A Formal Approach for Matching and Ranking Multi-featured Trustworthy Context Dependent Services

By
Afnan Bobaker Saeed Bawazir

A thesis submitted for the requirements of the degree of Master of Computer Science

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY
JEDDAH – SAUDI ARABIA

Shaban 1437H- May 2016G

بسم الله الرحمن الرحيم

قال تعالى﴿ قَالُوا سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا إِنَّكَ أَنتَ الْعَلِيمُ الْحَكِيمُ ﴾
سورة البقرة آية ٣٢

A Formal Approach for Matching and Ranking Multi-featured Trustworthy Context Dependent Services

By
Afnan Bobaker Saeed Bawazir

A thesis submitted for the requirements of the degree of Master of Computer Science

Supervised By
Dr. Wadee Saleh Alhalabi

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY
JEDDAH – SAUDI ARABIA
Shaban 1437H- May 2016G

المنهج الدقيق للمطابقة والتصنيف للخدمات متعددة الخصائص الموثوقة والمعتمدة على السياق

أفنان بوبكر سعيد باوزير

بحث مقدم لنيل درجة الماجستير في علوم الحاسبات

إشراف:
د. وديع صالح الحلبي

كلية الحاسبات وتقنية المعلومات
جامعة الملك عبدالعزيز
جدة – المملكة العربة السعودية
مايو ٢٠١٦ م – شعبان١٤٣٧ هـ

A Formal Approach for Matching and Ranking Multi-featured Trustworthy Context Dependent Services

By
Afnan Bobaker Saeed Bawazir

This thesis has been approved and accepted in partial fulfillment of the requirements for the degree of Master of Computer Science

EXAMINATION COMMITTEE

	Name	Rank	Field	Signature
Internal Examiner	Dr.Anas Fattouh	Associate Professor	Computer Science	
External Examiner	Dr.Abdullah Alghamdi	Associate Professor	Information Systems	
Advisor	Dr.Wadee Alhalabi	Associate Professor	Computer Science	

KING ABDULAZIZ UNIVERSITY
Shaban 1437H- May2016G

المنهج الدقيق للمطابقة والتصنيف للخدمات متعددة الخصائص الموثوقة والمعتمدة على السياق

إعداد
أفنان بوبكر سعيد باوزير

تمت الموافقة على قبول هذه الرسالة استكمالا لمتطلبات درجة الماجستير في علوم الحاسبات

لجنة المناقشة والحكم على الرسالة

	الاسم	المرتبة العلمية	التخصص	التوقيع
عضو داخلي	د.انس فتوح	أستاذ مشارك	علوم حاسبات	
عضو خارجي	د.عبدالله الغامدي	أستاذ مشارك	نظم معلومات	
مشرف رئيس	د. وديع الحلبي	أستاذ مشارك	علوم حاسبات	

جامعة الملك عبدالعزيز
مايو ٢٠١٦ م – شعبان ١٤٣٧ هـ

Dedicated to

My beloved mother and my dear father

ACKNOWLEDGEMENTS

In the Name of Allah, Most Gracious, Most Merciful. All praise be to Allah, the Lord of the Worlds, and prayers and peace be upon Mohamed, His servant, and messenger.

First and foremost, I would like to acknowledge and thank the Almighty Allah that enabled me to begin in this research and complete it. This work would have never been completed without His guidance.

I would like to express my very great appreciation to my supervisor Dr.Wadee Alhalabi, for his patience, support, effort and professional guidance. It was an honor for me to work with him. Also, I am particularly grateful for the guidance was given by Dr. Mubarak Mohammed. My special thanks are also extended to Dr. Akila Sarirete for providing me with very valuable tips during the writing stage. I truly learned a lot from them.

Deeply from my heart with love, I would like to thank my family (my father, mother, sisters, and brothers) for the generous support they have provided me since I was a child and particularly throughout my Master's degree. Because of their unconditional love, encouragement and prayers, I have had the chance to complete this thesis. May Allah bless them all forever.

This thesis was part of a project which was funded by the National Plan for Science, Technology and Innovation (MAARIFAH) – King Abdulaziz City for Science and Technology - the Kingdom of Saudi Arabia – award number (12-INF3046-03). I also, acknowledge with thanks Science and Technology Unit, King Abdulaziz University for technical support.

A Formal Approach for Matching and Ranking Multi-featured Trustworthy Context Dependent Services

Afnan Bobaker Saeed Bawazir

Abstract

Service-oriented computing (SOC) is shaping the future of distributed computing and enterprise application development. Due to the increasing volume of services, it is challenging for requesters to select the trustworthy service that suits the requester and provider context. Consequently, there is a need for a ranking process that takes into consideration the rich features of services, the context of the service provider and requester to enhance the relevance of the top of the ranking result. Also, it should include trustworthiness requirements to provide reliable services based on requesters' preferences. This thesis investigates the ranking process in service-oriented architecture, context awareness, trustworthiness, and recommender system. Besides, this thesis formalizes the criteria to model and rank context features, non-context features, and composite trustworthiness features. This thesis proposes a formal architecture for matching and ranking trustworthy context-dependent services. We present a real-world case study on King Abdullah Scholarship Program (KASP) and show how to rank trustworthy context universities and to evaluate the proposed architecture.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND TERMINOLOGY

SOC	Service Oriented Computing
SOA	Service Oriented Architecture
KASP	King Abdullah Scholarship Program
FrSeC	Formal Framwork for Providing Context-dependent Services
MB	More is better
LB	Low is better
EB	Exact is better
MAUT	Multi-Attribute utility Theory
SRC	Service Requester Context
SPC	Service Provider Context
PM	Prepared Matrix
UR	Unsorted Rank
MOE	Ministry of Education.
NSS	National Student Survey

Chapter 1

Introduction

Chapter I

Introduction

1.1 Introduction

The development of enterprise applications and distributed computing rely on Service-oriented computing (SOC) [1]. Service-oriented architecture (SOA) is an architecture model of SOC. Service is the primary element of this architecture. Many service providers advertised their services in the service registry. Service requester can browse the registry to select services. Due to the vast volume of services and the heterogeneity of their features, the selection of the right services is challenging and time-consuming. Usually, the service selection is based on functional and non-functional requirements. The problem of finding the services that match the query and ranking them based on the users' preferences is discussed in [2]. Bringing the context into the ranking system would enhance the relevance of the ranking list. For instance in on-line marketing, when ranking a product, the on-line marketing should take into account whether the product will ship to requester's country or not. To improve the relevance of the ranking list, the context of the service provider, requester and execution time should be taken into consideration. Through context awareness, each requester has its own ranking list.

To solve the redundant services that have similar functionality features ranked at the top of the list, trustworthiness plays a key role to rank and select the more trustworthy services based on users' preferences. This work focuses on designing a formal method for ranking context information and trustworthiness features.

1.2 Motivation

Suppose that there are postgraduate students searching for universities to apply to. Around the world there are many available universities and each university claims that it is the best. This is overwhelming to students as there is just too many offers to read and compare. In addition, the student needs to make sure that the existing qualification will be recognized by the foreign university that they will apply to. At the same time, they have to search for a legitimate university that is accredited by their countries. At the end, they simply fail to know who to trust and what to select. Different students from different countries have different context information. Therefore, there is a need to develop an efficient tool matching and ranking universities based on student criteria and taking into account the context information and the trustworthiness properties.

This problem can be encountered in many other domains such as jobs searching: we need to match the excellent job seekers with the best job vacancies. In the medical domain, the wrong choice of medication or dose would cause serious problems. The list can be extended to searching for flights, real estate, and hotels. That motivated us to design a formal approach for matching and ranking trustworthy context depended services.

Designing creative tool like this would be beneficial to both service provider and requester. Providers will be able to advertise their services to the wide markets and increase their clients' base. Requesters will be able to filter through the massive number of choices available. That is, requesters do need to invest more time and energy to select the right service as they did with the traditional method.

1.3 Problem Statement

One of the most important aspect that was addressed is represented by "how to formally model the context". This question lead to defining a formal criteria to distinguish context features from non-context features. During the design phase, these criteria help the developer to select the context features. We try to solve how to rank context features and combine them with non-context features to improve the relevance of the top ranking list.

Trustworthiness is a composite feature. For example, security involves, among many other features, authentication, authorization, and integrity; safety involves timeliness and privacy. In this work, we investigate how to model these composite features and how requester states their preferences. This composite feature is rarely discussed in the literature and addressing it is an important aspect of the current work. We also tried to solve how to match and rank services based on trustworthiness features and to take into consideration the context.

Currently, there is no published work that discusses the above challenges of matching and ranking rich services. Therefore, the motivation of this work is to provide a solution based on the formal foundation for the matching and ranking trustworthy context depended services.

1.4 Thesis Objectives

The proposed research aims to achieve the following objectives:

1. Matching and ranking context features.
2. Matching and ranking trustworthiness features.
3. Provide a formal specification for matching and ranking multi-featured trustworthy context-dependent services.
4. Propose an architecture for matching and ranking multi-featured trustworthy context-dependent services.
5. Provide a case study on real world application that illustrates the success of the proposed architecture.

1.5 Research Methodology

To achieve the thesis objectives, the following steps are performed:

1) Study and review of the existing ranking algorithms and choose the most suitable algorithm that can be followed in this research.
2) Review issues related to context awareness and trustworthiness concepts.
3) Define criteria for context, non-context and trustworthiness features.
4) Propose a formal model for context features.
5) Propose a formal approach for matching and ranking services based on context features.

6) Propose a formal model for trustworthiness features.
7) Propose a formal approach for matching and ranking services based on trustworthiness features.
8) Implement the proposed formal architecture on real world applications.
9) Collecting the data set, define the trustworthiness and the context information.
10) Provide a case study on king Abdullah scholarship program to illustrate the success of the proposed methodology.

1.6 Thesis Organization

The thesis is organized into five chapters. Chapter 2 provides a literature review of service oriented architecture, context awareness, and trustworthiness concepts. It focuses on recommender systems and reviews the prior works. Chapter 3 discusses the proposed architecture and methodology and provides the formal specification. In chapter 4, we present the case study used to test the proposed architecture, followed by a detailed discussion and analysis of results. Chapter 5 concludes the work, with a special emphasis on results and limitations. In addition, some directions for future work are suggested.

Chapter 2

Literature Review

Chapter II

Literature Review

2.1 Introduction

This section provides a literature survey on service oriented architecture (SOA), context, and context-awareness, trustworthiness and recommender systems. It discusses the prior works on trustworthy and context-aware service ranking.

2.2 Service-oriented Architecture

In traditional SOA interactions, the three main interacting elements are the service provider, the service requester, and the service registry. The service provider defines a service and publishes it through the service registry. The service registry acts as a data center that holds the services published by the different service providers. The service requester accesses the registry to get information about available services. This information is used to select a specific service that meets its requirements and that interacts with the service provider of the selected service. Figure 2.1 illustrates the SOA.

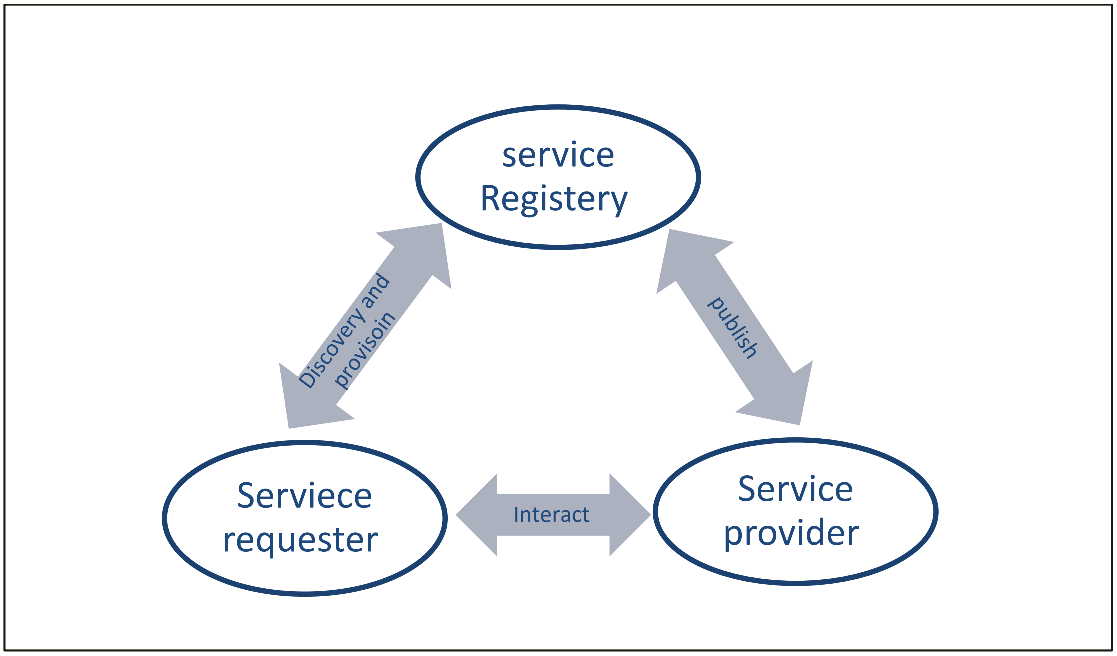

Figure 2.1 Fundamental blocks of SOA.

Hence, the main activities in SOA are service publication, service discovery, and service provision. In literature, these terms are used in the following sense:

- *Service publication* refers to defining the service contract by service providers and publishing them through available service registries.
- *Service discovery* refers to the process of finding services that have been previously published and that meet the requirements of a service requester [3]. Typically, service discovery includes service query, service matching, and service ranking. Service requesters define their requirements as service queries.

As Alsaig stated in [2], matching filters out all other services that do not exactly match a predefined value in service request, while ranking considers all available options and orders them according to their approximate closeness to a given query in respect to the defined weights of importance for each feature such that a higher rank indicates a very interesting option and a lower rank score indicates a less important option. In real life situations, there is no service that fully satisfies all the features stated in the user query. That means that matching is insufficient and is not effective since sometimes it returns no results at all. On the other hand, ranking is a significant way that allows partial satisfaction of user's preferences. Therefore, the system can prevent the case when no result is returned. Thereby, it decreases the user effort since the closer options to the defined query are presented. In other words, ranking enables the service requester to select a specific or a most relevant service from the top of the list of candidate services.

The basic building unit for SOA-based applications is service. Many features can describe each service. Typical features are: functionality, non-functional properties, and context features. In general, these features describe the qualitative and quantitative characteristics of the service. The authors in [4] suggested a formal model called ConfiguredService to define service including trustworthiness and considering contextual information. They also suggested a new service provision framework to support the provision of trustworthy context-dependent services called FrSeC. This work focuses on service ranking. Service matching and ranking are widely discussed in the literature of SOA as stated by Lu and Bellur and Vadodaria in [5] and [6] respectively. However, they are very limited to the functional aspects of services.

The first contribution in SOC for ranking services on multiple features is the vector-based ranking algorithm [4]. The planning unit in the FrSeC framework [7] executes this algorithm. However, it ranks only the numerical features of services, neglecting other data types of these features. Alsaig in [2] solved all the limitation of the previous work. He suggested a semantic-based, multi-featured ranking algorithm called X-Algorithm. In his work, the author took into consideration the semantic of each feature in order to provide better values, (low is better LB, more is better MB, or exact is better EB). For example, safety and reliability features should have higher values and thus, the semantic is ‘more is better’ (MB), while for cost and shipping time, low values are better. Thus, the semantic is ‘low is better’ (LB) For discrete features, such as brand and language (described textually or logically) exact values are preferred and thus, the semantic is ‘exact is better’ (EB) In addition, the X-algorithm is user-centric, providing results based on users' requirements and preferences. Also, this algorithm is executed by the planning unit in the FrSeC framework [8]. The architecture of X-algorithm is illustrated in figure 2.2. It contains the following elements:

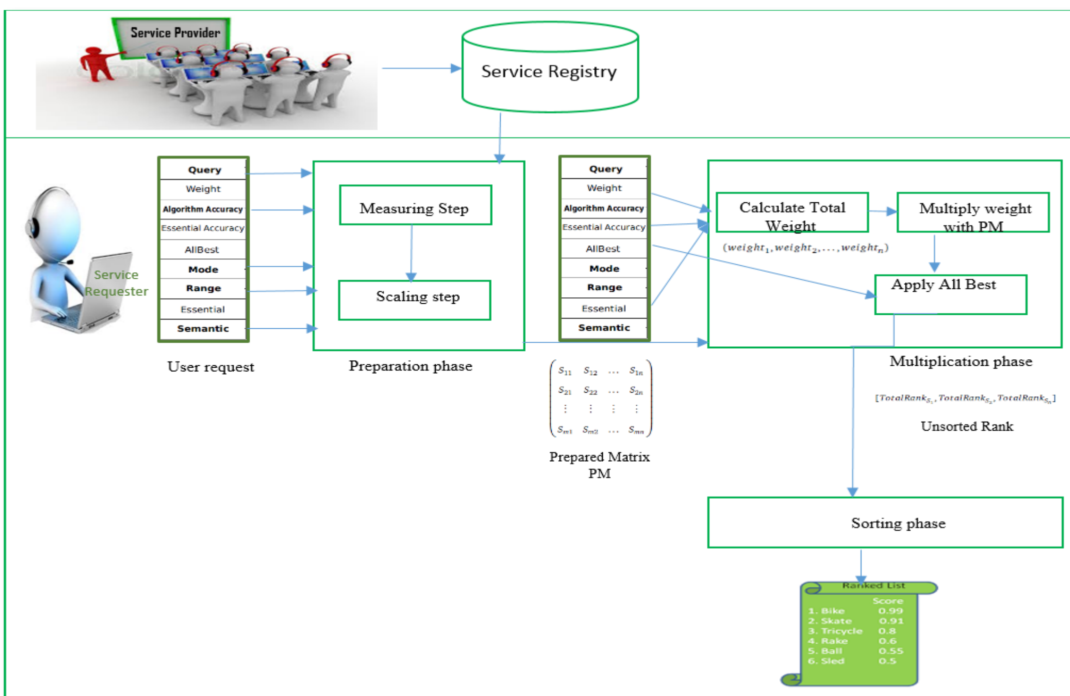

Figure 2.2The architecture of x-algorithm.

1) User request: to submit a query, the user explicitly defines the features' values, their weights to determine the level of importance, the preferred semantic, and mode (Exact mode corresponds to look for an exact match to a specific feature in

the query; Best mode corresponds to look for better values than a specific feature in the query; The range mode for numerical features if the user wants to search with specific range). Users have a choice to determine the essential features. For the entire query, all best options can be set to allow the algorithm to rank the services that satisfying all requirements for all features.

2) Preparation phase: it takes the set of available services in service registry and user request as an input. Then, produces the Prepared Matrix (PM) that contains normalized similarity scores between services and the query.
3) Multiplication phase: it takes the PM and multiplies it with weight vector to produce an unsorted rank (UR) vector.
4) Sorting phase: it takes the UR vector and apply quick sort algorithm with a descending order to produce a sorted rank list.

2.3 Context and Context awareness

According to the Oxford English Dictionary, the term “context” means the circumstances that form the setting for an event, statement, or idea. For example, the context of a meeting may comprise the date, time, participants, location, and agenda. It is important to mention that there are many works that tried to formalize the context definition. Abowd and Dey [9] stated that there are many different types of context such as location, identities of nearby people, objects and changes to these objects, and time. The more formal definition of context is given by Abowd and Dey [9]:

"*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between* *a user and an application, including the user and applications themselves*". This definition shows that the scope of the context is very broad and it is a very rich concept. It differs from one situation to another depending on the purpose of the application. Context awareness is the ability to take advantages of context information and to provide adaptable services. We follow the notion "there is more to context than location" [10]. This indicates that context means much more than location. In general, as defined in [9] "*A system is Context-aware if it uses contextual information to provide relevant information and services to the user, where relevance depends on the user's task*".

A comprehensive survey for defining and modeling context is provided in [11]. However, in [12] Wan gave a formal representation of context. Context is formally modeled as a typed relation that has a representation (structure) and semantics based on the knowledge enveloped by the context. The formal definition involves a set of dimensions and their associated types.

In the definition of "configured service"[7], context is divided into context information and context rule. Context information is molded as defined in [12]. Context information is formalized as a set of dimensions and tags. The set of dimensions "who, when, where, what, and why" are introduced to construct any general context. The context rule is the service provider rule that has to be satisfied to get the service.

In [13], the authors studied the effect of incorporating context-awareness with service discovery and ranking. They categorized the effects into three categories: request optimization, result optimization, and personalization. Request optimization is the process of enhancing the service request. The service request is the main input of discovery process. Therefore, the quality of service request is important. Context-awareness can be applied to optimize the request, it may include as an example request rewrite [14] and request expansion [15]. Result optimization is the process solving the problem of some web services that are ranked lower in the retrieved set but they are more relevant to the user's needs. This problem occurs when the service request is inaccurate. Therefore, result re-ranking with contextual information is one of the most significant when solving this problem. The notion of personalization means providing different result set with the same service request. In other words, no two users have the same ranking list because users are different and they have different needs. Context information is an ideal mechanism to solve the problem of understanding the user's needs.

Work in [16] incorporates contexts with service discovery for mobile environments. It defines context as a special kind of service attribute that is part of the service description. The context attribute value dynamically changes over the time (for example, user location and network bandwidth). On the other hand, static service attributes' values do not change over time. The authors proposed to rank services based on static attributes then use context attribute for filtering out the services. The context is requester-related and it did not take into consideration the provider context.

There have been many studies on service discovery in a pervasive environment in [17]. It concludes that context attributes act as filters and the major limitation is that they fail to rank the suitable services. To clarify, assume we have two services that matched the user context as an example that considered services nearby the user. The work fail to determine which one is more suitable and nearer to the user.

Authors in [18] and [19] provided a quality relevance result by modeling context as a historical search data. The drawbacks of these methods are described as follows: 1) they rely on past context; not considering the current context; 2) the problem of shared devices or account is raised; thereby it is hard to personalize the result; 3) the problem for new user *cold start* problem is raised.

Patent work in [20] invented a method for re-ranking documents in search engine based on contextual signals. Contextual signals include location, time, date, and historical data for the user, demographic information, and social based information. Each of contextual signal was evaluated using machine learning algorithm to make sure the new positions of each document is more relevant to the user. However, [21] discussed that it is hard to employ methods for ranking document and web pages to rank services because of the lack of service descriptions. Thereby, they proposed a method to rank the services based on the service usefulness and the content similarity. Context is defined as how much the services are included in applications. Service usefulness means how this service is useful for certain application; for example, the popularity of the services. Service usefulness is computed based on context information. Context is modeled as weighted bipartite graphs that represent the relationship between services and the involved applications. The algorithm is generic and represents one of the first work in SOA for improving service ranking based on context information. However, the major limitation it is based on keyword matching. It cannot capture all the features of services.

We conclude that context is a rich concept because all researchers have such different views on how to define the concept. To our knowledge, there is no published work on formalizing and ranking context features in SOA and considering the context of the service provider, requester and execution time.

2.4 Trustworthiness

Trustworthiness is the system property that denotes that the system will behave as expected. Trust is a social aspect that is hard to define formally. In the literature, the terms dependability and trustworthiness are used interchangeably [22]. The original definition of dependability is the ability to deliver service that can justifiably be trusted [23].

There is a common agreement in the literature that trustworthiness comprises non-functional requirements such as safety, security, reliability, availability, and timeliness [24]. In the state-of-the-art, trustworthiness is a composite feature [23] including, safety, security, reliability, and availability that are not quantified easily. For example, security involves, among many other attributes, authentication, authorization, and integrity; safety involves timeliness and liveness; reliability involves failure and repair models.

In the definition of “configured service” [7], trustworthiness is divided into service trust and provider trust. Service trust includes services quality such as safety, security, reliability and availability, while provider trust includes recommendations of peers, or consumers’ ratings and reviews. In real life, we consider a service to be trustworthy if it is has a high trust value, high reputation or a combination of these attributes.

Trustworthiness can play a major role in the decision making when choosing a service over the other promoted ones. For example, if two services provide exactly the same functional features but differ in the degree of safety, then arguably, the safer service is chosen.

Works in [25],[26] and [27] shed the light on the importance of incorporating rating and reviews into ranking model to help the user select a high reputation service based on their preferences.

Approaches such as [[28],[29], [30], and [31]] use non-functional properties to enhance service discovery and ranking.

In a recent work, the authors in [30] proposed a method to select services based on user's non-functional requirement. The proposed method studied the relationship between non-functional features using fuzzy logic.

The work in [31] simplified the process of electing non-functional requirements by building a user profile for non-functional preferences; thereby the users are not normally required to specify them for every single query. The services are ranked based on functional requirements and then re-ranked based on non-functional requirements. However, all the previous works neglected the composite trustworthiness feature to rank services.

To our knowledge, there are no prior formal work rank services based on composite trustworthiness features. And there is no work done yet that takes into consideration at the same time the trustworthiness and reputation to rank services.

2.5 Recommender systems

When we talk about ranking services, recommendation systems are widely taken into consideration because can be treated as creating a ranking list. A recommender system is an assistance tool to solve the problem of information overload; thereby, it helps to reduce the number of choices of products or services. Recommender systems are categorized based on the user preferences [32] into 1) rating-based systems; 2) feature-based systems; and 3) personality-based system. In rating-based systems, users explicitly state their preferences by giving a rating to items that they have already experienced. These initial ratings are used to anticipate what the user's desire and provide a recommendation for them in the future. The work in [32] categorized the rating recommender systems in two categories: content-based and collaborative filtering methods. Content-based recommender recommends new items similar to those the user has preferred in the past, while collaborative filtering recommenders work based on this assumption "the users that showed similar tastes (like-dislike) in the past tends to agree again in the future". As in Amazon and Netflix, they usually say, "people who bought this also bought this". The recommendation seems it is specific to the user but the data is collected from many other users. However, in some domains, it is not useful to convince users by saying "people who bought this also bought this", users need to pay attention to every detail. This problem is solved by introducing feature-based systems.

Feature-based systems permit users to explicitly state their preferences on specific item features. Thereby, it helps for matching between user’s need and the set of the available options. This type of recommendation is useful when users are searching for services or products that carry a lot of financial risks (for example cars, real estate agents, finding job and scholarship). Four types of recommender systems are categorized under feature based-system: case-based system, utility based-system, knowledge based-system and critique based-system. Multi-Attributes utility theory (MAUT) [33] is mostly employed in these systems to measure the item's utility. MAUT is a theory taking into consideration the conflicting value preferences and producing a score for each item to represent its overall satisfaction degree with the user preferences. This approach uses the weighted additive utility function as follows:

$$U(< x_1, \ldots, x_n >) = \sum_{i=1}^{n} w_i \; V_i \,(x_i)$$

Where n is the number of features that the items may have, the weight w_i(1 <i <n) is the importance of the feature i, and V_i is a value function of the feature x_i which can be given according to the application domain during the design phase.

The personality-based systems aim to provide more personalized services by understanding the users better from a psychological perspective. Nunes [34] suggested building a user profile based on personality questionnaires to explicitly extract the information about the user. Then the system is able to provide personalized recommendations. The user profile information was used for filtering items with matching characteristics. However, it is hard to formalize and extract the personality. In this work, we focus on a personalized recommendation based on context.

In the literature [35], [36] and [37], most current recommender systems that incorporate with context are the rating-based systems. There is no work-incorporating context and trustworthiness with the features-based system.

From the viewpoint of recommender systems, X-algorithm [2] falls under the features-based system category. [2] It shows a promising result for modeling preferences with semantic and produce a fair recommendation list.

The major limitation of X-algorithm [2] is that it calls "one fits all" algorithm. That is, different users have the same results if the input requested ("query ") is the same. These different users have different intentions and needs. Awareness is an ideal mechanism to solve the problem of understanding the user's intention context.

From the user’s perspective, X- Algorithm has some constraints for the user request. It requires users to enter the query values, weight "to determine the level of importance" and the mood "to determine the preferences“. Since the algorithm is user-centric, tuning the parameters manually is time-consuming and needs effort. The user needs to understand the impact of each weight. Therefore, we suggest in this work to use contextual information to predict these values with less user interaction. In addition, X-Algorithm does not take into consideration trustworthiness features to provide reliable ranking results.

2.6 Conclusion

In this chapter, we reviewed the literature of SOA, context awareness, trustworthiness, and recommender system. We found the best algorithm for ranking multi features services is x-algorithm [2]. We demonstrated the drawbacks of this algorithm.

In the next chapter, we will explain how we extend x-algorithm to rank context features and trustworthiness features in more details.

Chapter 3

Proposed Methodology

Chapter III

Proposed Methodology

3.1 Introduction

This chapter first describes an overview of the proposed framework that aims to improve the relevance of trustworthy service ranking results by taking into consideration the context of the service requester and provider. We introduce the criteria to discriminate the context features, non-context features, and trustworthiness features during the design phase. Second, we proposed an approach related to how to match and rank context information. Thirdly, this chapter includes how to model trustworthiness features and incorporate them into context ranking result. Fourth, the detailed description of each phase in the proposed framework to obtain the final ranking result is presented. Finally, formal specification of the proposed framework is presented to assure the generic of the proposed framework and can be applied in many diverse application domains.

3.2 Fundamental blocks of the proposed framework

Figure 3.1 presents the fundamental components of the proposed framework: service model, user model, ranking unit, context aware unit, and explanation unit. These blocks are described below.

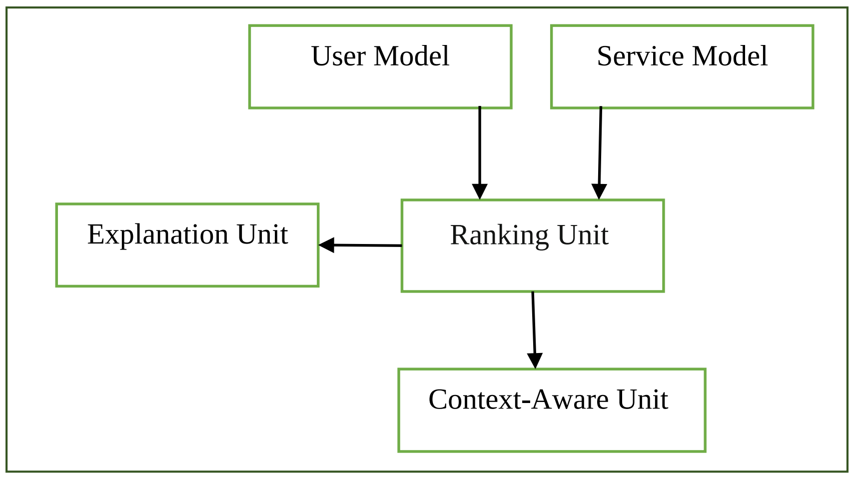

Figure 3.1 Fundamental blocks of the proposed framework.

3.2.1 Service model

A service provider prepares the services and stores them in service registry; each service can be described by many features, which provide sufficient information that is unique to a service. Typical features of a service are functionality, price, and other properties that describe the qualitative and quantitative characteristics of the service. The set of features that are relevant for the application are to be determined by the domain. These features are classified into two groups: ranked features and non-ranked features. Ranked features are a group of features that are included as criteria in ranking model. Non-ranked features that store additional information and are not included as criteria in ranking model such as email, phone number and links.

The ranked features are categorized into three parts: context features, non-context features and trustworthiness features. Following we will describe in more details each category.

3.2.1.1 Context features

Context is any type of information used to characterize an object or situation [38], which helps to obtain the desired and most relevant services according to the service requester.

We model context features as a combination of service requester context (SRC), service provider context (SPC) and context rules.

3.2.1.1.1 Service Requester Context features

The feature that satisfies some of the following criteria is called a SRC feature:

1) Tangible: attributes that can be sensed implicitly from another context and quantified as in the GPS system, which senses implicitly the location of a user on the map.

2) Dynamic attribute: changes over the time; an example of a dynamic attribute for a printer service is the number of prints jobs in the print queue.

3) Computed attributes: attributes that can be computed by using two or more parameters. An example of a computed attribute is to calculate the distance between two GPS locations and rank the nearest one.

4) Inferred from data: ("user profile" or environment). For example, the system can infer the users' age from their birth of dates.

5) Skills required by the service provider (can be a negotiable skill). For example, some universities provide conditional admission for students who have an insufficient score on the IELTS or TOEFL.

6) Not-privacy attribute: because privacy is considered as a fundamental right, it is difficult to be inferred. For instance in the car market system, sometimes users do not want to disclose their personal information such as income or the number of children to allow the system to recommend for them the right car. Trading off between privacy and quality of the ranking result is a huge topic and outside the scope of this thesis.

7) In our research, we have been using the five dimensions WHERE, WHEN. WHAT, WHO, and WHY to construct any general context.

3.2.1.1.2 Service Provider Context features

The SPC features store information that is related only to service provider that characterize the context of service provider and the context of execution time. In general, SPC features have more than one possible value. The context of service requester is used to select one value. Therefore, for each possible value, a context constraint is defined such that only one context constraint can be true at an instant. Consequently, only one value will be selected.

A context constraint is a special type of constraint that is used to decide whether a specific value for service provider should be selected. The decision is based on evaluating a logical expression defined over the values of the context of service request associated with the query. The value is given only if the constraint evaluates to true.

3.2.1.2 Trustworthiness features

Trustworthiness features store information related to service's quality such as safety, availability, security, reliability and timeliness in some of the quantitative and qualitative terms or related to service provider's quality. This information is earned with other party or sources not assumed and therefore, it is considered as a feedback. We have two types of feedback: domain expert feedback and random public feedback. Domain expert feedback is related to verified features that comes from another trusted independent organizations or has to be accompanied with a proof (for instance: ISO Certification). Random public feedback is related to claimed features that come from customers such as customers' satisfaction, reviews or claimed by the service provider.

3.2.1.3 Non- context features

The rest of the features that are not categorized as service provider context feature, service request context feature or trustworthiness features are considered as a non-context feature. Figure 3.2 illustrates the classification of service's features.

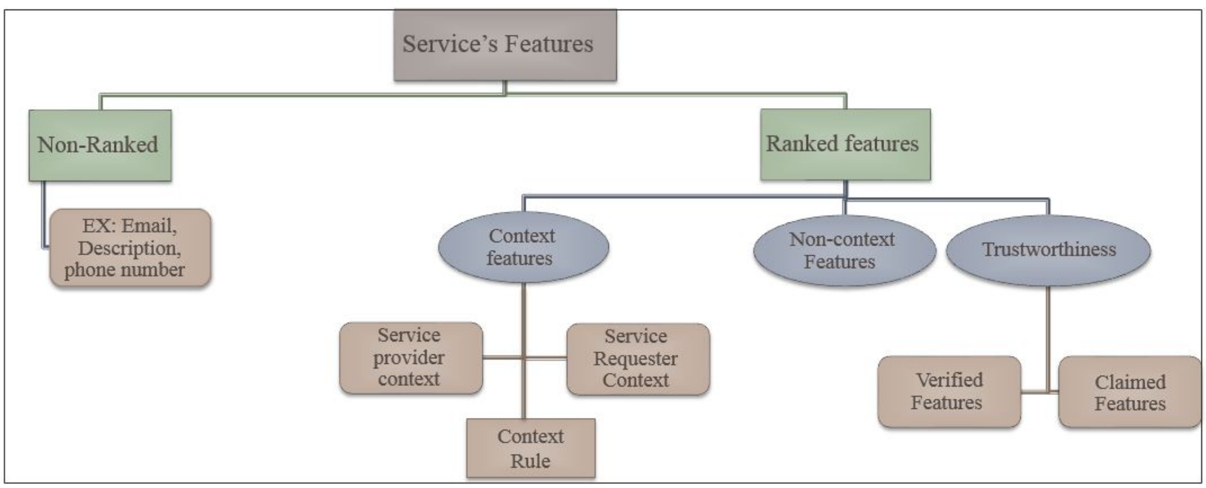

Figure 3.2 Classification of service's features.

3.2.2 User Model

The author in [39] explained the need for user models because users are different: they have different background, different knowledge about a subject, different preferences, goals and interests. A user model should be applied to allow the selection of individualized responses to the user.

In [40], the authors called the task of collecting user's preference information to define the query by user modeling. User modeling can be done in two ways: explicit and implicit. Explicit data is collected by explicitly asking users to determine preferences for each feature. Implicit data refers to applying different mechanisms to collect data by monitoring observable user's behaviors, interaction history or by inferring it from another data

In order to build a context-dependent application, user profile and identity are the main components to model the user. Usually, the user profile, as Koch states in [39], represents cognitive skills, intellectual abilities, and intentions, learning styles, preferences and interactions with the system. In general, the user profile in our work is composed of three sections:

1. Personal information about the user, which is final and not subject to change such as name, birth of date, nationality, country of residence.

2. Dynamic information that may change over time such as qualification, job, English requirement and skills.

These basic data in user profile are used to infer the context information and user preference.

3. User identity [41], [9] defines the preferences and privileges the requester has with regard to a service. Identity holds more personalized data about the requester context, which allows the system to better rank the services that fit the requester requirements. A user may have a scholar identity or self-funded student identity.

This work focuses on making a user profile and defines identity using an explicit method.

3.2.3 Ranking Unit

The ranking unit is a central processing component that is used to match and rank trustworthy context-depended services. Ranking unit takes the queries and services as inputs and computes a matching score using a similarity measure formula [2]. The formula supports deferent data type (string, integers, numeric) and also takes into consideration the semantic of feature to provide better values (LB,MB or EB). A more in-depth explanation of this unit is performed in section five.

3.2.4 Context-Aware Unit

Since we aim to build a context depended application, the ranking process in this work is not based on assumptions. It relies on domain knowledge base and reasoning engine. Domain knowledge base contains important facts about service provider and service requester. The reasoning engine is an intelligent component that can use these facts and guides to provide a context-based ranking.

3.2.5 Explanation Unit

This unit is responsible for generating good explanations for the ranked results. For example, services can be grouped by their context information and by their trustworthiness information. Each group is labeled with a label explaining the characteristics of the services for instance: "Services suit your context", "Trustworthy services suit your context"; therefore, the service requester can understand how the services are related to each other and why these services are presented to them. The authors in [32] stated the guidelines for building recommender system. They described the design guidelines for providing explanation as follows:

“*Consider explaining why the system recommends the suggested items. These aspects can be highly correlated to users’ satisfaction, sense of control, and trust-inspired behavior intentions, such as the intention to save effort and the intention to return.*”

3.3 Matching and ranking context

Service ranking is the process of ordering the services according to their approximate relevance to a given query. Incorporating context-awareness in service ranking can greatly enhance the relevance of service ranking results and reduce the distraction to the service requester. Without consideration of context-awareness in service ranking, the result of service ranking turns to be less relevant to service requester's context and thus can lead to frustrations and reduced use of service.

This incorporation increases the challenges of how to provide suitable services to the right users with the right form (most suitable to the device) at the right time, under the full consideration of context rule.

In order to solve these challenges, we need to model context information as a combination of service requester context, service provider context and context rules. Following we explain how to treat and incorporate them into ranking process in more details.

3.3.1 Context rules

The context rule is a condition, which restricts the service eligibility for a specific service requester at service ranking. Service providers or local laws of service requester can define rules to determine the type of services to be selected.

In [7], the author expressed context rules as a logical expression statement. These rules are defined by the service requester context features. They act as filters for all the services that do not suit the user context. Therefore, this approach dramatically reduces the irrelevant services. As an example, the movie downloading service has some age restriction rule. For example, the rule age > 18 might be used to determine whether to provide video services service.

3.3.2 Service provider context features

According to [42], context adaptation influences all three applications dimensions: service adaptation, content adaptation and UI adaptation, where services refers to the services of the application, content is the exchanged data with the user, and UI is the visualization and presentation. Therefore, we use the service provider context features to adapt and personalize the content "features' values" based on the context of service requester and execution time.

3.3.3 Service requester context features

Mainly, we distinguish between service requester context features and non-context features in the ranking process. Service requester context features are must-have features that need to be fulfilled and satisfied. For example, if English score for scholar student is 6.5, the top results should only contain universities that accept students' score equal or lower than 6.5. A student cannot be satisfied by providing in the top results any other universities that required 7 or 7.5. English requirement can be identified as a must-have requirement. It is important to note that system should be able to anticipate the threshold value and its direction lower than or greater than for service requester context features.

Non-context features are nice to have features where service requesters' preferences are flexible, which may or may not be exactly satisfied. Service requester accepts services that closely fulfill their non-context features and it's semantic. The fee can be identified as a nice-to-have feature.

With respect to X-algorithm [2], the essential option is a feature option, that can be set for each feature independently. This option helps to rank the matching service that best suit essential feature to be ranked higher in the list. This definition completely fits with service requester context features "must-have features “definition. Therefore, the essential options are set for these features.

We aim to reduce the number of features in the query entered by the requester. Therefore, service requester context features should be hidden from the requester and so that the requester does not feel the existence of these features. The system considers these features as essential options, and, in order to solve the tradeoff between these features the domain expert specify the regular weights for them. Based on the experimental results, we found that, if the user chooses essential options for non-context features, it may prompt to rank the service that does not satisfy the service requester context features in the top. Therefore, we removed essential options from non-context features' query.

3.3.4 Context threshold value

Our aim requires us to identify the method that classifies the services into two groups: (results suit the context) and (results did not suit the context). Therefore, we need to understand the context features in terms of data types in more detail. That is, based on service requester context feature and its semantic (LB or MB); the system anticipates the threshold value and its direction lower than or greater than. For instance, English requirement is a context feature and can be defined as {type= numeric, semantic =LB, Mood = EB, value = 6.5} this implies that all services lower than or equal to 6.5 are considered as context services. The second example for string features is "type of qualification". It can be defined as {Name = type of qualification, type = string, semantic = EB, value = MRes}. This implies that all services equal to 1 are considered as a context services.

The system designer can also define the kind of threshold that the system should take into consideration. For example, if the context feature is computed by using two parameters to compute the distance such as in GPS, the system designer needs to define the threshold value as a certain value so all the services less than or greater than the threshold value are considered as a context services.

3.3.5 Context dominance problem

Context dominance problem arises when the Top-k results in large data set typically include some services that are satisfied service request context "SRC" requirement but does not fulfill the other features "Non-context" thereby they can't meet the requester requirement, so end up with top services that people do not like. Solving the context dominance problem is done by selecting key features.

3.3.6 Key features

A key feature is a feature option, that can be set for each feature independently. This option helps requester to exclude services from *"services suit the context* "ranking list if the key features for the non-context feature are not valid. The labeling for this list is changed to *"Services suit the context and key"*. The excluded services are inserted into the top group of *"different services suit the context"*. That is we do not lose the ranking scores.

3.4 Matching and ranking trustworthiness

The ranking process is based on context and non-context features. However, the result is large numbers of available services providing similar or even identical functionality; therefore, requestors are forced to choose between them. This adds pressure and more responsibility on the requester when selecting an appropriate trustworthy service. To mitigate requester concerns, and to help them in selecting trustworthy context services, we integrate trustworthy features into ranking process so the service requester will be able to take a better decision when selecting a service guided by a set of trustworthiness scores.

The challenge that arises is that trustworthiness feature is a composite feature that contains sub features. For example, security involves, among many other attributes, authentication, authorization, and integrity; safety involves timeliness and liveness; reliability involves failure and repair models.

A second example for rating a hotel based on sub-criteria: location, cleanliness, communication and check-in. It is possible that some users want to rank the services based on the overall score, some user wants to rank based on sub-criteria. Thus, the services might be compared according to their sub-criteria instead of whole criteria.

The ranking problem in this work is solved using vector space method that makes it difficult to model composite features. Composite features can be represented as a hierarchical model. Like any sort of hierarchical relationship among people, things or object can be modeled as a tree. Therefore, we suggest using graph "Tree “techniques to model these features and to rank it. The tree is a special case of the graph. It consists of a root, edges, nodes, and leaves. Each composite feature represents as a sub-tree. Each node represents a feature with its associated values and semantic, each edge represents a weight and mode. Figure 3.3 shows the graph tree technique to model trustworthiness features.

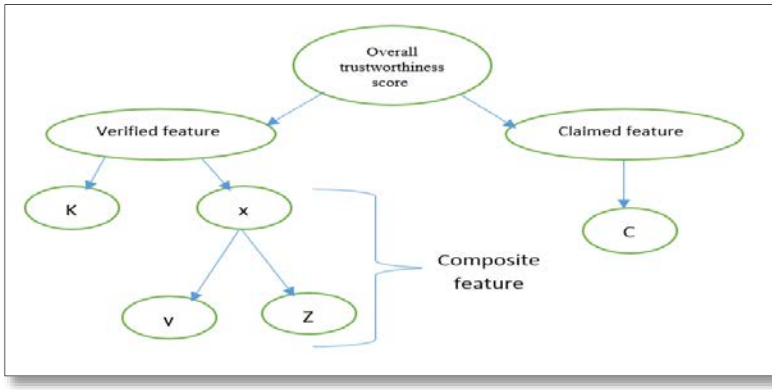

Figure 3.3 Graph tree technique to model trustworthiness features.

At the first level : we have two sides the left side is for " Verified features " that have a higher weight (0,0.001 ,003 ,004 ,005) and the right side is for "Claimed feature ".Since this claimed features may be biased, subjective or even malicious, we give it a lower weight than verified features (0,0.0001,0.001).

We employ the equations by [2] for calculating the similarity score for each node, and then we multiplied each node value with its edge "weight". The way to traverse the trustworthiness tree is post order: "Left, Right, Root" so starting from bottom to top. The basic root is to represent the overall score. The user should give weight to the composite feature and give weights for each sub features, the values for the root of the composite feature comes from the summation of all sub features' score.

After computing the overall score for trustworthiness features, we found we have two independent ranking lists. The first list is for context and non-context features, the second list is for trustworthiness features. To combine them we faced three cases:

1) When we have two services with the same trustworthiness score, the algorithm gives priory to the service ranked higher based on context and non-context features.

2) When we have two services with the same rank for context and non-context feature, the algorithm gives priority to the service that ranked higher based on trustworthiness features.

3) When we have two services with different ranking scores, for example, the service ranked lower in the group of services did not suit the context but it ranked higher based on trustworthiness, which one should take the priority. The user should be aware of the fact that service always came at the cost of the other. Consequently, we suggest to let the user decide which one should take the priority. This issue is resolved in the following section.

3.4.1 Priority option

Priority option is a query option that is can be set for the entire query. Users need to decide to go with one of the following choices: (Trustworthy services - Trustworthy context services - Context services).

- Trustworthy services: are related to the services that met the context rule and ranked based on trustworthiness features. This choice neglects the service request context features and non-context features.

- Trustworthy context services relate to the services that met the context rule and context features and then re-rank them based on trustworthiness features. All the services that fulfill the service requester context requirements are grouping and ranking the services in one list with a label "service suit the context" in order to identify them. This list is re-ranked based on trustworthiness features. Therefore, each group is re-ranked based on trustworthiness features.

- Context services: relate to the services that met the context rules and context features, the services here ranked based on context and non-context features. This choice neglects the trustworthiness features.

Figure 3.4 illustrates the priority option. It shows how different priority affect the ranking list. For example, S7 did not suit the user context, therefore; it ranked lower with *"context service priority* “but with *"trustworthiness priority"* S7 is first. If the user decides to go with *"trustworthy-context priority"* each ranking list in "context service priority" are re-ranked based on trustworthiness features.

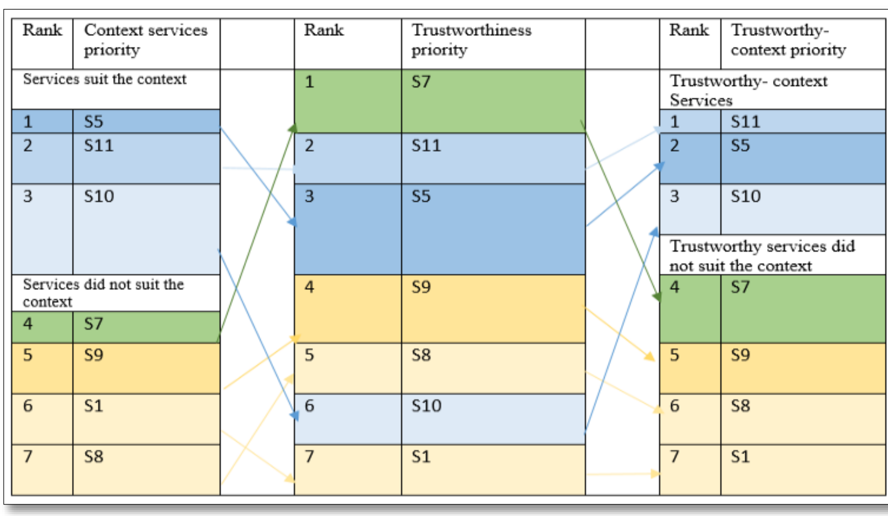

Figure 3.4 visualize the priority option.

3.5 The Proposed Architecture

Figure 3.5 shows our proposed architecture for a formal approach for matching and ranking trustworthy context-depended services framework. The architecture follows the spirit of "pipes and filter architecture ". It contains components (filters) that process data and connections (pipes) that move the data emitted by one component to the next one for consumption. Each filter performs a particular task that is needed in the application. It does so by reading a stream of data that it received at its input interface, performing some operation, and then evicting a stream of data at its output interface. A pipe is only responsible for transmitting data between filters; it does not carry out any processing of data [43]. The rest of this section presents a detailed information on the main elements shown in the figure 3.5.

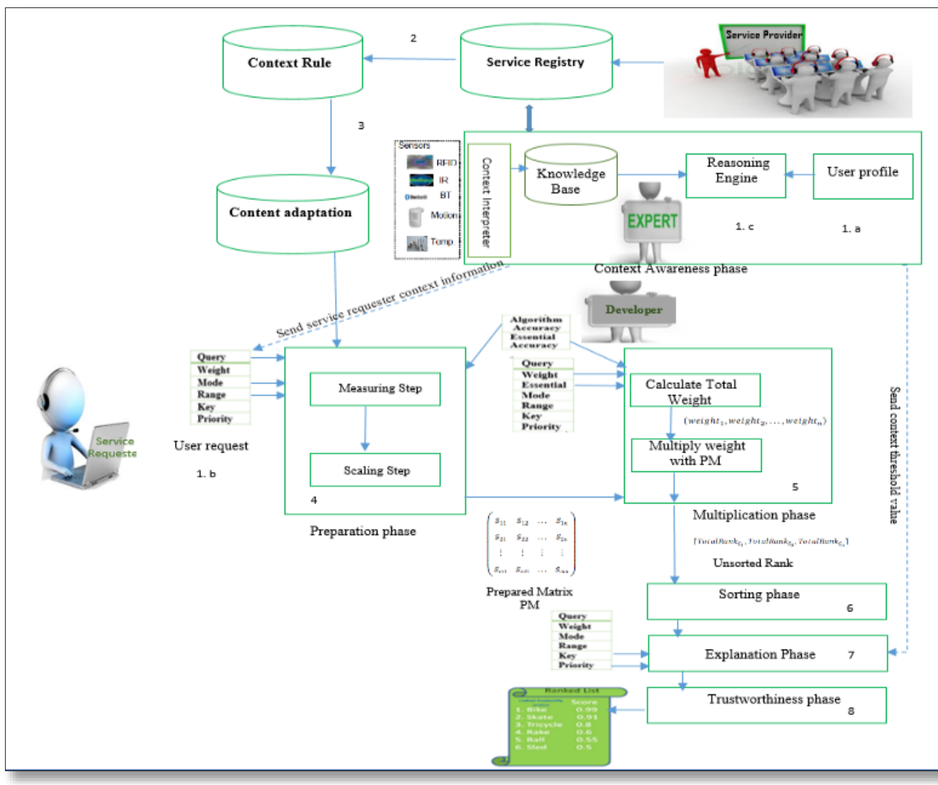

Figure 3.5 Framework of formal approatch for matching and ranking trustworthy context dependent services.

Service Registry

The service registry is the main component in SOC that is responsible for storing services published by service providers. It acts as a database that stores multi-features services associated with their semantic. Service requester can browse the contents of a service registry, and then query the system for seeking services that match user queries. The internal process and implementation of the service registry are outside the scope of this thesis.

Context Awareness component

This component includes knowledge base, reasoning engine, user profile and context interpreter.

Knowledge Base

The knowledge based is domain depended. It contains important facts about service provider and service requester. These facts are supplied by a domain expert.

Reasoning Engine

Reasoning Engine is an intelligent component that can use the facts stored in the knowledge base and guides to provide a context-based ranking.

Below are the reasons for using the reasoning engine:

1) To infer context features based on the service requester's identity.

2) To anticipate the threshold value for service requester context features and its direction.

3) To infer context information and context rule that restrict the service eligibility for a specific service requester at service ranking.

4) To infer preferences, because preferences are context depended and not all context information gathered directly from context sources sometimes we need to study the correlation between features to infer the context information.

Context Interpreter

The context interpreter is responsible for interaction with sensors to turn sensor data from low-level data to high-level data context information. For example, if the sensor sense the GPS coordinates, then the interpreter maps this coordinates to a street address. The internal process of context interpreters is outside the scope of this thesis.

User Profile

The user profile is an important component in generating context services. In general, the user profile in our work is composed of three sections:

1. Personal information about the user, which is final and not subject to change such as name, the birth of date, nationality, country of residence.

2. Dynamic information that may change over time such as qualification, job, English requirement and skills.

These basic data in the user profile are used to infer the context information and user preference.

3. User identity [9] [41] defines the preferences and privileges the requester has with regard to a service. Identity holds more personalized data about the requester context, which allows the system to better rank the services that fit the requester requirements.

Context Rule

This phase is responsible for the applying context rule, which restricts the service eligibility for a specific service requester at service ranking. It acts as a filter for all the services that do not suit the user context. Therefore, this approach dramatically reduces the irrelevant services and saves a significant amount of computation resources done in preparation, multiplication and sorting phases.

The goal of context rule phase is the following:

1. Apply context rules.
2. Generate a list of services most suitable to the service requester.

Content Adaptation

This phase is responsible for applying SPC conditions to personalize the content "features' values" based on the context of service request. The results of this phase proceed to the next phase, which is the preparation phase.

Preparation phase

This phase is divided into two steps: measuring and scaling. In the measuring step, the attributes of query and service are involved in a pairwise comparison. This comparison produces un-normalized numbers that are based on different scales. Therefore, the results of the pairwise comparison are used in the next step, which is the scaling step. Thus, the goals of preparation phase are:

i. Perform a pairwise comparison between the attributes of query and available services associated with its semantics for each feature.
ii. Consider the following parts of the user request: query, mode, range, and algorithm accuracy.
iii. Normalize the results of the pairwise comparison to a common scale.
iv. Generate the Prepared Matrix (PM).

In the measuring step, the algorithm uses query, mode, range, and the algorithm accuracy as specified in the user request. In addition, the available list of services associated with its semantics is considered as an another input in this step. Hence, the algorithm takes each feature of the query and compares it against a feature of one service using the defined similarity measure as in [2] .

In the scaling step, the produced numbers from the measuring step are justified into a common scale. Hence, each pairwise comparison result passes through a scaling method. This scaling method adjusts the results to one common scale that is used for all other results. Finally, results of the scaling step are recorded in PM. The rows of this matrix are the features of different Services with respect to the features of query. That is, the i_{th}row is PM matrix contains the scores of the i_{th} features of available Services with respect to the i_{th} feature in query. The columns of PM matrix are the available services.

Multiplication phase

In this phase, the results are tailored to suit the service requesters' context and their needs. Service requester explicitly trades off some features for others using weights. Weights are used to determine the level of importance for each feature. Hence, weights, ‘essential’, and ‘essential accuracy’ are considered in the calculations. However, because ‘essential’ option is basically an extra weight assigned to the essential feature to outweigh the importance of other non-essential features, the weight vectors divided to two sub-vectors; Regular Weight vector and Essential Weight vector [2]. Consequently, Weight = Regular Weights + Essential Weights

Because service request does not need the existence of the context features, at the very beginning, the domain expert specifies the regular weight for the service request context features, while the service requester specifies the regular weights for non-context features. Service request context feature considers as an essential feature to outweigh the importance of other not- context features. Domain expert specifies the regular weights for each service request context feature to specify the level of importance to perform the tradeoff between them.

This phase is responsible for performing the following tasks:

1. Calculate the total Weight vector (Regular Weight + Essential Weight).
2. Multiply PM by the Weight vector.
3. Generate the Unsorted Ranks (UR) list.

Sorting Phase

This phase receives UR vector from the Multiplication Phase and applies a sorting function to sort the services. The output of this phase is what we call sorted ranks. Thus, this phase performs the following tasks:

1. Apply a sorting algorithm on UR vector and sort it in decreasing order.
2. Generate the sorted ranks list.

Explanation phase

The explanation phase uses the following as inputs: the sorted rank list, priority requirement, key requirements and context threshold value. With these inputs, based on the specified priority "context services" or "trustworthy context services", the following procedure will be followed.

First, the services are classified into two groups according to the context threshold value. Services that fulfill the context requirements and satisfied the threshold values grouped and ranked in one list with a label *"Services suit the context"* to identify them. Otherwise, the services is ranked in another list and labeled with *"Services did not suit the context"*. The services in each list are evaluated based on key values and produce four lists:

- *"Services suit the context and key"*: related to the services that satisfy the context and key requirement.
- *"Different services suit the context":* related to the services that satisfy the context requirement but do not fulfill the key requirements.
- *"Different services suit your key* “related to the services that satisfy the key requirement but do not fulfill the context requirements.
- "*Services did not suit the context and key":* related to the services that do not fulfill the key requirement and context requirements.

If the service requester specifies the priority to context services, then the results are provided to the requester. If the priority is set to "Trustworthy context services" then the lists are sent to the trustworthiness phase.

This phase performs the following tasks:

1. Apply the context threshold values.
2. Apply the key requirements.
3. Categorize the ranking result and each category are labeled with a label explaining the characteristics of the services.

If the service requester set the priority to "Trustworthy services", then the services are forwarded directly to trustworthiness phase and skipped the explanation phase. That is, the services are ranked based on trustworthiness features.

Trustworthiness phase

In this phase, the result of a query with trustworthy context services priority or trustworthy services priority is used as input in this phase. Then, the overall scores for trustworthiness features are computed. That is the services in each list are re-ranked according to the trustworthiness overall score. In case "trustworthy services" priority the following list produced:

- *Trustworthy services:* related to the services that met the context rule and ranked based on trustworthiness features.

In case "Trustworthy context services" priority without key requirements, the following lists are produced:

- *Trustworthy- context Services:* related to the services that fulfill the context requirements, and satisfy the threshold values, and re-ranked according to trustworthiness scores.
- *Trustworthy services did not suit the context:* related to the services that do not suit the context threshold value and re-ranked according to trustworthiness scores.

In case "Trustworthy context services" priority plus there is a key requirement and the following lists are produced:

- *Trustworthy services suit the context and key:* related to the services that satisfy the context and key requirement re-ranked according to trustworthiness score.
- *Different trustworthy services suits the context:* related to the services that satisfy the context requirement but do not fulfill the key requirements. These services are re-ranked according to trustworthiness score.
- *Different trustworthy services suit the key*: related to the services that satisfy the key requirement but do not fulfill the context requirements. These services are re-ranked according to trustworthiness score.
- *Trustworthy services did not suit the context and the key*: related to the services that do not fulfill the key requirement and context requirements. These services are re-ranked according to trustworthiness score.

The goals of this phase are the following:

1. Compute the overall scores for composite and non-composite trustworthiness features of each service.
2. Re-rank the services in each list according to trustworthiness score.
3. Generate explanations.

3.6 Who are the stakeholders?

This section identifies the relevant stakeholders in our proposed framework as shown in figure 3.5.

1. Service provider: Responsible for providing services and publishing them in the service registry.

2. Service requester: browse the contents of a service registry, and then query the system for seeking services that best match his/her queries.

Our goal is to extend X-algorithm [2] from user-centric to context-awareness. It consists in adding further complexity to the development process and paying attention to some aspects. Therefore, we need:

3. Domain expert: determine the important features with its sematic and supply the knowledge base with important facts about service provider and request and their context information.
4. System developer: determine the essential accuracy and algorithm accuracy.

3.7 Formal Specification

In this section, we demonstrate the contribution of this thesis, which is represented by the formalization of the proposed framework that described in a natural language previously using logic and set theory.

In order to formalize the proposed framework, we have to define the following notations used in all subsequent definition:

- T denotes the set of all data types, including abstract data types.
- $Dt \in T$ means Dt is a data type.
- v:Dt denotes that v is either constant or variable of type Dt
- Xv is a constraint on v.
- C denotes constraint which defined as the set of all such logical expressions defined over features.

Service definition: services with their descriptions "features" are storing in service registry SR.

Definition 1: Let SR be set of services in service registry, SR= $\{s_{1,}, s_2, \ldots\ldots, s_n\}$.

Feature definition: Each service can be described by many features, which provides sufficient information that is unique to a service.

The set of features is F= $\{f_{1,}, f_2, \ldots\ldots, f_n\}$. Typical features of a service are its functionality, its price, and other properties that describe the qualitative or quantitative characteristics of the service. These features are classified into two groups: i) the ranked features group that are included as a criterion in ranking. RF = {x: x is a ranking feature}; and ii) non-ranked features which store additional information and are not included as criteria in ranking algorithm. NRF = {x: x is not a ranking feature} as an example, email, summery of the course, links.

Definition 2: A ranked feature RF has a name, type, and semantic associated with this feature. RF is defined as 3-tuple RF= (n, fDt, fs) where n : string is the feature identification name, fDt :data type of this feature, fs is feature semantic which is defined as a set of ordered pairs fs= {(x,y)| x:feature ,y ∈ {MB, LB, EB} }. Where MB stands for more is better, LB stands for low is better and EB stands for exact is better. As an example (Fee, numerical, LB).

The set of ranked features is RF = {(Dt, vα, s) |Dt ∈ T, vα: Dt, s ∈fs}.

Context Information: Context is any type of information used to characterize an object or situation [38] which helps to obtain the desired and most relevant services according to user requests.

Definition 3: Context information CI is formally specified, as defined in [12], Let τ:$DIM \rightarrow I$, where $DIM = \{X1, X2,..., Xn\}$ is a finite set of dimensions and $I = \{a1, a2, ..., an\}$ is a set of types. The function τ associates a dimension to a type. *Let* $(Xi) = ai, ai \in I$. We write CI as an aggregation of ordered pairs (Xj , vj), where $Xj \in DIM$, and $vj \in \tau (Xj)$.

In SOC, context is any element that could affect the service provision and execution operation. Therefore, it is necessary to take into consideration the context of the service provider, service requester, the execution time, and context rule that restricts the service eligibility for a specific service requester at service ranking. A context rule is a case that might be true for service delivery in some contexts and false in some others.

Definition 4: A context in SOC is a tuple CSOC = (SPC, SRC, CR) where SPC is a service provider context and SRC is a service requester context and CR is the context rule.

SPC is any information that is related to the only service provider (SP) that characterizes the context of SP and the context of execution time. In general, SPC features have more than one possible value. The context of service requester is used to select one value. Therefore, for each possible value، a context constraint is defined such that only one context constraint can be true at an instant. Therefore, only one value will be selected. A context constraint is a special type of constraint that is used to decide whether a specific value for service provider should be selected. The decision is based on evaluating a logical expression defined over the values of the context of SR associated with the query. The value is given only if the constraint evaluates to true.

Definition 5: The set of SPC features can be defined as $SPC = \{p \in RF \mid p:(Vq, XVq) \}$ where Vq denotes the set of values of data type q and XVq denotes the set of constrains defined over the set of values. If *Vq* has n values than there must be n number of context constraints *XVq* defined for the values of *Vq*.

As an example: For most UK universities the fee feature has two values (one for home student and one for an international student) and the selected value is determined by student's nationality or by the country where he/she is resident.

SRC is any information that is related only to service request (SR) that characterizes the context of SR while requesting or receiving a service. As an example location.

From definition 2 some of the features in RF set to satisfy some of the following condition are called an SRC feature:

1) Tangible: attributes that can be sensed implicitly form another context and quantified. *Sense(x)*
2) Dynamic attribute: change over the time, an example of a dynamic attribute for a printer is the number of prints jobs in the print queue. *Dynamic(x)*
3) Not-privacy attribute: because privacy considered as a fundamental right, so it is difficult to infer it. *Privacy(x)*
4) Infer it from data "user profile" or environment. *Infer(x).*
5) Computed attributes. *Compute(x1,x2)*
6) Negotiable skill. *Negotiable_skill(x)*
7) In our research, we have been using the five dimensions *WHERE, WHEN. WHAT, WHO*, and *WHY* to construct any general context. Let *CD* is the set of Context Dimension to define SRC $CD = \{what, where, who, when, why\}$

Definition 6: SRC $=\{x \in RF \mid x: (sense(x) \vee Infer(x) \vee Dynamic(x) \wedge \neg\, privacy(x)) \vee x \in CD\}$

Definition 7 : Context rules (CR) are a logical expression statement which restricts the service eligibility for a specific (SR) at service ranking, CR : Δ denotes a logical expression that validates by the service provider or local laws of service requester CR $\in$ C defined over the set of SRC. We write CR = {(SRC, XSRC)| XSRC $\in$ C} to represent the set of context rules. As an example, the movie downloading service has some age restriction rule. For example, the rule age > 18 might be used to determine whether or not to provide video services service.

Trustworthiness features: Trustworthiness is divided into two parts: i) Domain expert feedback is a verified feature that it comes from another trusted independent organization or it has to be accompanied with a proof related to service quality features such as (safety, availability, security, reliability and timeliness); and ii) Random public feedback which is claimed by the service provider or by consumers such as consumers' ratings and reviews.

Definition 8: Trustworthiness is defined as a tuple (Ȅ, Ṙ) where Ȅ is a domain expert feedback and Ṙ is random public feedback.

Ȅ can be defined as Ȅ = {(f,Ṫ)| f∈ RF , Ṫ ∈ {x: x is trusted organization V trusted proof }}, the pair (f,Ṫ) represents a verified trustworthy feature f which comes from trusted independent organization Ṫ or verified features that accompanied by proof.

Ṙ = { (ʆ,ṝ)| ʆ ∈ RF, ṝ ∈{ r:r is consumer rating V service provider } } the pair (ʆ,ṝ) represents a claim trustworthy feature ʆ which comes from consumers' ratings or claimed by service provider ṝ.

Definition 9: The set of trustworthiness features can be defined as TF= {tf ∈ RF|tf ∈ Ȅ V tf ∈ Ṙ }.

Composite trustworthiness feature: Trustworthiness feature is a composite feature where one feature contains sub features. For example, safety involves timeliness and liveness; Reliability involves failure and repair models.

Definition 10: Trustworthiness represented as the tuple (CT,ST,Ỹ) where CT is a set of trustworthiness features and ST is set of sub-features related to trustworthiness features and *Ỹ :{ct1,ct2,ct3, ctn}*→ *Ὸ{st1,st12, st1n}* is a function that associates each trustworthy feature with it is subset features . if *Ỹ(CTi)= 0* then means this trustworthy feature CTi is not a composite feature .

Non-context feature: The rest of the ranking features that is not categorized as service request context feature or trustworthiness features are considered as a non-context feature.

Definition 11: NCF= $\{ncf \in RF|\ ncf \notin SPC\ \Lambda\ ncf \notin TF\}$.

Query: Query has constraints for the format of the input request: all possible features should be available in a query definition, the query include 3 parts context features, non-context features and trustworthiness features associated with their weights to model the level of importance for each feature. A higher weight indicates a higher importance. As in [2] we use numerical values from 0 to 0.005 to indicate the selected weight for a specific query feature such than 0 indicates not consider feature and 0.005 high importance. In addition, the user needs to define the mode to model the preferences exact match or best match.

Definition 12: A query is defined as q=(ḟ, ҫ, Ṱ, ̔P, Ḱ, Ess, AlgAcc, ṵ) where ḟ is a non-context query , ҫ is a context query, Ṱ is a trustworthiness query. ̔P is priority requirement set for the entire query (x ∈ { Trustworthy services, Trustworthy context services,Context services }). Ḱ *is a key feature.* Ess is Essential Accuracy, AlgAcc is Algorithm Accuracy that determined by the provider. ṵ is the user profile.

Definition 13: The non-context query is defined as ḟ = (ḟf ,Ṁ, ,ẅ) where ḟf is non-context features requrment. Ṁ: (x ∈ {best mode, Exact mode})→ (y∈ {ḟf })is a function that assign preferred mood to the feature of query. ẅ : (x ∈ {not consider , insignificant, low, normal, significant } where 0 denotes not-consider, 0.0001 denotes insignificant, 0.001 denotes low , 0.003 denotes normal,0.005 denotes significant) → (y∈ { ḟ}) is a function that assign weights to the feature of the query.

Definition 14: The context query is defined as ḉ = (f,Ṁ, ,ẅ, Ȩ) where f is a service request context features requirement . Ṁ and ẅ are defined by a domain expert as in the not - context query. Ȩ is the Essential weight for service requester context.

Definition 15: The trustworthiness query is defined as Ţ= (Ȅ, Ṙ, Ṁ, ẅȄ, ẅṘ) where Ȅ and Ṙare identical to the previous definition. Ṁ is defined as in the non -context query. And ẅȄ: (x ∈ {not consider, low, normal, significant} where 0 denotes not-consider, 0.001 denotes low, 0.003 denotes normal, 0.005 denotes significant) → (y∈ {Ȅ}) is a function that assign weights to the verified trustworthy features. ẅṘ: (x ∈ {not consider, normal, significant} where 0 denotes not-consider, 0.0001 denotes normal, 0.001 denotes significant) → (y∈ {Ṙ }) is a function that assign weights to the claimed trustworthy features.

Correlation between features: Sometimes we need to study the correlation between feature to infer the service request context value. In order to build context-depended application, there is additional work. Because preferences are also context depended, the system has to study the correlations between features. For example, the preferred choices for students employed as a university teaching assistants are master of research but for non-academic student master of taught might be preferred. That is the correlation between job and type of qualification used to infer the context value preferences.

Definition 16: Correlation between feature can be formalize as let D is a set of dimensions = {d1,d2,….dn}, let Ta is a set of tags value {t1,t2,….tn} and let P is a set of inferred values {p1,p2…pn}.

So $d1(x) \rightarrow t1(x)$

$d2(y) \rightarrow t2(y) \quad \dashv \; d1(x) \wedge d2(y) \rightarrow p1(z)$.

Example: Suppose we have the following facts: 1) if the user has a bachelor (B) then his future qualification will be Master of research program denoted by Msc"Re" or Master of taught program denoted by Msc"ta".

2) If he is a bachelor his job title will be teaching assistant (L) or not.

3) If he is teaching assistant, it is not permitted to study Msc"ta".

Therefore, if he has a bachelor and he is a teaching assistant, his qualification will be Msc"Re".

To prove, first we need to symbolize the facts:

1 B → Msc"Re" V MSc"tau".

2 B → L V ¬ L

3 L → ¬MSc"tau". ┤ B Λ L → Msc"Re".

Proof : "induction method "

4 B Λ L

5 (L V ¬ L) Λ L "from 2"

6 $(L \wedge L) \vee (L \wedge \neg L)$ " idempotent rule".

7 $L \vee F$ "negation rule"

8 $L \rightarrow$ Msc"Re".

3.8 Conclusion

In this chapter, we presented our new proposed architecture for matching and ranking trustworthy context-dependent services. We also formalized the proposed architecture to provide a precise description of the system. This chapter introduced to the reader the fundamental information that helps them to apply the proposed architecture in many diverse application domains.

In the next chapter, we present our real-word case study on King Abdul-Allah scholarship program to prove the success of the proposed architecture.

Chapter 4

A Case Study on King Abdullah Scholarship Program.

Chapter IV

A Case Study on King Abdullah Scholarship Program.

4.1 Introduction

In this chapter, we provide a case study on King Abdul Allah scholarship program to illustrate the success of our proposed architecture to achieve the required goal and to evaluate its results. First, we are going to introduce the problem we are trying to solve through this case study. Then, we explain the case study in more details. After that, we collect and classify the data. Finally, we discuss the result and examine the accuracy manually.

4.2 Problem of current choosing algorithm in university

There are many ranking systems for universities around the world such as Quacquarelli Symonds World University ranking QS, Shanghai ranking and national ranking systems (such as the complete university guide for UK universities and Asia's best universities for Asian universities). The majority of students consider the position of the University of their interest in the ranking lists. However, the student should not count on these ranking systems as a guide for choosing a university and they should look for additional information before making a selection of institution such as course details, accommodation and fees. The manual and traditional selection requires students to visit every university website looking for their preferred courses. Some students prefer to talk to advisers and recruiters and get help. Students do not keep in their mind that those advisers and recruiters might have a financial interest to direct students to certain universities. Therefore, the risk of applying to a wrong institution is increased.

Instead of this manual and traditional selection, there are many searching websites that help students in searching and finding their courses such as find master [44], prospects [45], course guide chooser [46] and QS course finder [47].

Recently in 2014, U-Multirank [48] launched their website as an interactive web tool that helps students to compare institutions with similar institutional profiles and allows students to develop personalized rankings by selecting performance features in terms of their own preferences.

Although theses websites provide searching by features, the search process is based on a matching process that acts as filters. Therefore, it filters out all other options that do not exactly match a predefined value. Then the search results are sorted in ascending or descending order based on one feature, which is most of the times insufficient. In real life, there is no choice that meets all the student preferences. Therefore, there is a need for a ranking process that defines the best options.

Students around the world would appreciate a platform that allows them to consider when they are searching for a legitimate university accredited by their countries and help them to find the right university that recognizes their existing qualifications.

Currently, there is no published tool that takes into consideration the rich features of universities, its trustworthiness capabilities and context dependencies. Therefore, this is the motivation of this case study, providing a solution for ranking multi- feature trustworthy context depended on universities. This case study particularly investigates King Abdullah Scholarship Program described in the next section.

4.3 King Abdullah Scholarship Program

King Abdullah Scholarship Program (KASP) was created in 2005 by sending Saudi students to the United States. KASP now is the largest scholarship program in Saudi Arabia's history and the scope of the scholarship program was broadened to include a number of advanced countries like the United Kingdom, Australia, and Canada.

Saudi scholar students have obtained bachelor, masters, and doctorate degrees as well as medical fellowships as a result of this scholarship program. The major of study that students are allowed to enroll in are chosen carefully by the Saudi Arabian Government and Ministry of Education (MoE) based on the perceived need of the country and economy [49].

Although MoE decides on the major that the students can choose from each university, it does not choose the course program for each student. An accredited list of universities is compiled for all available majors that the students can choose from with wider alternatives. The accredited list of universities is subject to periodic review in order to meet the needs of the Saudi labor market [50]. It is the student responsibility to search and find the suitable university and major from this list of accredited universities by MoE. Students search themselves for the right courses. Now it is time to automate and facilitate the process of searching for the right courses and make it much easier than before.

