



## An empirical study of modeling self-management capabilities in autonomic systems using case-based reasoning

Malik Jahan Khan<sup>a,b,\*</sup>, Mian Muhammad Awais<sup>a</sup>, Shafay Shamail<sup>a</sup>, Irfan Awan<sup>b,c,d</sup>

<sup>a</sup> Department of Computer Science, Lahore University of Management Sciences (LUMS), Lahore, Pakistan

<sup>b</sup> Namal College, Mianwali, Pakistan

<sup>c</sup> School of Computing, Informatics and Media, University of Bradford, Bradford, UK

<sup>d</sup> Computer Science Department, King Saud University, Saudi Arabia

### ARTICLE INFO

#### Article history:

Received 1 July 2011

Received in revised form 12 August 2011

Accepted 13 August 2011

Available online 14 September 2011

#### Keywords:

Autonomic computing

Case-based reasoning

CBR based modeling

Simulating self-management

Simulating problem diagnosis

### ABSTRACT

Autonomic systems promise to inject self-managing capabilities in software systems. The major objectives of autonomic computing are to minimize human intervention and to enable a seamless self-adaptive behavior in the software systems. To achieve self-managing behavior, various methods have been exploited in past. Case-based reasoning (CBR) is a problem solving paradigm of artificial intelligence which exploits past experience, stored in the form of problem–solution pairs. We have applied CBR based modeling approach to achieve autonomicity in software systems. The proposed algorithms have been described and CBR implementation on externalization and internalization architectures of autonomic systems using two case studies RUBiS and Autonomic Forest Fire Application (AFFA) have been shown. The study highlights the effect of 10 different similarity measures, the role of adaptation and the effect of changing nearest neighborhood cardinality for a CBR solution cycle in autonomic managers. The results presented in this paper show that the proposed CBR based autonomic model exhibits 90–98% accuracy in diagnosing the problem and planning the solution.

© 2011 Elsevier B.V. All rights reserved.

### 1. Introduction

With the passage of time, the reliance and dependence on computers has increased tremendously due to automation and computerization of manual processes. This has led to increased process speed and higher accuracies, but has added to the complexity. The software systems have grown in size, functionality and complexity. As a result, it has become difficult to solve an unexpected problem. Solving a problem manually after its diagnosis in large scale systems is not a trivial job. Immediate solution strategies are needed that not just minimize the downtime but also the involvement of the human administrator.

Inspired from the human nervous system, in which most of the activities are performed without explicit permission of human itself, autonomic computing [8,23,28,36,56] is a concept which injects some self-managing capabilities in the large systems. The complexity of the systems increases by adding autonomic capabilities but in return the human intervention for managing such systems can be minimized. The additional layer of complexity thus ensures a lot of problem solving responsibilities to be performed automatically and autonomously.

\* Corresponding author at: Department of Computer Science, Lahore University of Management Sciences (LUMS), Lahore, Pakistan.

E-mail addresses: [jahan@lums.edu.pk](mailto:jahan@lums.edu.pk) (M.J. Khan), [awais@lums.edu.pk](mailto:awais@lums.edu.pk) (M.M. Awais), [sshamail@lums.edu.pk](mailto:sshamail@lums.edu.pk) (S. Shamail), [i.u.awan@bradford.ac.uk](mailto:i.u.awan@bradford.ac.uk) (I. Awan).

Two architectures are available that incorporate autonomic behavior in a computing system: externalization and internalization. In externalization architecture, autonomic manager which is responsible for injecting the self-managing behavior, resides outside the self-managed component. In internalization architecture, autonomic manager is built in within the self-managed component. The properties of autonomic systems are collectively known as self-\* properties. These include: self-configuration, self-optimization, self-protection, self-healing, self-awareness, context-awareness, open, and anticipatory [28,36].

Self-configuration capability enables seamless adaptation of a new component or a new execution environment without significant human intervention. Self-configuring systems also need to configure themselves according to user-defined high level goals. User only specifies what is needed and not how it should be achieved. Autonomic manager prepares a configuration plan to seamlessly adapt changes in environment according to user-defined specifications [6,33,71]. A self-optimizing system is supposed to continuously seek for the possibilities of optimization in the computing environment.

