A FORMAL APPROACH FOR MATCHING AND RANKING CONTEXT-DEPENDENT SERVICES

[1]AFNAN BAWAZIR, [2]WADEE ALHALABI, [3]MUBARAK MOHAMMAD

1,2Dept. of Computer Science, King Abdul Aziz University, Jeddah, KSA
3Dept. of Computer Science and Software Eng. Concordia University, Montreal, QC, Canada

Abstract— Service-oriented computing (SOC) is shaping the future of distributed computing and enterprise application development. In SOC, service provider and requester are the main interacting elements. Due to the vast volume of services, it is overwhelming for requesters to select the 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. This work studies the state of the art of the ranking process in service-oriented architecture, and context awareness. Also, it investigates recommender system. Besides, we formalize the criteria to model and rank context features and non-context features using logic and set theory . This work proposes a formal architecture for matching and ranking context-dependent services. We present a real-world case study on King Abdullah Scholarship Program (KASP) and show how to rank context universities/programs and to evaluate the proposed architecture.

Keywords— Service ranking; Context-aware; Archtecture.

I. 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 overwhelming 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.

In real life if there are prospective postgraduate students searching for universities to apply to. They will find available universities around the world 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. Websites such as find master [3], prospects [4], course guide chooser [5], QS course finder [6] and U-Multirank [7] help students in searching and finding their universities and programs .although , these 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. The problem is the students need 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. Different students from different countries have different context information. In real life, there is no choice that meets all the student preferences. Consequently, there is a need to develop an efficient tool for matching and ranking universities based on student criteria and taking into account the context information.

This problem can be encountered in many other domains such as jobs, medical domain, flights, real estate, and hotels. This motivated us to design a formal method for ranking context features and non-context features according to user preferences.

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.

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 context depended services.

The major contributions of our work are following.

1. Review the existing ranking algorithm and issues related to context awareness concept and recommender systems as discussed in section II.

2. Provide a formal specification for matching and ranking multi-featured context-dependent services, as discussed in section III.
3. Matching and ranking context features, as discussed in section IV.
4. Propose an architecture for matching and ranking multi-featured context-dependent services, as discussed in section VI.
5. Provide a case study on real world application that illustrates the success of the proposed architecture, as discussed in section VII.

In section VIII, we conclude our paper and list several future direction.

II. LITERATURE REVIEW

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

Service-oriented Architecture

Service discovery process in SOA is the process of finding services in service registry that meet the requirements of a service requester [3]. Typically, service discovery includes service query, service matching, and service ranking. Service query is the main input in discovery process. Service requesters define their requirements as service queries.

Matching process 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 based on context information.

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 contains the following elements: 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.

1. **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.
2. **Multiplication phase:** it takes the PM and multiplies it with weight vector to produce an unsorted rank (UR) vector.
3. **Sorting phase:** it takes the UR vector and apply quick sort algorithm with a descending order to produce a sorted rank list.

Context and Context awareness

Abowd and Dey [8] stated that there are many different types of context, such as location, identities of nearby people, objects and changes to these objects, and time. They give a more formal definition of context [8]: "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 advantage of context information and to provide adaptable services. We follow the notion that "there is more to context than location" [9]. In general, as defined in [8] "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 [10]. However, in [11] Wan gave a formal representation of context, which 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"[12], context is divided into context information and context rule. Context information is moulded, as defined in [11] and 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 (the process of enhancing the service request), result optimization (the process of solving the problem of some web services that are ranked lower in the retrieved set but are more relevant to the user's needs), and personalization (providing a different result set with the same service request). There have been many studies on service discovery in a pervasive environment [14]. Context attributes act as filters, is requester-related does not take into consideration the provider context and the major limitation is that they fail to rank the suitable services. To clarify, assume, as an example, we have two services that matched the user context that considered services nearby the user. The work failed to determine which one was more suitable and nearer to the user. Authors in [15] and [16] provided a quality relevance result by modeling context as historical search data. The drawbacks of these methods are: (1) they rely on past context, not considering the current context; (2) the problem of shared devices or accounts is raised, thereby it is hard to personalize the result; (3) for new users, the cold start problem is raised.

Work in [17] 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.

Featured-based Recommender systems

When we talk about ranking services, recommendation systems are widely taken into consideration because they 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. Feature-based systems permit users to explicitly state their preferences on specific item features. This type of recommendation is useful when users are searching for services or products that carry many financial risks. Four types of feature based recommender systems:

1) case based-systems,
2) utility based-systems,
3) knowledge based-systems and
4) critique based-systems.

Multi-Attributes utility theory (MAUT) [18] is usually employed in these systems to measure the item's utility. From the viewpoint of recommender systems, the X-algorithm [2] shows a promising result for modeling preferences with semantics and produces a fair recommendation list.

The major limitation of the X-algorithm [2] is that it is a "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, the X-algorithm has some user request constraints. 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 and knowledge. 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, and take into consideration trustworthiness features to provide reliable services.