The Scholarship program KASP has some of the restriction rules for choosing universities and program even if the university is included in the accredited list [50].

Some of the restriction rules and conditions are:

1. Students are not permitted to enroll in programs designed specifically for international students.
2. Students are not permitted to enroll in part-time or distance learning program. They must be a full-time student.
3. Students are not permitted to enroll in fast-track bachelor's degree programs.
4. Postgraduate Students are not permitted to enroll in vocational programs or non-specialized program.

Due to time constrains and the difficulty to collect data related to these conditions for each university and program, we restrict our work only to rule 2 above.

4.3.1 Types of students:

Three types of students fall under the umbrella of KASP and therefore, we have three identities:

- Scholar student: a student who studies abroad with a scholarship with complete funding.
- Self-funded student: a student who is able to study abroad without a scholarship
- Scholar employment: a student whom funding is covered by the employer. Typically, this type of students studies abroad to learn a specific skill or study in scientific or high-tech fields. Once their study is completed, they are expected to return to their employment in Saudi Arabia.

Each identity has its own context and encompasses multiple features. For "Scholar student" identity “fee” feature, “mode of study” and “qualification” features will be hidden because the scholarship typically covers full academic fees and students are not permitted to enroll in certificate or diploma programs, part-time, online, or distance-learning courses. If the identity is “Self-funded student “fee” feature, “mode of study”, “type of qualification” features will appear in the search.

4.4 Data Collection

To be able to apply our proposed architecture in any application domain, we have to understand the data in terms of 1) number of features. 2) Potential data types and its semantic. In addition, 3) categorize the features into context, non-context and trustworthiness features based on criteria defined in the previous chapter.

No dataset with all of the required features in universities' domain was available. Therefore, we had to collect raw data from different web pages. Data was extracted from the course guide chooser [46] supplied by UCAS. The data was organized in Comma-Separated Value (CSV) sheet format. Context information was extracted from MoE for Saudi Scholar and from supreme education council for Qatari Scholar. The features, their data types and semantics are as follows:

- **Subject**: a string value describing the desired subject area of study. Thus, it is EB feature.
- **Program Title**: a string value describing the main subject of the program. Thus, it is EB feature .
- **University**: a string value indicating the name of the university offering the program. Thus, it is EB feature.

- **City**: a string value indicating what is the city of the university. Thus, it is EB feature.
- **Duration**: an integer value indicating the number of years the student is expected to spend to complete the program. The duration is better as it decreases. Hence, it is considered as LB feature.
- **Mode of study**: a string value indicating a preferred mode of study Full time, Part time, or Distance /online learning. Thus, it is EB feature.
- **Qualification**: a string value indicating a preferred degree program to study Bachelor's Degree, Postgraduate (research/ taught). Thus, it is EB feature.
- **Entry qualifications**: a string value indicating students' qualifications that they should have prior to entering higher or further education. Thus, it is EB feature.
- **English requirement**: a numeric value indicating the level of English scores for students who do not have English as their first language. The English requirement is better as it decreases. Hence, it is considered as LB feature.
- **Fee home student**: a numeric value indicating the cost of the program for the home student. The fee is better as it decreases. Hence, it is considered as LB feature.
- **Fee oversees student**: a numeric value indicating the cost of the program for oversees student. The fee is better as it decreases. Hence, it is considered as LB feature.
- **Accredited universities**: a finite set of universities accredited based on users' country. Thus, it is EB feature .
- **The program recognized by the different situation**: a string value describing the recognized certification. Thus, it is EB feature.

- **League Table Ranking**: a numeric value indicating the university's position in the league table in 2016. The league table is a domestic rankings table for UK universities. The rank is better as it decreases. Hence, it is considered as LB feature.
- **Student satisfaction**: a numeric value indicating the feedback of the programs' students to assess the program course and university. The student satisfaction is better as it increased. Hence, it is considered as MB feature.
- **Safety "crime statistics"**: A numeric value indicating the best and worst universities and colleges for student-relevant crime that reflected safety feature. The crime statistics is better as it decreases. Hence, it is considered as LB feature. Safety includes a set of properties that indicating the crimes most relevant to students. It includes three sub-features: burglary, robbery and violent crime.
    - **Burglary**: a numeric value indicating offenses where a person enters a house or other building with the intention of stealing. The burglary is better as it decreases. Hence, it is considered as LB feature.
    - **Robbery:** a numeric value indicating offenses where a person uses force or threat of force to steal. The robbery is better as it decreases. Hence, it is considered as LB feature.
    - **Violent crime:** a numeric value indicating offenses against the person such as common assaults, grievous bodily harm and sexual offenses. The Violent crime is better as it decreases. Hence, it is considered as LB feature. The definitions of burglary, robbery and violent crime were defined by the police [51].

Note:

- The crime statistics are derived from data produced by the police.UK, Ordnance Survey and the Office for National Statistics, and used under Open Government License v3.[52]
- The student satisfaction data comes from the National Student Survey (NSS).[53]
- League table ranking produced by the complete university guide organization in 2016. [54]
- Accredited universities data for Saudi scholar comes from the MoE [50].
- Accredited universities data for Qatari scholar comes from the ministry of education and higher education [55].

Regarding unranked features, the features that are not included as criteria in ranking, we include two feature as follow:

- **Program Description:** a string value describing the program so the students can look at the specific details of the programs that interest them.
- **Email:** a string value describing the email address for university so the students can get in touch with universities by email.

The table below illustrates the relationship between features and their categories, as well as between identities and their features.

Table 4.1 The relationship between features, their categories and identities.

Category	Identifier	
Category	Identity: Scholar Student, Scholar employment.	Identity : Self-funded student
Context Rule	Accredited universities, Entry qualifications, Qualification and Mode of study	Accredited universities, Entry qualifications, and Qualification.
Service Requester context	English requirement and type of qualification	English requirement and type of qualification.
Service Provider context	-	fee (home or overseas) student
Non- context feature	Subject, program title, University, City and Duration	Subject, program title, University, City, Duration and Mode of study.
Verified trustworthiness features	League Table Ranking, program recognized by different situation, Safety	
Claimed Trustworthiness Features	Student satisfaction.	
Unranked feature	Program description and email.	

4.5 Applying the proposed solution

We have implemented our architecture as a Web-based platform using HTML, C#.net and ASP.net MVC 5.The data is stored in SQL server 2014, and using Entity Framework 6.1 for connecting and querying the database. For matrix multiplication operation, we used Lightweight fast matrix class in C#. By using Bootstrap library, we make it responsive to any browsing environment such as desktop, laptop, tablet, or smartphone. For the sake of simplicity, we have included only eleven records ordered by league table ranking 2016. The records shown in tables 4.2 and 4.3 need to be ranked to show the benefits of our proposed framework. Plus, we considered the variables *EssentialAccuarcy* and *AlgorithimAccuarcy* to be preset to the values 0.01, 0.1 respectively.

Table 4.2. The eleven services and their context and non-context features.

Record Number	Title	University	Category	Duration	Mode Of Study	Qualification	City	English Requirement	Fee	
									Home	Overseas students
S1	Computer Science	University of Cambridge	Computer Science	4	Full-time	PhD	Paisley	-	£7,362	£23,889
S2	Computer Science	University of Birmingham	Computer Science	2	Part-time	MRes	Birmingham	6	£7,200	£19,200
S3	Advanced Computer Science	University of Birmingham	Computer Science	1	Full-time	MRes	Birmingham	6	£6,840	£17,960
S4	Advanced Computer Science	The University of Manchester	Computer Science	1	Full-time	MRes	Manchester	7	-	-
S5	Computer Systems	Heriot-Watt	Computer Science	3	Full-time	BSc	Edinburgh	-	£16420	£1820
S6	Network Systems Engineering	Plymouth University	Computer Science	1	Full-time	MSc	Plymouth	6.5	£5,660	£13,250
S7	Advanced Computer Science	University of Salford	Computer Science	1	Full-time	MSc	Salford	6.5	£4,845	£13,050
S8	Computing Science	Staffordshire	Computer information system	1	Full-time	MRes	Stoke	6	-	£11,500
S9	Networking and Data Communications	Kingston University	Computer Science	1	Full-time	MSc	Kingston	6.0	£5,900	£12,500
S10	Computing	Bedfordshire	Computer Science	1	Full-time	MRes	Luton	6	£3,996	-
S11	Computer Science	Bedfordshire	Computer Science	3	Full-time	MSc	Luton	6	£9,000	-

Table 4.3. The eleven services and their trustworthiness features.

	Verified features							Claimed feature
Record Number	Rank	Professionally Recognized by:	Safety" Crime statistics"			Safety Total score	Student Satisfaction
			Burglary	Robbery	Violent Crime		
S1	1	-	7.02	0.49	14.03	21.54	4.18
S2	18	-	8.15	3.14	18.13	29.42	4.08
S3	18	-	8.15	3.14	18.13	29.42	4.08
S4	28	British Computer Society	13.62	3.08	22.43	39.13	4.02
S5	37	-	-	-	-		4.09
S6	90	-	5.10	0.42	21.80	27.32	4.03
S7	96	-	12.60	3.09	23.44	39.14	3.93
S8	103	British Computer Society	6.25	0.92	23.89	31.06	4.04
S9	104	-	7.51	0.81	13.46	21.78	3.90
S10	110	-	9.99	1.25	18.08	29.32	4.04
S11	110	-	9.99	1.25	18.08	29.32	3.89

Case 1: *identity* has been set to "scholar student" and the *priority* has been set to "Trustworthy context services" for the entire query.

Here, in this case, we demonstrate how Ahmad and Omar, prospective postgraduate students, utilize our framework to find the suitable institutions and courses to apply to in the UK. They start by building their profiles and identify their identities as shown in tables 4.4 and 4.5, and they submit the same request query as in table 4.6, the title has a significant weight and the duration is set to best mode. University and city features have not considered weights, therefore, students are willing to accept any values for these features and the priority is set to trustworthy context services. The query for trustworthiness is shown in table 4.7, the rank is set to exact mode and safety and student satisfaction is set to best mode.

Table 4.4. The user profile for Ahmad.

User profile	
Name	Ahmad
Nationality	Saudi
Identity	√ KASP Scholar. Self-funded student. Scholar employment.
qualification	Bachelor
Job	teaching assistant
English Requirement	6.5

Table 4.5. The user profile for Omar.

User profile	
Name	Omar
Nationality	Qatari
Identity	√ SEC Scholar. Self-funded student. Scholar employment.
qualification	Bachelor
Job	teaching assistant
English Requirement	6.5

Table 4.6 Query for non-context features.

Non-Context Features			
Query	**Title**	**Category**	**Duration**
Values	Computer Science	computer Science	2
Weights	significant	normal	normal
Mode	EB	EB	BB
Range			
Key			
priority	Trustworthy services , √ Trustworthy context services, Context services		

Table 4.7 Query for trustworthiness features.

Trustworthiness feature							
	Verified Features						Claims Features
Query	Rank	Professionally Recognized by	Burglary	Robbery	Violent Crime	Safety	Student Satisfaction
Values	20	British Computer Society	10	2	12		4
Weights	normal	significant	low	normal	low	normal	normal
Mode	EB		BB	BB	BB		BB
Range							

- From the user profile, the system can infer the list of accredited universities based on user nationality.
- From identity, the set of features and context rules that relate to them can be defined.
- Since they are scholar students, the system inferences the following:
  - The mode of study will be full-time because they are not permitted to enroll in the part-time or online program.
  - Also, the system can infer entry requirement from student's qualification and predict the preferred qualification type.
  - The reasoning engine study the correlation between two features as the job here is teaching assistant, the preferred type will be master of research.
- In advance, during the design stage we stated that the context features are set to essential features (both having insignificant weights) and that the mode for English requirement is set to exact better. Table 4.8 shows the query for context features and context rule.

Table 4.8. Query for context features.

	Context Features			
Query	Qualification	Preferred Qualification	Mode of study	English Requirement
Values	Master	MRes	Full-time	6.5
Weights	CR	insignificant	CR	insignificant
Mode				EB
Range				

Result for Saudi Scholar Student:

- First, the system will apply context rule.Therefore:
  - S1 is excluded because S1 is a Ph.D. degree and the student does not qualify to study Ph.D.
  - S2 also is excluded because it is a part-time study and scholar students are not permitted to enroll in part-time programs.
  - S5 is excluded because it is bachelor's degree and the student has a bachelor degree. The system infers that it is searching for a master degree.
  - S7 is excluded because its major is not in the list of accredited university for Saudi scholar.
- Secondly, the rest of services S3, S4, S6, S8, S9, S10, and S11 are ranked based on context and non-context features as shown in table 4.9.

Table 4.9. Ranking the result based on context and non-context features for case1 for Saudi Scholar.

Record Number	Title	University	Category	Duration	Mode Of Study	Qualification	City	English Requirement	Ranking Score
S3	Advanced Computer Science	University of Birmingham	Computer Science	1	Full-time	MRes	Birmingham	6	0.809222288
S4	Advanced Computer Science	The University of Manchester	Computer Science	1	Full-time	MRes	Manchester	7	0.808363008
S8	Computing Science	Staffordshire	Computer information system	1	Full-time	MRes	Stoke	6	0.807722288
S10	Computing	Bedfordshire	Computer Science	1	Full-time	MRes	Luton	6	0.806722288
S6	Network Systems Engineering	Plymouth University	Computer Science	1	Full-time	MSc	Plymouth	6.5	0.4375756
S11	Computer Science	Bedfordshire	Computer Science	3	Full-time	MSc	Luton	6	0.378148688
S9	Networking and Data Communications	Kingston University	Computer Science	1	Full-time	MSc	Kingston	6.0	0.377082288

From table 4.9, we notice the following:

- If we analyze the first four services S3, S4, S8 and S10, we observe that the context features were met and the service were ranked based on context features English requirement and qualification.
  - Although S3 and S4 have, the same title value but algorithm preferred S3 to S4 because the semantic for English requirement is LB.
  - However, S4 has a greater value for English requirement 7 than the defined value in the user profile 6.5.
- Therefore, as in table 4.10, the algorithm classifies the services into two groups:

- The first group called services suit the context, includes all the services fulfilling the context information S3, S8 and S10. S4 is not included in the context group because its English requirement is greater than the user context.
- The second group, called services did not suit the context, includes the rest of the services S4, S6, S11 and S9.

Table 4.10. Classifying the results based on context features.

Record Number	Title	University	Category	Duration	Mode Of Study	Qualification	City	English Requirement	Ranking Score
Services suit the context									
S3	Advanced Computer Science	University of Birmingham	Computer Science	1	Full-time	MRes	Birmingham	6	0.809222288
S8	Computing Science	Staffordshire	Computer information system	1	Full-time	MRes	Stoke	6	0.807722288
S10	Computing	Bedfordshire	Computer Science	1	Full-time	MRes	Luton	6	0.806722288
Services did not suit the context									
S4	Advanced Computer Science	The University of Manchester	Computer Science	1	Full-time	MRes	Manchester	7	0.808363008
S6	Network Systems Engineering	Plymouth University	Computer Science	1	Full-time	MSc	Plymouth	6.5	0.4375756
S11	Computer Science	Bedfordshire	Computer Science	3	Full-time	MSc	Luton	6	0.378148688
S9	Networking and Data Communications	Kingston University	Computer Science	1	Full-time	MSc	Kingston	6.0	0.377082288

- We notice the algorithm preferred S3 over S8 and S10 because of the value of title feature is "Computer Science". The value of title feature is considered because its weight in the query has been set to significant and the weights of duration and category are set to normal.
- Thus, the services were ranked based on context features and taking into consideration the non-context features.
- If we look at the last three services S6, S11 and S9, we found the context feature qualification was not met.
    - Although S11 its title is "Computer Science" but the algorithm preferred S6 to S11 because of the value of English requirement feature was met exactly as defined in the user profile.
    - Therefore, S11 is preferred to S9 because of its title value.
- Thus, when context features were not met, the services are ranked based on non-context features.
- Finally, re-rank the results based on trustworthiness features.
    - In this step, we rank the services in each group based on trustworthiness features separately.