Self-optimization capability optimizes various parameters at run time according to environmental conditions [19,50]. It enables monitoring, experimenting and tuning various parameters to make best use of the existing resources [36].

Systems having the self-protection capability are expected to defend against various malicious attacks and risks causing large-scale system-wide problems or localized problems within the components [16,72].

Self-healing property detects various system failures and recovers from failures to enable minimum downtime of the system [4,52,57,58]. A self-healing system is supposed to remain conscious of the failures or abnormal behaviors of services and should diagnose and plan recovery action [36]. Autonomic systems should be self-aware of their state, contextually aware of environmental changes, open for portability and common standards, and anticipate their needs, context and behavior [28,56].

In the literature, various intelligent techniques like rule-based systems [14,33,34], artificial intelligence planning [6], hot swapping [5], PeerPressure [67], process query systems based on Hidden Markov Model (HMM) and state machines [60], active probing [59], and control theoretic approach [22,25], etc. have been used to enable various self-management capabilities in autonomic systems. Conversational case-based reasoning [29] has been applied for diagnosis of system failures and recovery which exploits the knowledge gained from recent successes and failures but has been applied at a very limited level in context of autonomic systems. Rule-based systems required rich well-formalized domain knowledge to construct the rule-base. Planning-based techniques require complete state space before searching the goal state. Hot swapping has been applied for self-configuration and adaptation of various code snippets dynamically. Hence, its application scope is limited. Control theory is model-based approach which uses mathematical models that are developed based on certain assumptions, thus reducing their applicability. Other probabilistic approaches like PeerPressure, HMM, state machines, active probing, etc. also rely on the availability of rich prior knowledge for learning the models. These techniques get benefit from the past experience to some extent but recent experiences are not considered while devising the solution. So, experience base is not continuously updated and utilized. These approaches limit the learning growth with passage of time and having new experiences. They do not learn from the recent failures or successes.

Case-based reasoning (CBR) [1,9,17,49,70] is a lazy learning paradigm which is an attractive option to be used in the domain of autonomic computing because of the re-occurring nature of the autonomic computing problems in particular and computing systems in general. CBR is a reasoning methodology which uses past experience to find solution of a new problem. It is a good candidate technique to be adapted for autonomic systems due to various reasons. The detailed discussion on CBR cycle follows in Section 4.

CBR works well where well-structured knowledge is not available before hand because it offers a flexible case structure. Its proposed solution is not rigid and can be revised using adaptation algorithms or human intervention. Revised solution is then finally adapted as the solution of the current problem. Current problem–solution pair may be retained in the case-base to add new knowledge for future use. Past experience is maintained in a repository called case-base.

Though some work on CBR has been performed in autonomic computing earlier [38,40], the focus of this research work is to evaluate the impact of CBR based modeling approach in autonomic systems for the externalization and internalization architectures. The details follow in Section 5. Their focus was on improving the effectiveness of CBR cycle by presenting a clustered CBR approach that resulted in the reduced computational cost and case-base size. However the focus of the present work is to apply CBR in both of the autonomic architectures exploiting its different similarity measures and adaptation algorithms.

In this paper, we present CBR-based autonomic computing models and self-management algorithms for externalization and internalization options of self-management. We propose to apply CBR model within autonomic manager as an analysis and planning tool. Our proposed approach promises to change and improve the way autonomic managers analyze the monitored problem and plan the remedial action. We have designed and implemented a tool called *CaseBasedAutonomie* (CBA) which can be tailored to be used for both externalization and internalization options of self-management. We have investigated the efficiency and effectiveness of our proposed approach empirically. The empirical study is based on a multi-dimensional research methodology. These dimensions include similarity measure for retrieval of nearest neighbors, cardinality of set of nearest neighbors, solution algorithm, adaptation algorithm and two different case studies pertaining to externalization and internalization respectively.

We selected Rice University Bidding System (RUBiS) [65] for our experiments based on externalization architecture. RUBiS is used as a benchmark for performance and optimization of distributed systems. It is a Java based three tier application which provides basic functionality of an auction site like eBay. It uses MySQL as the database management system at

متن کامل مقاله

دریافت فوری ←

**ISI**Articles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات