Potential formal models for autonomic computing applications

Todor Stoilov, Krasimira Stoilova

Abstract: An analysis of the potential formalism, used in the hierarchical system theory is performed, addressing the autonomic computing system design. In that manner, self management of the overall autonomic system can be achieved by influencing the local resources and the goals of the local computing subsystems. Thus, a quantitative formalization of the control processes is used for the autonomic self-management.

Key words: Autonomic computing systems, control theory, hierarchical system theory, optimization

INTRODUCTION

The term Autonomic Computing was introduced as an initiative of IBM in 2001 in order to reduce the complexity of the management of computer systems and to explore the possibilities to introduce self management and self organizing intelligent systems. The autonomic computing is a tendency to cope the dynamical behaviour of growing scale of computer systems, the increase of the complexity of the system management, exploitation and the choice of the operating systems parameters [14]. The benefits of the autonomic computing are expected to influence the efficiency of the computer system exploitation and the quality of the services, which the users exploit. The idea of the autonomic computing was inspired by the human body’s nervous system [2]. Commonly, the nervous system is managing the different parts of the body without special efforts from the human brain. The autonomic functionality of the body is the model, which is targeted for the management of disperse computing systems.

The Autonomic Computing behavior is decomposed by IBM to four general properties, concerning the system self management: self-optimizing, self-configuring, self-protecting and self healing. Such a computer management has to satisfy real-time requirements towards user applications and quality of services [8].

The paper gives short analysis of the behavior of the computing systems. The autonomy can be achieved, applying methods from the control system theory. Due to a quantitative assessment of computer behavior, appropriate closed loop control policy can be implemented as self- organizing and self-optimizing computer polices. The integration of the autonomic behavior of the different computer parts is possible by application of coordination theory. A particular self-optimizing behavior in dynamically changed workload is targeted by the application of the non-iterative coordination policy.
EXAMPLES OF AUTONOMIC BEHAVIOUR OF COMPUTING SYSTEMS

The Oceano project [7] concerns multi-customer hosting over a virtualized collection of hardware resources. The computing utility infrastructure consists of a farm of massively parallel densely-packed servers, interconnected by high-speed switched LANS. The computer farm is managed, supporting self-optimization, component monitoring, autonomic resource distribution under user demands.

In [11] Q-Fabric system, supporting continuous on-line quality management, is presented. It is regarded as a continuous process that involves monitoring and controlling the use of resources, reacting to dynamic changes in user requirements, to resource availabilities, to system anomalies for efficient resource utilization. The Q-Fabric system performs self-organization features for the on-line management, on-line optimization for quality of user services.

In [5] a large data storage system “OceanStore” is presented. The system is introduced for its management components like self-configuration, self-optimization, self-protection and self-healing. The key-words, which the system supports for its operation are computing everywhere and connectivity everywhere. Features, which the system achieves, are: autonomic maintenance, adaptation in routing decisions and finding data locations, conflict resolution on encrypted data, autonomic replication and archival storage, optimization and self repairing.

In [12] the automated taxonomy generator Sabio is presented. It is a program that can analyze thousands of documents and creates taxonomy for the entire collection. Sabio is being incorporated into a knowledge-management program Raven, developed by Lotus that provide a suit of tools for organizing searching and viewing information in a company’s intranet. During the autonomic document classification Sabio applies properties as self-organization and self management.

It is expected that the principles of the autonomic computing will be applied widely in different engineering applications and technological domains. Potential candidates are the control systems in industrial automation and control process, autonomic manufacturing processes, traffic systems and transportation behavior; information processes for load balancing, spam detection, secure data protection.

POTENTIAL OF THE CONTROL THEORY FOR THE IMPLEMENTATION OF THE AUTONOMIC SELF* FUNCTIONALITIES

The control theory is a theoretical background for the design of self-managing and self-optimization systems. In [4] it has been demonstrated a case study for the application of the control theory to the management of IBM software product. The essential elements of the feedback control system are presented in Figure 1.