Table 4.11 Re-rank the result based on trustworthiness features for case 1 for Saudi Scholar.

Record Number	Verified features						Claimed feature	Ranking Score
	Rank	Professionally Recognized by:	Safety" Crime statistics"			Safety Total score	Student Satisfaction	
			Burglary	Robbery	Violent Crime			
Trustworthy- context Services								
S8	103	British Computer Society	6.25	0.92	23.89	31.06	4.04	0.00219008328417
S3	18	-	8.15	3.14	18.13	29.42	4.08	-0.007432314815
S10	110	-	9.99	1.25	18.08	29.32	4.04	-0.011789375
Trustworthy services did not suit the context								
S4	28	British Computer Society	13.62	3.08	22.43	39.13	4.02	0.00639082262
S6	90	-	5.10	0.42	21.80	27.32	4.03	-0.011543293524
S9	104	-	7.51	0.81	13.46	21.78	3.90	-0.01172909357
S11	110	-	9.99	1.25	18.08	29.32	3.89	-0.01179684201

From table 4.11 we noticed the following:

- S8 is ranked higher because it professionally recognized by British Computer Society and the student gave it a significant weight.
- Regarding safety feature, the user gave the highest weight to robbery feature and S8 has a lower robbery value than S3.
    - Although S10 has the lower robbery value than S3 but the algorithm prefers S3 over S10 since the weight for the whole safety feature has been set to normal to compare it with other features. S3 has a better rank and higher student satisfaction value than S10.
- Thus, when one sub-feature is met the weight for the whole composite feature is taken into consideration to compare with other non- composite or composite features.

Result for Qatari Scholar Student:

- First of all, the system will apply context rule and therefore:
    - S1 is excluded because S1 is a PhD degree and the student does not qualified to study PhD.
    - S2 also is excluded because it is a part time study and scholar students are not permitted to enroll in part-time program.
    - S5 is excluded because it is bachelor's degree and the student has a bachelor degree. The system infers that he is searching for a master degree.
    - S7, S8, S9, S10 and S11 excluded because they are not in the list of accredited university for Qatari scholar.
- Secondly, the rest of services S3, S4 and S6 are ranked based on context and non-context features as in table 4.12.

Table 4.12. Ranking the result based on context and non-context features for case1 for Qatari Scholar.

Record Number	Title	University	Category	Duration	Mode Of Study	Qualification	City	English Requirement	Ranking Score
Services suit the context									
S3	Advanced Computer Science	University of Birmingham	Computer Science	1	Full-time	MRes	Birmingham	6	0.809222288
Services did not suit the context									
S4	Advanced Computer Science	The University of Manchester	Computer Science	1	Full-time	MRes	Manchester	7	0.808363008
S6	Network Systems Engineering	Plymouth University	Computer Science	1	Full-time	MSc	Plymouth	6.5	0.4375756

- Finally, re rank the result based on trustworthiness features as in table 4.13.

Table 4.13. Re-rank the result based on trustworthiness features for Qatari Scholar for case 1.

Record Number	Verified features						Claimed feature	Ranking Score
	Rank	Professionally Recognized by:	Safety" Crime statistics"			Safety Total score	Student Satisfaction	
			Burglary	Robbery	Violent Crime			
Trustworthy- context Services								
S3	18	-	8.15	3.14	18.13	29.42	4.08	-0.007432314815
Trustworthy services did not suit the context								
S4	28	British Computer Society	13.62	3.08	22.43	39.13	4.02	0.00639082262
S6	90	-	5.10	0.42	21.80	27.32	4.03	-0.011543293524

From the above example, we demonstrated how users submit the same request and get different results based on their context. In the real scenario, the benefit would be more obvious, when there are more records returned.

Case 2: When *identity* has been set to *"scholar student"* and the *priority* has been set to *"Trustworthy services"* for the entire query.

In this example, we consider the same user profile shown in table 4.4 and the same query for not-context and trustworthiness shown in tables 4.6 and 4.7 with priority option set to "Trustworthy services". The context information does not change and is the one shown in table 4.8. After applying the context rule as in Case 1 for Saudi scholar, the result is ranked based on trustworthiness features as shown in table 4.14 below. The algorithm skips the step of ranking the result based on context and non-context features because the priority has been set to "Trustworthy services".

Table 4.14. Rank the result based on trustworthiness features for case 2.

Record Number	Verified features						Claimed feature	Ranking Scores
	Rank	Professionally Recognized by:	Safety" Crime statistics"			Safety Total score	Student Satisfaction	
			Burglary	Robbery	Violent Crime			
Trustworthy- Services								
S4	28	British Computer Society	13.62	3.08	22.43	39.13	4.02	0.00639082262
S8	103	British Computer Society	6.25	0.92	23.89	31.06	4.04	0.00219008328417
S3	18	-	8.15	3.14	18.13	29.42	4.08	-0.007432314815
S6	90	-	5.10	0.42	21.80	27.32	4.03	-0.011543293524
S9	104	-	7.51	0.81	13.46	21.78	3.90	-0.01172909357
S10	110	-	9.99	1.25	18.08	29.32	4.04	-0.011789375
S11	110	-	9.99	1.25	18.08	29.32	3.89	-0.01179684201

Notice the change from the results of Case 1 in table 4.9 to table 4.14:

- The main reason is because S4 and S8 are closer to the user request from the perspective of trustworthiness features.
    - S4 did not suit the user context in terms of English requirement feature since it has a greater value than the query value.
- Thus, when "Trustworthy services" priority is on, the algorithm ranks the services that met the context rule based on trustworthiness features and it is labeled with *"Trustworthy- Services"* label.
    - The trustworthiness features take precedence on context features.

Case 3: When the *identity* is *"self-funded student"* and the *priority* is set to *"Trustworthy context services"*.

In the following example, the qualification is set as a key feature. The fee and duration are set to best mode. The query for non-context features is shown in table 4.15. The query for trustworthiness is shown in table 4.16. The rank is set to exact mode, safety as a whole feature and student satisfaction are set to best mode.

The Rank here is applied to S1, S2, S3, S5, S6, S7, S8 and S9 because they have an available value for fee feature.

Remark: the algorithm has the ability to deal with missing features and null values but for the sake of simplicity, we omit the services that has a null overseas fee.

Table 4.15. Query for non-context features for case 3.

Non-Context Features						
Query	Title	Category	Duration	Qualification	Mode of study	Fee
Values	Computer Science	Computer Science	2	MRes	Full-time	£12,500
Weights	Low	Low	Normal	Normal	Normal	Significant
Mode	EB	EB	BB	EB	EB	BB
Range						
Key				√		
priority	Trustworthy services , √ Trustworthy context services, Context services					

Table 4.16 Query for Trustworthiness features for case 3.

Trustworthiness feature							
	Verified Features						Claims Features
Query	Rank	Professionally Recognized by	Burglary	Robbery	Violent Crime	Safety	Student Satisfaction
Values	20	British Computer Society	-	-	-	25	4
Weights	normal	significant				normal	normal
Mode	EB					BB	BB
Range							

- From the user profile defined in table 4.17, the system can infer the list of accredited universities based on user nationality.

- And from identity, it can define the set of features and context rule that relate to the user.
    - Since the student is a self-funded student, the mode of study and fee appears.
    - Also, the system can infer entry requirement from student's qualification and predict the qualification type.
- Table 4.18 shows the query for context features and context rule. The context features are set to essential features and has an insignificant weight and the mode for English requirement is set to exact better.

Table 4.17. User profile for Case 3.

User profile	
Name	Ahmad
Nationality	Saudi
Identity	KASP Scholar. √ Self-funded Student. Scholar employment.
Qualification	Bachelor
Job	-
English Requirement	6.5

Table 4.18. Query for context features for case3.

	Context Features	
Query	Qualification	English Requirement
Values	Master	6.5
Weights	CR	insignificant
Mode		EB
Range		
Essential		√

- Initially, the system applies context rule. Therefore:
    - S1 is excluded because S1 is a Ph.D. degree and the student does not qualify to study Ph.D.

- S5 is excluded because it is bachelor's degree, the student has a bachelor degree and he is searching for a master degree.
- S7 is excluded because its major is not in the list of accredited university for Saudi scholar.

- Then the system applies the service provider context for fee feature. Since the student nationality is Saudi, the fee for overseas students is selected.
- Secondly, the rest of services S1, S2, S3, S6 and S9 are ranked based on context and non-context features.

Table 4.19. Ranking the result based on context and non-context features for case3.

Rank	Record	Title	University	Category	Duration	Mode Of Study	Qualification	City	English Requirem	Fee	Ranking Score
										Overseas students	
1	S6	Network Systems Engineering	Plymouth University	Computer Science	1	Full-time	MSc	Plymouth	6.5	£13,250	8.95406956
2	S8	Computing Science	Staffordshire	Computer information system	1	Full-time	MRes	Stoke	6	£11,500	7.69952033
3	S3	Advanced Computer Science	University of Birmingham	Computer Science	1	Full-time	MRes	Birmingham	6	£17,960	7.69624495
4	S9	Networking and Data Communications	Kingston University	Computer Science	1	Full-time	MSc	Kingston	6.0	£12,500	7.69578504
5	S2	Advanced Computer Science	University of Birmingham	Computer Science	2	Part-time	MRes	Birmingham	6	£19,200	7.690859855

From Table 4.19 we noticed the following:

- S6 ranked first because of the English requirement met exactly the student context but it did not fulfill the key feature MRes qualification type.
- The second service S8 has the lowest fee suited the user context and it is closer to

the user request. Table 4.20 shows the change when we applied the key feature.

Table 4.20 Re-ranking the result based on key feature for case3.

Rank	Record Number	Title	University	Category	Duration	Mode Of Study	Qualification	City	English Requirement	Fee Overseas students	Ranking Score
Services Suit the context and key											
2	S8	Computing Science	Staffordshire	Computer information system	1	Full-time	MRes	Stoke	6	£11,500	7.69952033
3	S3	Advanced Computer Science	University of Birmingham	Computer Science	1	Full-time	MRes	Birmingham	6	£17,960	7.69624495
5	S2	Advanced Computer Science	University of Birmingham	Computer Science	2	Part-time	MRes	Birmingham	6	£19,200	7.690859855
Different services suit your context											
1	S6	Network Systems Engineering	Plymouth University	Computer Science	1	Full-time	MSc	Plymouth	6.5	£13,250	8.95406956
4	S9	Networking and Data Communications	Kingston University	Computer Science	1	Full-time	MSc	Kingston	6.0	£12,500	7.69578504

- By using key feature, we notice the following:
  - The algorithm excluded all the services that do not satisfy the context feature and key features from the top.
  - In table 4.20, S8, S3 and S2 satisfied the user context and key feature. Therefore, to identify them, they are grouped in one list with a label "result suit the context and key".
  - Since S6 did not satisfy the key feature. Therefore, the algorithm removed it from the first group and ranked it as a first service in the second group that label with different result suit your context.
    - Thus, we do not lose the original ranking score.
  - By using labels we help the student to make a better decision and faster.

- Finally, Re-rank the services in each group based on trustworthiness overall scores as in table 4.21.

Table 4.21 Re-rank the result based on trustworthiness features for case 3.

Record Number	Verified features						Claimed feature	Ranking Score
	Rank	Professionally Recognized by:	Safety" Crime statistics"			Safety	Student Satisfaction	
			Burglary	Robbery	Violent Crime			
Trustworthy services suit the context and key								
S3	18	-	8.15	3.14	18.13	29.42	4.08	0.69508445
S2	18	-	8.15	3.14	18.13	29.42	4.08	0.69508445
S8	103	British Computer Society	6.25	0.92	23.89	31.06	4.04	0.00398959497
Different trustworthy services suit the context								
S9	104	-	7.51	0.81	13.46	21.78	3.90	-0.00853927559
S6	90	-	5.10	0.42	21.80	27.32	4.03	-0.00907452885

From table 4.21, we noticed the following:

- All the services in the first group that are identified by "Trustworthy services suit the context and key" are the same services in Table 20 that identified by "Services Suit the context and key" but here are ranked based on trustworthiness features.
- Relative to the requested query in table 4.16 we noticed that S3 and S2 raised up and they had the same trustworthiness score = 0.69508445. However, the priority for ranking higher has been given to the service that has a higher score based on context and non-context features.
    - Back to table 4.20 S3 = 7.69624495 and S2 = 7.690859855.
        - Thus, S3 ranked higher than S2.

- In this example, we notice how trustworthy-context services priority affected the ranking and how this priority sometimes comes at the cost of others.
  - That is because student set a fee as a significant feature but he gave the priority to trustworthy-context services.
  - S3 and S2 are most trustworthy services but they have a higher fee than S8 and higher than the value that set in the query.
- All the services in the second list identified by "Different trustworthy services suit the context" are the same services of table 4.20 identified by "Different services suit your context" but here are ranked based on trustworthiness features.
  - Therefore, we found here all the services in this list suit the user context and trustworthy services but it did not satisfy the key features.
  - We noticed that S9 becomes the first service in this group because S9 is more trustworthy than S6 and it has a lower fee than S6.
  - This means that not always priority come at the cost of other features, sometimes it helps to place each service in its specific place relative to the requested features.

4.6 Conclusion:

In this chapter, we have provided an evaluation for the proposed formal framework by applying it in the field of King Abdul Allah Scholarship program and Supreme Education Council of Qatar. The case study illustrated the success of our proposed architecture. We found that the proposed framework performed excellently in all cases.

Applying the formal specification to the case study showed the power of the formalism and the ability to be applied and used in diverse range of applications and it can be translated to many programming languages.

Chapter 5

Conclusion and Future Work

Chapter V

Conclusion and Future Work

5.1 Research Summary

In this thesis, we have proposed a generic architecture for matching and ranking trustworthy context-dependent services. We provided a formal definition for matching and ranking services that are trustworthy and context-dependent using logic and set theory.

We mainly extended X-algorithm from semantic-based and user-centric[2] to context-centric. Then we extended it to rank composite trustworthy features.

X-algorithm[2] is aiming for a consistent ranking that gives the same answer if given the same input. When we incorporated it with context awareness concept it gives different results based on user context even when the user request are the same.

Through context awareness, context information can be used to expand the query to retrieve services that are more relevant to the users. Users do not explicitly specify this context information, so we reduced the number of features in the query. The context rule can be used by the service provider to control the willingness and eligibility of providing services to certain users.

