SOME ASPECTS OF AUTONOMIC COMPUTING SYSTEMS USER INTERFACES

Geru Vitaly

Institute of Mathematics and Computer Science, Academy of Sciences of Moldova
Academiei str.5, Chisinau MD-2028, Moldova

Abstract: The main goal of autonomic computing is to make systems more simple. Even so the human presence will not be completely removed. System administrators maintaining and monitoring large autonomic systems, users making their usual works helped by autonomic personal computers, humans will always be a part of computational process. Autonomic actions performed by the system must be understandable and user should be capable of reviewing and changing the state of the system. In order to achieve this level of transparency different scale of system state details will be required. The ability of system to explain what is happening in the situation that most of actions and decisions are made autonomously is a serious issue. This paper presents some aspects and concepts of autonomic computing systems user interfaces taking into account autonomic nature of them.

Key words: autonomic computing, user interface, human-computer interaction

INTRODUCTION

The existing computing systems are often large and complex. Human management of theses systems requires extensive and highly skilled IT staff. There is a need for computing systems to take care of themselves at a higher level. Autonomic computing is a new paradigm of computing in the core of which is the ability of the systems to manage themselves, adjust to varying circumstances and handle their resources most efficiently. A helpful analogy with human autonomic nervous system can be made [2]. Autonomic nervous system adjusts breathing rate, tell your heart how fast to beat, adjusts your blood flow, etc.

From the user perspective as systems become more autonomic, they often become less understandable. Therefore if we are not careful in developing autonomic computing systems user interfaces then automation can increase system complexity, decrease system transparency and limit opportunities for human-computer interactions.

USER INTERFACES AND TOOLS OF AUTONOMIC COMPUTING SYSTEMS

In the paper [2] the following eight characteristics of autonomic computing systems are outlined:

1) To be autonomic, a system needs to "know itself" – and consist of components that also possess a system identity.
2) An autonomic system must configure and reconfigure itself under varying and unpredictable conditions.
3) An autonomic system never settles for the status quo – it always looks for ways to optimize its workings.
4) An autonomic system must perform something akin to healing – it must be able to recover from routine and extraordinary events that might cause some parts to malfunction.
5) A virtual world is no less dangerous than the physical one, so an autonomic computing system must be an expert in self-protection.
6) An autonomic computing system knows its environment and the context surrounding its activity, and acts accordingly.
7) An autonomic system cannot exist in a hermetic environment (and must adhere to open standards).
8) Perhaps most critical for the user, an autonomic computing system will anticipate the optimized resources needed to meet a user's information needs while keeping its complexity hidden.

In order to have these characteristics future autonomic computing systems will have four fundamental features: self-configuring, self-healing, self-optimizing and self-protecting.

The evolution to autonomic computing can be conceptually divided into five levels [1]: basic (represents the starting point), managed, predictive, adaptive, autonomic. With the evolution of systems the user interfaces will also change. For different levels specific set of tools will be used:

- Basic – local, platform and product specific;
- Managed – consolidated resource management consoles with problem management system, automated software install, intrusion detection, load balancing;
- Predictive – role-based consoles with analysis and recommendations; product configuration advisors; real-time view of current and future IT performance; automation of some repetitive tasks; common knowledge base of inventory and dependency management;
- Adaptive – policy management tools that drive dynamic change based on resource specific policies;

- Autonomic – costing/financial analysis tools, business and IT modeling tools, tradeoff analysis; automation of some e-business management roles.

The idea of autonomic computing is not entirely new. Computer systems have been managing their resources for years: memory is monitored for errors and corrected on the fly, communication channels are monitored for mistakes which are corrected when they arise without human intervention, and RAID systems anticipate failures by distributing stored content over multiple independent disks with some redundancy.

In the article [3] some cases of computing systems with autonomic parts are examined: Aegis missiles defense system, airplane control systems that compensate the gradual deterioration of some subsystems, spell-checking and automatic correction facility of Microsoft Word, Microsoft Windows browser, RAID (redundant array of independent disks) disk system. From these cases and analogous ones the following conclusions regarding the design of user interfaces can be drawn:

- Autonomic applications have to provide reliable information in different situations in order to ensure proper operation in unpredictable environments;
- Required data should be accessed easily by the user;
- User attention is a vital resource: autonomic computing systems need to know what and how to tell the user in particular situations. Either extremes, telling or hiding everything, should be avoided. In different situations an appropriate level of details should be chosen. This will require multi-layer data model and information synthesis methods to be used;
- Carefully tuning autonomic interactions without disturbing user from primary tasks;
- The possibility to roll back autonomically driven actions;
- The ability to describe behaviors in terms of goals, rather than in terms of actions;
- Watch what a user does and tune the user interface for better human-computer interaction;
- Changes of the system have to be logged, especially autonomic ones;
- When trying to anticipate error situations, it is important to take into account at the designing level of user interfaces not just hardware or software failures, but also potentially incorrect human intentions (or user attempts to destabilize the system).

CONCLUSIONS

Autonomic computing systems will require great attention at the designing of user interfaces. System behavior must be understandable, or systems and users cannot work together successfully. In order to design efficient user interfaces of such systems cognitive human abilities of understanding autonomic systems should be examined. Multi-layer data model and methods of information synthesis could be useful in the designing of autonomic computing systems user interfaces.

REFERENCES:

[1] A. G. Ganek, "The Dawning of the Autonomic Computing Era," *IBM Systems Journal*, V. 42, N. 1, pp. 5–18 (2003).

[2] P. Horn, *Autonomic Computing: IBM's Perspective on the State of Information Technology*, IBM Corporation (2001) http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf.

[3] D. M. Russell, P. P. Maglio, R. Dordick, and C. Neti, "Dealing with Ghosts: Managing the User Experience of Autonomic Computing," *IBM Systems Journal,* V. 42, N. 1, pp. 177–188 (2003).