UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF GRADUATE STUDIES AND ACADEMIC ADVANCEMENT

Autonomic Computing Architecture

By Self-defined Uniform Resource Identifier

A Thesis

Submitted to the College of Graduate Studies and Academic Advancements

in Partial Fulfillment of the Requirement for the Degree of Master of Science in Information Technology

Prepared By:

Abdelhafiz Ahmad Khoudour

Supervisor By:

Dr. Nadir Kamal Idriss

January 2017

Abstract

Architecture is a concept provide a mechanism for realizing the self-management which is a good solution for computing resources, and the main objective of the automaticity is to realize the self-management by using the autonomic architecture.

Moreover, important problem arise when publishing information in the web by semantic technologies, represented in clash names, to avoid that we applied the self-defined, which specify a uniform way to retrieve information about the resource being identify by our Uniform Resource Identifier (URI), and to realize the self-management. We focused on applying the Autonomic architecture for self-management coronary heart disease by providing the self-defined URI$_{S}$, which help to obtain a good result used to adding self-management capabilities to a system. To achieve the goal, five patient records with coronary artery disease collected from Statistic unit in Ahmed Gasim Hospital/ Khartoum contain different attributes based on the causes of the disease, applying for that the open source application (Prot é g é) to solve the problem statement, and obtain a good result used to reduces errors, costs, and improve the medical productivity of the hospital with the possibility of editing reports and querying from anywhere.

Finally Semantic web rule language (SWRL) and other recommended future work are, building autonomic application, development of a system with object-oriented methodology, in addition, build a Sudanese medical ontology in coronary heart disease.

المستخلص

هيكلية الحوسبة اللارادية هي عبارة عن مفهوم لتحقيق الإدارة الذاتية ، ويعتبر حل جيد لادارة الموارد الحاسوبية والهدف الرئيسي منه هو جعل النظام قادراً على إدارة مكوناته دون تدخل بشري، وعليه فقد تطرأ مشاكل عديدة اثناء نشر المعلومات على شبكة الإنترنت يتمثل في تصادم او تداخل المعلومات عند استخدام تقنيات الويب الدلالي ، ولتجنب ذلك تم تطبيق معرَف الموارد الموحد لتحديد طريقة موحدة لاسترداد المعلومات التي تم تعريفها مسبقا عن طريق ال(URI) Uniform Resource Identifierاضافة الى تحقيق الادارة الذاتية. عليه فقد ركزنا على تطبيق الإدارة الذاتية باستخدام وتنفيذ معرّف الموارد الموحد (URI) من خلال هيكلية الحوسبة اللارادية ,ذلك للحصول على نتائج جيدة تستخدم في إضافة قدرات للإدارة الذاتية للنظام.

ولتحقيق ذلك فقد تم اخذ ٥ عينات لبيانات مرضى القلب والاوعية الدموية من مكتب الاحصاء بمستشفى احمد قاسم بالخرطوم وذلك بناء على اسباب المرض. ولحل مشكلة البحث استخدامنا البرنامج المفتوح المصدر(Protégé) وذلك للحصول على نتائج تستخدم في تقليل الاخطاء،التكلفه وتحسين الاداء الطبي للمستشفى وذلك باستخراج التقاريرالطبية مع امكانية الاستعلام عن معلومات المرضى من اي مكان .

ومن توصيات العمل المستقبلي لهذا البحث هو استخدام لغة الويب الدلالي او "الويب ذات الدلالات اللفظية" هي عبارة عن لغة تستخدم لتمثيل المعرفة بطريقة تجعلها قابلة للفهم والمعالجة اوتوماتيكيا من قبل الحواسيب وامكانية الاستفادة منها في استخدام هذا التطبيق بالاضافة الى توصيات اخرى.

1.1**Background**

In this chapter, we describe autonomic computing concepts, challenges, self-defined and Problem statement.

In mid-October 2001, IBM released a manifesto observing that the main obstacle to further progress in the information technology industry is a looming software complexity crisis.

The term autonomic computing is emblematic of a vast and somewhat tangled hierarchy of natural self-governing systems, many of which consist of myriad interacting; self-governing components that in turn comprise large numbers of interacting, autonomous, self-governing components at the next level down.

It will be profitable to seek inspiration in the self-governance of social and economic systems as well as purely biological ones.

An autonomic element will typically consist of one or more managed elements couple with a single autonomic manager that controls and represents them. The autonomic manager will relieve human responsibility of directly managing the managed element.

IBM frequently cites four aspects of self-management. Early autonomic systems may treat these aspects as distinct, with different product teams creating solutions that address each one separately. Ultimately, these aspects will be emergent properties of a general architecture, and distinctions will blur into a more general notion of self-maintenance.

System administrators and end users will take the benefits of autonomic computing for granted. Self-managing systems and devices will seem completely natural and unremarkable, as will automated software and middleware upgrades.

Autonomic computing is a grand Challenge that reaches far beyond a single organization; its realization will take a concerted, long-term, worldwide effort by researchers in a diversity of fields [1].

The ideas behind autonomic computing are not new. It is possible to find some aspects of autonomic computing already in today’s software products. For instance, Windows XP optimizes its user interface (UI) by creating a list of most often-used programs in the start menu. Thus, it is self-configuring in that it adapts the UI to the behavior of the user. It can also download and install new critical updates without user intervention, sometimes without restarting the system. Therefore, it also exhibits basic self-healing properties. Dynamic host communication protocol (DHCP) and domain name service (DNS) services allow devices to self-configure to access a TCP/IP network.

However, our definition of what we mean by autonomic computing is that of a self-adaptive system as opposed to an adaptive system. Therefore, standard query optimizers would not be consider as providing automaticity.

However if a query was running and the DBMS was monitoring the query's execution and deciding on a different query plan, then we would consider that autonomic [6].

An architecture for autonomic computing must accomplish three fundamental goals:

- **First, it must describe the external interfaces and behaviors required of Individual system components.**
- **Second, it must describe how to compose these components so that the components can cooperate toward the goals of system-wide self-management**
- **Finally, it must describe how to compose systems from these components in such a way that the system as a whole is self-managing [5].**

1.2 Problem Statement

After having an overview on autonomic computing area, RDF, OWL and SPARQL concepts, many of problems facing the realization of self-management such as complexity, heterogeneity, security, lack of guarantee and open source application.

A. **How to realize the self-management by using the architecture.**

By combining architecture components, which consist of autonomic manager, managed element and semantic technology tools (RDF, OWL and SPARQL) to working together.

The autonomic manager used its internal components (monitor, analyze, plan and execute) capabilities to make the system self-management.

The Resource Description Framework (RDF) is used to build a framework from the triple (subject, predicate and object) to represent information as a graph in the web defined by Ahmed Gasim hospital uniform resource identifier (URI).

Ontology web language (OWL) is to define ontologies that include classes, properties and their relationship to share these common understanding of the structure of information among people or software agents.

SPARQL query language is a query engine to return defined resources by the URI (triples) from a collection of published data in the web.

B. **How to realize the self-management**

We applied the self-defined using for that Ahmad Gasim hospital URI$_S$ features to define resources.

C. **How to improve the healthcare system**

Self-defined improve the productivity of the system by defining resources using the hospital URI$_S$ which help to realize the self-management and avoid any clash name when information distributed in the web.

1.3 Objectives

The main objectives in this research are:

- **To realize self-management system by providing self-defined using RDF, OWL, and SPARQL.**
- **Expect result and performance of the autonomic architecture.**

1.4 Thesis Structure

The remainder of this thesis structured in five chapters.

Chapter 1: Represent an introduction about the research, problems, and objectives.

Chapter 2: literature Review

Chapter 3: Methodology explain RDF concept and its implementation, OWL and SPARQL, research architecture, case study used to apply the autonomic and to solve the problem.

Chapter 4: Implementation

Chapter 5: Result and discussion, finding and analyze the performance, conclusion and future work. As illustrated in the following figure:

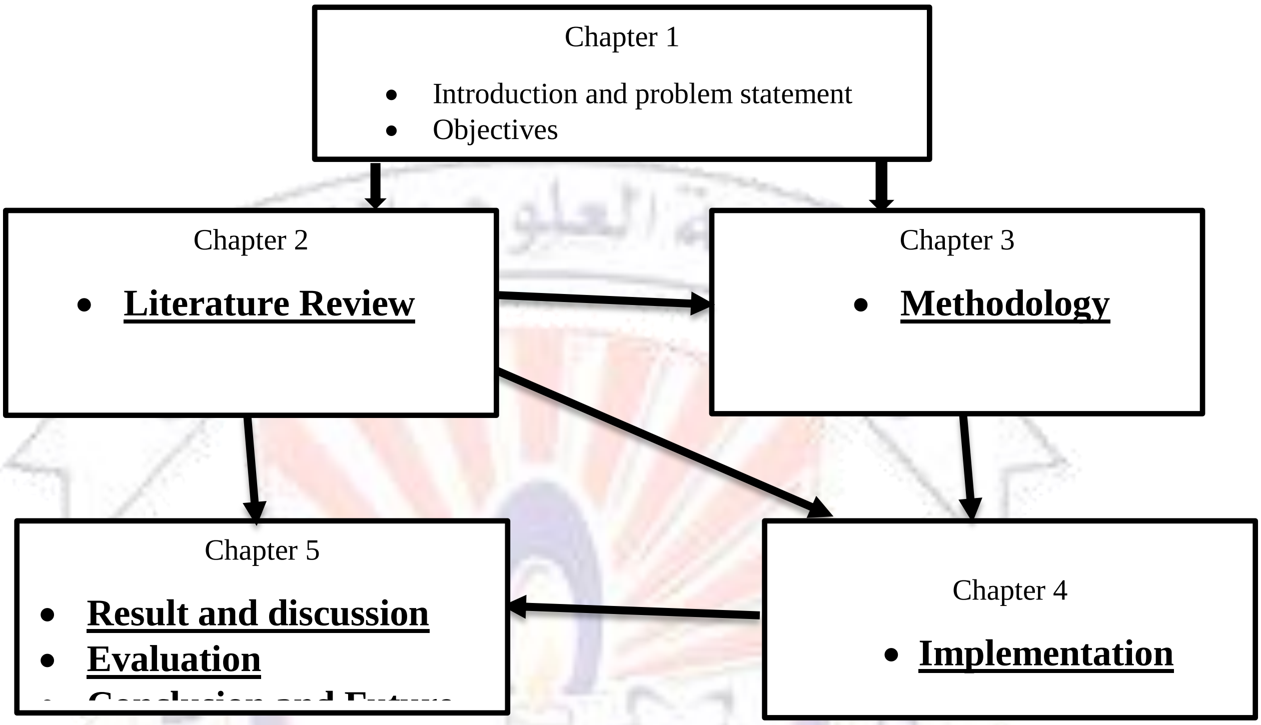

Figure (1.1): Thesis structure