Trustworthiness is a composite feature. Composite features can be represented as a hierarchical model. Therefore, we suggested to model and rank them as a tree technique. These features split into verified features and claimed features. Verified features are related to verified trustworthiness features or rating from another trusted organization. Claimed features related trustworthiness features claimed by the service provider itself or to user feedback rating of services. Finally, the verified and claimed trustworthiness features aggregated to compute the overall trustworthiness of services.

Since in real life, some service is better in trustworthiness features, other is better in context features and neither is better than other overall. We let the user decide the priority to rank services. In addition, we are using explanation techniques to categorize services based on context information and trustworthiness. By providing labels to explain to the user why these services are presented to them.

A full implementation of a case study in the KASP and Supreme Education Council of Qatar is presented. We conducted different combinations of queries with different priority options and identities. The produced results were satisfactory and expected.

This implementation shows and proves the success of the proposed architecture. It was the first work to automate and facilitate the process of searching for the right courses based on context information and trustworthiness.

To the best of our knowledge, there is no prior work achieving all these results.

5.2 Conclusion of Results and Findings

1. Applying context rule will dramatically reduce the irrelevant services and save a significant amount of computation resources done in preparation, multiplication and sorting phases in X-algorithm [2].
2. Applying SPC will help to personalize the content based on requester context.
3. Defining Identity will help to define the related features that the requester has with regard to an application.
4. The services are ranked based on service request context features and it's semantic. Then, it takes into consideration the non-context features.
5. In case we have more than one service request context feature, domain expert use regular weight to tradeoff between them.
6. When service request context features were not met, the services are ranked based on non-context features.
7. In the composite feature, when one sub-feature is met, the weight for the whole composite feature is taken into consideration to compare with other on-composite or composite features.
8. All the services that fulfill the context requirements are grouped and ranked in one list with a label "services suit the context" to identify them. Then re-rank this list based on trustworthiness features.
9. When "Trustworthy services" priority is on, the algorithm ranks the services that met the context rule based on trustworthiness features. The trustworthiness features take precedence on context features.

10. By using the key feature, we exclude and categorize all the services that do not satisfy the context feature and key features. At the same time, we do not lose the original ranking for these services.

5.3 Future Works

Our near future work will be focused on following aspects:

Automated services' features extraction: Currently, extracting features and assigning sematic to these features is done by domain experts. For instance, service descriptions are located on the web. The future direction is to automatically extract these descriptions then convert them into distinct features and assign semantic to them.

Automated context information acquisition: Context information is dynamic and after a period of time is changed. In addition, it is extracted from many different sources from the web. For example, the list of accredited university for Saudi scholar is listed on the MoE. This list is updated periodically. The question that we need to investigate is can we automatically detect these changes on these context information and update the knowledge base automatically.

Visualizing tool: Incorporating the visualizing technique in [56] to visualize multi features ranking list. It would be better to users for helping them to keep track the changes between these three independent lists. Instead of state the query priority (context services, trustworthy context services or trustworthiness services) for every single query.

Mobile application or pervasive computing: Applying the proposed architecture for devices that have limited resources.

Building user profile automatically: The user profile in this work is built manually by the user. The future work focuses on building the profile automatically and reducing the user effort.

LIST OF REFERENCES

[1] T. Erl, *SOA: Principles of Service Design*, 1st edition. Upper Saddle River, NJ: Prentice Hall, 2007.

[2] A. Alsaig, "Semantic-Based, Multi-Featured Ranking Algorithm for Services in Service-Oriented Computing," masters, Concordia University, 2013.

[3] M. Papazoglou, *Web Services: Principles and Technology*, 1st ed. Prentice Hall, 2008.

[4] N. Ibrahim, M. Mohammad, and V. Alagar, "Adaptable Discovery and Ranking of Context-Dependent Services," in *Services Computing Conference (APSCC), 2011 IEEE Asia-Pacific*, 2011, pp. 223–230.

[5] H. Lu, "Semantic Web Services Discovery and Ranking," in *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, Washington, DC, USA, 2005, pp. 157–160.

[6] U. Bellur and H. Vadodaria, "Web Service Ranking Using Semantic Profile Information," in *IEEE International Conference on Web Services, 2009. ICWS 2009*, 2009, pp. 872–879.

[7] N. I. Ibrahim, "Specification, Composition and Provision of Trustworthy Context-dependent Services," Ph.D, Concordia University, 2012.

[8] N. Ibrahim, V. Alagar, and M. Mohammad, "A Context-dependent Service Model.," *EAI Endorsed Trans Context-Aware Syst Appl*, vol. 2, p. e3, 2014.

[9] G. D. Abowd and A. K. Dey, "Towards a Better Understanding of Context and Context-Awareness," in *Handheld and Ubiquitous Computing*, H.-W. Gellersen, Ed. Springer Berlin Heidelberg, 1999, pp. 304–307.

[10] A. Schmidt, M. Beigl, and H. Gellersen, “There is more to Context than Location,” *Comput. Graph.*, vol. 23, pp. 893–901, 1998.

[11] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, “A survey of context modelling and reasoning techniques,” *Pervasive Mob. Comput.*, vol. 6, no. 2, pp. 161–180, Apr. 2010.

[12] K. Wan, “Lucx: Lucid Enriched with Context,” Ph.D, Concordia University, Montreal, P.Q., Canada, 2006.

[13] W. Rong and K. Liu, “A Survey of Context Aware Web Service Discovery: From User’s Perspective,” in *2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE)*, 2010, pp. 15–22.

[14] B. Benatallah, M.-S. Hacid, C. Rey, and F. Toumani, “Request Rewriting-Based Web Service Discovery,” in *The Semantic Web - ISWC 2003*, D. Fensel, K. Sycara, and J. Mylopoulos, Eds. Springer Berlin Heidelberg, 2003, pp. 242–257.

[15] A. V. Paliwal, N. R. Adam, and C. Bornhovd, “Web Service Discovery: Adding Semantics through Service Request Expansion and Latent Semantic Indexing,” in *IEEE International Conference on Services Computing, 2007. SCC 2007*, 2007, pp. 106–113.

[16] C. Lee and S. Helal, “Context attributes: an approach to enable context-awareness for service discovery,” in *2003 Symposium on Applications and the Internet, 2003. Proceedings*, 2003, pp. 22–30.

[17] H. Tianfield, “Context-aware service discovery in pervasive environments: A survey,” *Int. Trans. Syst. Sci. Appl.*, vol. 7, no. 3/4, pp. 314–338, 2011.

[18] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li, “Context-aware Query Suggestion by Mining Click-through and Session Data,” in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2008, pp. 875–883.

[19] N. Kokash, A. Birukou, and V. D’Andrea, “Web Service Discovery Based on Past User Experience,” in *Business Information Systems*, W. Abramowicz, Ed. Springer Berlin Heidelberg, 2007, pp. 95–107.

[20] P. R. Bailey, S. De La CHICA, A. Crook, N. B. DANDEKAR, D. R. Schwartz, W. D. Ramsey, and N. Craswell, *Ranking contextual signals for search personalization*. Google Patents, 2014.

[21] R. Zhang, K. Zettsu, Y. Kidawara, and Y. Kiyoki, “Web Service Ranking Based on Context,” in *2012 Second International Conference on Cloud and Green Computing (CGC)*, 2012, pp. 375–382.

[22] I. Sommerville, *Software Engineering: (Update) (8th Edition) (International Computer Science)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006.

[23] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing,” *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 1, pp. 11–33, Jan. 2004.

[24] M. Mohammad and V. Alagar, “A Formal Approach for the Specification and Verification of Trustworthy Component-based Systems,” *J Syst Softw*, vol. 84, no. 1, pp. 77–104, Jan. 2011.

[25] K. Ganesan and C. Zhai, “Opinion-based entity ranking,” *Inf Retr*, vol. 15, no. 2, pp. 116–150, 2012.

[26] S. Bashir, W. Afzal, and A. R. Baig, “Opinion-Based Entity Ranking using learning to rank,” *Appl. Soft Comput.*, vol. 38, pp. 151–163, Jan. 2016.

[27] H. Wang, Y. Lu, and C. Zhai, “Latent aspect rating analysis on review text data: a rating regression approach,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, DC, USA, 2010, pp. 783–792.

[28] E. Al-Masri and Q. H. Mahmoud, “QoS-based Discovery and Ranking of Web Services,” in *Proceedings of 16th International Conference on Computer Communications and Networks, 2007. ICCCN 2007*, 2007, pp. 529–534.

[29] J. Yan and J. Piao, “Towards QoS-Based Web Services Discovery,” in *Service-Oriented Computing – ICSOC 2008 Workshops*, G. Feuerlicht and W. Lamersdorf, Eds. Springer Berlin Heidelberg, 2008, pp. 200–210.

[30] K. K. Fletcher, X. F. Liu, and M. Tang, "Elastic Personalized Nonfunctional Attribute Preference and Trade-off Based Service Selection," *ACM Trans Web*, vol. 9, no. 1, pp. 1:1–1:26, Jan. 2015.

[31] R. Mirmotalebi, C. Ding, and C.-H. Chi, "Modeling User's Non-functional Preferences for Personalized Service Ranking," in *Service-Oriented Computing*, Springer, 2012, pp. 359–373.

[32] P. Pu, L. Chen, and R. Hu, "Evaluating recommender systems from the user's perspective: survey of the state of the art," *User Model. User-Adapt. Interact.*, vol. 22, no. 4–5, pp. 317–355, Mar. 2012.

[33] R. L. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge England ; New York, NY, USA: Cambridge University Press, 1993.

[34] M. A. S. N. Nunes, "Recommender systems based on personality traits," Université Montpellier II-Sciences et Techniques du Languedoc, 2008.

[35] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach," *ACM Trans Inf Syst*, vol. 23, no. 1, pp. 103–145, Jan. 2005.

[36] G. Adomavicius and A. Tuzhilin, "Context-Aware Recommender Systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer US, 2011, pp. 217–253.

[37] M. E. Matthias Braunhofer, "Techniques for cold-starting context-aware mobile recommender systems for tourism," *Intell. Artif.*, vol. 8, pp. 129–143, 2014.

[38] A. K. Dey, G. D. Abowd, and D. Salber, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-aware Applications," *Hum-Comput Interact*, vol. 16, no. 2, pp. 97–166, Dec. 2001.

[39] N. P. de Koch, "Software engineering for adaptive hypermedia systems," Ph.DThesis, Verlag Uni-Druck, Munich, 2001.

[40] F. Qiu and J. Cho, “Automatic Identification of User Interest for Personalized Search,” in *Proceedings of the 15th International Conference on World Wide Web*, New York, NY, USA, 2006, pp. 727–736.

[41] J. Berri and R. Benlamri, “Context-Aware Mobile Search Engine,” *Gener. Search Engines Adv. Models Inf. Retr. Adv. Models Inf. Retr.*, p. 371, 2012.

[42] W. Mansoor, M. Khedr, D. Benslimane, Z. Maamar, M. Hauswirth, K. Aberer, T. Chaari, F. Laforest, and A. Celentano, “Adaptation in context-aware pervasive information systems: the SECAS project,” *Int. J. Pervasive Comput. Commun.*, vol. 3, no. 4, pp. 400–425, 2008.

[43] R. Meunier, “The pipes and filters architecture,” in *Pattern languages of program design*, 1995, pp. 427–440.

[44] “About FindAMasters.com,” *www.FindAMasters.com*. [Online]. Available: http://www.findamasters.com/about/about-findamasters.aspx. [Accessed: 06-Dec-2015].

[45] “Search postgraduate courses | Prospects.ac.uk.” [Online]. Available: https://www.prospects.ac.uk/postgraduate-courses. [Accessed: 23-Mar-2016].

[46] “CUG Course Chooser.” [Online]. Available: http://www.thecompleteuniversityguide.co.uk/courses/search. [Accessed: 08-Dec-2015].

[47] “Business Courses, Degrees, Masters, MBAs QS Course Finder.” [Online]. Available: http://www.qscourses.com/. [Accessed: 23-Mar-2016].

[48] “U-Multirank | Universities compared. Your way.” [Online]. Available: http://www.umultirank.org/#!/about?trackType=home&sightMode=undefined&section=undefined. [Accessed: 23-Mar-2016].

[49] “Saudi Cultural bureau in Canada.” [Online]. Available: http://www.saudibureau.org/en/inside.php?ID=16#. [Accessed: 06-Dec-2015].

[50] “سفير الجامعات - بحث عن جامعة.” [Online]. Available: https://ru.moe.gov.sa/Search. [Accessed: 06-Dec-2015].

[51] "Police.uk." [Online]. Available: https://www.police.uk/. [Accessed: 09-Dec-2015].

[52] "Crime in University Towns and Cities." [Online]. Available: http://www.thecompleteuniversityguide.co.uk/crime-in-university-towns-and-cities/. [Accessed: 09-Dec-2015].

[53] "The National Student Survey 2016." [Online]. Available: http://www.thestudentsurvey.com/. [Accessed: 09-Dec-2015].

[54] "Top UK University League Tables and Rankings 2016." [Online]. Available: http://www.thecompleteuniversityguide.co.uk/league-tables/rankings. [Accessed: 08-Dec-2015].

[55] "Ministry of Education and Higher Education - Supported Universities." [Online]. Available: http://www.sec.gov.qa/En/ServicesCenter/Pages/Supported-Universities.aspx. [Accessed: 11-Jun-2015].

[56] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit, "Lineup: Visual analysis of multi-attribute rankings," *Vis. Comput. Graph. IEEE Trans. On*, vol. 19, no. 12, pp. 2277–2286, 2013.

المنهج الدقيق للمطابقة والتصنيف للخدمات متعددة الخصائص الموثوقة والمعتمدة على السياق

أفنان بوبكر سعيد باوزير

المستخلص

الخدمات الموجهة للحوسبة (SOC) تشكل مستقبل الحوسبة الموزعة Distributed Computing ومستقبل تطوير تطبيقات المشاريع العملاقة Enterprise Applications. ونظرا لعدد الخدمات المتزايد توجد صعوبة على طالبي الخدمة لاختيار خدمة موثوقة تناسب سياق طالب الخدمة ومقدمها. لذلك فان هناك حاجة لعملية مطابقة وترتيب الخدمات والتي تأخذ بعين الاعتبار خصائص الخدمة، ويجب أن تأخذ في الاعتبار السياقات الخاصة بطالب الخدمة ومنفذ الخدمة لتحسين ترتيب الخدمات وللحصول على أفضل النتائج. وكذلك يجب ان تتضمن عمليات المطابقة والترتيب متطلبات الثقة لتقديم خدمات موثوقة بناء على تفضيلات الطالب. هذه الرسالة وفرت منهج دقيق ومعايير لتمثيل وترتيب خصائص السياق، الخصائص ليست مرتبطة بالسياق وخصائص الثقة المركبة. في هذه الرسالة اقترحنا هيكل مبني على أساس المنهج الدقيق للمطابقة والترتيب للخدمات الغنية والتي تحتوي على الموثوقية والسياق. تم تطبيق دراسة الحالة على بيانات حقيقيه على برنامج ابتعاث الملك عبد الله (KASP) لنوضح كيفية ترتيب الجامعات الموثوقية والمعتمدة على السياق لتقييم المنهج المقترح.