed computing environments of today.

This series of lectures will describe IBM’s vision for autonomic computing and the plan to apply the model of autonomic systems to self-managing systems with a focus on such topics as problem determination, configuration, and optimization.

*School of Computer Science, University of St Andrews, North Haugh, St Andrews, Fife KY169SS.
Tel No. 01334 463253. <http://www.dcs.st-and.ac.uk>.*

Biography

Ric Telford's professional business career highlights 20 years of software development experience and is noted for bringing innovative approaches to the design and development of key software technologies. Telford joined IBM in 1983, as a developer for PROFS in their software lab in Dallas, Texas. Prior to the acquisition of Lotus, Telford led much of the office systems development for IBM, including distributed calendaring and groupware products. During his tenure at IBM, Telford has played a number of key roles in various software initiatives for IBM, including the imaging products unit, networking and security software, and software mobility products. Ric tends to be at the forefront of emerging technologies at IBM. He served as Director of Technology for the IBM CIO, responsible for the development, implementation and adoption of technologies that hastened the transformation of IBM into an e-business. Ric was the Director of Technology for Intelligent Infrastructure, the precursor in IBM to "e-business on demand". Most recently, Ric was responsible for defining and delivering software solutions for the service provider market, also known as "xSPs. In his current assignment, Ric is responsible for defining and delivering the architecture, technology and standards for "Autonomic Computing." Autonomic Computing is the set of capabilities required to make a computing system more self-managing, much like the human autonomic system. Ric works across IBM (including servers, software and storage) and the industry to develop an end-to-end, open architecture solution for self-managing systems. Ric holds a Bachelor of Science degree in Computer Science from Trinity University in San Antonio, Texas, graduating magna cum laude and Phi Beta Kappa. He holds several U.S. Patents.

Programme

10.15 – 11.15 Lecture 1: The Autonomic Computing Vision

The concept of “Autonomic Computing” takes its name from the human autonomic nervous system. Several years ago, Paul Horn, Sr VP of Research at IBM, made a “call to arms” to the IT industry to start focusing on the complexities of IT by borrowing from the human “self-managing system.” At the current rate of growth, the cost of maintaining and managing IT infrastructure will soon become unaffordable. Autonomic Computing offers an approach to address the costs associated with managing IT.

This lecture presents the concept of Autonomic Computing, the business drivers behind it, and the value that a self-managing infrastructure provides. Included in this is the set of constructs and tools that will hasten the adoption of autonomic computing technologies.

11.15 – 11.45 *Coffee*

11.45 – 13.00 Lecture 2: The Autonomic Computing Architecture

In order to fulfill the promise of self-managing systems, a comprehensive architecture is required. The architecture needs to be abstract enough to allow for adaptation to various environments, but prescriptive enough to ensure interoperability across heterogeneous systems. Finally, the architecture must lend itself to standardization across the industry to ensure broad-based adoption.

This lecture describes the overarching architecture for Autonomic computing as being put forth by IBM. The architecture defines the basic constructs of an autonomic computing system, and the interactions of these constructs. The architecture presumes the existence of some “core technologies”, which will also be discussed. Finally, the topic of open standards will be covered, with an overview of the emerging standards in the autonomic computing space.

14.30 – 15.30 Lecture 3: Autonomic Computing in Action

Lectures 2 covered an abstract view of an autonomic computing system, describing the architectural elements, formats, protocols and interfaces. Although it is important to understand this as background, it is much more interesting to examine Autonomic Computing using real scenarios.

This lecture describes some examples of autonomic computing “in action” at the system level. It will discuss how the architectural elements are instantiated and give examples of possible flows between elements. The scenarios discussed will cover a broad range of

autonomic capabilities including examples of self-healing and self-optimizing systems.