The reference input \( v \) gives the desired value of the system’s measured output \( y \). The controller evaluates the control influence \( u \), according to the system error \( \varepsilon = v - y \) and applies this control influence to the target object. The state of the object \( x \) is identified by the observer, which assesses the object behavior by measuring the output \( y \). The target of the overall system is to keep the error \( \varepsilon \) to zero as long as the disturbances and noises affect the controller and target object. The control input \( u \) affects the behavior of the target object and it is changed dynamically during the system operation. Generally, the disturbances and noises are
uncontrollable inputs and their influences to the target object have to be compensated by changes of the control value $u$.

![Feedback Control System](image)

Figure 1 Feedback Control System

For the case of a cluster of HTTP Apache Web servers, the reference value $v$ of the control can be the utilization of the Apache web servers for example less than predefined percentage ($v \leq 66\%$) [4]. Thus, if the farm contains three servers, the fail of one of them will keep the operability of the system unchanged. The control influence $u$ for the target object is the maximum number of connections, defined by the server parameter $Max-Clients$. Changing this value, the CPU utilization of the mainframe can be adjusted. The object state $x$ refers to the response time of the service. It strongly depends on the CPU utilization in the mainframe system. But the CPU utilization is defined by the active users, who perform different remote procedure calls to the HTTP Web servers. For the control system the output $y$ represents the number of active users. The noise, affecting the target object is the arrival rates of requests of the active users, addressing the HTTP services. In [4] it has been estimated an autoregressive model using least squares regression, which has a linear form:

$$y(k+1) = ay(k) + bu(k),$$

where $k$ is the discrete time, $a$ and $b$ are linear regressive coefficients.

This discrete time model is transformed in $Z$ discrete model in the form:

$$W(z) = \frac{a}{z-b}.$$  

Having the transfer function, the steady state output of the system can be easily calculated by putting $z=1$. This means that for steady state, active users calling the HTTP servers have to be $a/(1-b)$ of the $MaxUsers$ parameter.

The dynamical behavior of the control system is easily assessed according to the pole placement. The system dynamics corresponds to the settling time of the HTTP servers. The pole of the transfer function for this linear model is $b$. If $b \geq 1$, the system becomes unstable. That is why keeping the value of $b$ close to zero will decrease the settling time of the system and its response time.

For the case of unstable target object with poles, staying outside the unity circle in $Z$ plane, a controller has to be added. For the case of PI control law, the controller in $Z$ space has a transfer function in the form:
The purpose of the synthesis of the controller is to support system stability and quality of control behavior: accuracy, settling time, overshooting. The transfer function of the closed loop control with a PI regulator is

\[ W(\alpha) = \frac{zK_i}{z - 1}. \]

By setting different values of the \( K_i \), different poles of the close loop system is resulting, Figure 2. For the case of a larger pole very close to 1, this will result in a long settling time. A value of the pole to 0.8 reduces considerably the settling time.

These considerations illustrate the powerful techniques, which the control theory provide for the design of self-managed and self-optimization behavior of autonomic systems. The control theory is a potential candidate for the formalization of problems, related to the management and design of self* autonomic computing system. The control theory applies a set of methodologies, originating from state space optimal control, model predictive and adaptive control, stochastic control and these methodologies can be deployed directly to the autonomic computer systems.

COORDINATION THEORY – A BRIDGE TO SELF-MANAGEMENT SYSTEMS

The need to control large scale systems with nonlinearities and the dynamical nature of the control processes insist development and implementation of new methods. The classical control, which requires a centralized manner of data acquisition, does not produce satisfactory results for complex and distributed systems because of the large scale control problems, which have to be resolved for limited time period. Respectively, the technological control devices have to satisfy the increased requirements in real time. The implementation of such control devices is very difficult, impossible or costly for real applications [9, 10, 13].

One solution for that is the hierarchical, multilevel system theory developing and applying decomposition approaches for the management of dispersed local subsystems, which operate independently. The formal approach for presenting the independent operation of subsystems and its coordination, targeting an overall goal is presented as a set of solutions of hierarchical ordered optimization problems. The multilevel optimization, which is more general than mathematical programming, does not have a lot of real-time applications, unfortunately. One of the reasons is that the advantage of multilevel programming in representing the decision-making processes is hardly to achieve due to the applied complex iterative algorithms [3]. However, the multilevel optimization allows to be ordered and consciously linked the sequence of operation, resulting in self-organizing systems: control, optimization, adaptation self-organization, Figure 3.
This hierarchical order allows to be applied for the formalization of autonomic computer systems. The general system operation is presented as a process of solution of hierarchically interconnected optimization problems, formulated as \[ \min_{x^k} f_k(x^1, x^2, ..., x^k) \]  
\[ g_k(x^1, x^2, ..., x^k) \leq 0 \]  \[ (1) \]
where \( x_1 \) solves \[ \min_{x^1} f_1(x^1, x^2, ..., x^k) \]  
\[ g_1(x^1, x^2, ..., x^k) \leq 0 \]  \[ (1a) \].

Problem (1) is the highest-level one, corresponding to the highest level of the hierarchy. The decision maker (coordinator) at this level controls the decision variables \( x^k \) and its objective is to minimize the function \( f_k \). Similarly, (1b) is the first level problem and it corresponds to the lowest level of the hierarchy. The multilevel optimization problem (1) is known to be extremely hard to solve [1, 2]. Even in the simplest version of a two level optimization it becomes non-convex and/or non-smooth and falls into the class of global optimization [2].

The solution of the optimal design problems with non-smooth structure can be found applying a penalty function method; Karush-Kuhn-Tucker type conditions; a pure non-differentiable optimization technique (bundle optimization algorithm). Following the dynamical behaviour of the autonomic systems, such an approach of solving hierarchically ordered optimization problems looks to be not appropriate.

A potential solution for the coordination strategy is so called “non-iterative” coordination [15]. The local subsystems solve and send to the coordinator their suggestions \( x(0) \), evaluated with lack of coordination \( y = 0 \) (each subsystem is working independently for achieving its local goal). The coordinator, having these “suggestions” modifies \( x(0) \) towards the global optimal solution \( x^{opt} \), which concerns the compromise of mutual operation in the framework of overall system. Then, these corrections are transmitted to the subsystems for implementation.

For two level hierarchical systems the iterative data transfer between the levels is reduced to only two communications, Figure 4. The non-iterative coordination has been developed for the case of goal coordination and model predictive strategies. The coordinator influences the goal functions of the subsystems for goal coordination. For the case of model prediction, the coordinator influences the resources, which each subsystem uses for its operation. The formal models of the non-iterative
coordination founds on approximation of inexplicitly given dual Lagrange problems and corresponding inexplicitly given extreme functions [15]. By solving sequentially optimization problem an autonomic computing system can be managed in a way in accordance with environmental changes, mainly for the workload of the system and respecting the individual independency of each control subsystem.

CONCLUSIONS

The development of autonomic computing systems needs to be applied quantitative formal models, describing their behaviour and dynamical management. The multilevel system theory has its formal potential in modelling and management of independent operating subsystems. Particularly, the coordination strategies achieve optimal control and system management. The non-iterative coordination allows being performed on-line adaptation towards the changes of the system behavior. The paper suggests application of the control system theory and hierarchical system management which have strong potential to respond to the requirements of autonomic system design.

REFERENCES


ABOUT THE AUTHORS

Prof. Todor Stoilov, D.Sc., PhD, E-mail: todor@hsi.iccs.bas.bg
Assoc.Prof. Krasimira Stoilova, D.Sc., PhD, E-mail: k.stoilova@hsi.iccs.bas.bg
Institute of Computer and Communication Systems, Bulgarian Academy of Sciences
This work is partly supported by projects BG 051 PO 001/07/3-02/21 and DVU 01/0031 (INPORT) for the development of on-line investment portfolio models of the Bulgarian Scientific Fund.