III. FORMALIZING

In this section, we formalize the criteria to distinguish context features from non-context features, 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 ∈ 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.

A. Service Model

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, ……,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, ……,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 classified into two groups: 1) the ranked features group that are included as a criterion in ranking. RF = {x: x is a ranking feature}. 2) 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. The set of features that are relevant for the application are to be determined by the domain expert.

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 and 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}.

The RF are categorized into two parts: context features CF and non-context features NCF. Following we will describe in more details each category.

Definition 3: RF = {x ∈ RF| x ∈ CF V x ∈ NCF }.

Context Information: Context is any type of information used to characterize an object or situation [19] which helps to obtain the desired and most relevant services according to user requests.

Definition 4: Context information CI is formally specified, as defined in [11], Let τ:DIM → 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 ∈ I. We write CI as an aggregation of ordered pairs (Xj , vj), where Xj ∈ DIM, and vj ∈ τ (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 5: 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 service requester (SR) associated with the query. The value is given only if the constraint evaluates to true.

Definition 6: the set of SPC features can be defined as SPC = {p∈ RF |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 fee feature has two values one for home student and one for an international student, 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 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 that satisfied some of the following condition is called an SRC feature:

1. **Tangible:** attributes that can be sensed implicitly form another context and quantified as in the GPS system, which senses implicitly the location of a user on the map.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. 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 paper. Privacy(x)
4. **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. Compute(x1,x2)
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. Negotiable_skill(x).
6. 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}.
7. Infer it from data "user profile" or environment. For example, the system can infer the users' age from their birth of dates. Infer(x).
8. Infer it by correlating between features: sometimes we need to study the correlation between features to infer the service request context value. In order to build a context-depended application, there is additional work. Because preferences are also context dependent, the system has to study the correlations between features. For example, the preferred choice for students employed as university teaching assistants is a Master of Research, but for non-academic students, a Master of Teaching might be preferred. That is the correlation between job and type of qualification used to infer the context value preferences.

Definition 7: Correlation between features can be formalized as: let D be a set of dimensions = {d1,d2,….dn}, let Ta be a set of tags of value {t1,t2,….tn} and let P be a set of inferred values {p1,p2…pn}.

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

$d2(y) \rightarrow t2(y)$ ┤ $d1(x)$ Ë $d2(y) \rightarrow p1(z)$.

Definition 8: SRC ={x ∈ RF |x: (sense(x) V Infer(x) V Dynamic(x) Ë ¬ privacy(x)) V x ∈ CD }

Definition 9: context rules (CR) are 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 ∈ C defined over the set of SRC. We write CR = {(SRC, XSRC)| XSRC ∈ 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.

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

Definition 13: NCF= {ncf ∈ RF| ncf ∉ SRC }.

IV. MATCHING AND RANKING CONTEXT INFORMATION

Incorporating context-awareness in service ranking can greatly enhance the relevance of service ranking results and reduce the distraction to the service requester. 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 discuss how to treat and incorporate them into ranking process.

A. 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 [12], the author expressed context rules as a logical expression statement. These rules are defined by the service requester context features. Context rules act as filters for all the services that do not suit the requester context. Therefore, this approach dramatically reduces the irrelevant services.

B. Service provider context features

According to [20], 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.

C. Service requester context features

Generally, 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 the English score for a scholar student is 6.5, the top results should only contain universities that accept students' scores equal to or lower than 6.5. A student cannot be satisfied by being provided with the top results of any universities that require 7 or 7.5. The English requirement can be identified as a must-have requirement. It is important to note that the system should be able to anticipate the threshold value and its direction as lower than or higher than for the service requester context feature. Therefore, we need to understand the context features in terms of data types in more detail. Based on the service requester context feature and its semantic, the system anticipates the threshold value and its direction. For instance, the 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 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 a distance, such as in a GPS, the system designer needs to define the threshold value as a certain value so that all the services less than or greater than the threshold value are considered as context services.

Non-context features are nice-to-have features where service requesters' preferences are flexible, which may or may not be exactly satisfied. The service requester accepts services that closely fulfill their non-context features and their semantics. 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.

Key features

A key feature is a feature option that can be set for each feature independently to solve the context dominance problem. When the top-k results in a large data set typically include some services that are satisfied (service request context requirement) but do not fulfill the other "non-context" features, therefore they cannot meet the requester requirement, so end up with top services that people do not like. Key features helps the 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".

V. DISCRIPTION OF THE PROPOSED ARCHTECTURE

Fig 1 shows our proposed architecture for a formal approach for matching and ranking context-depended services framework. The rest of this section presents a detailed information on the main elements shown in the fig 2.

Service Registry : The service registry 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 paper.

User request: user request has constraints for the format of the query : all possible features should be available in a query definition, the query include 2 parts context features and non-context features associated with their weights to model the level of importance for each feature. A higher weight indicates a higher important. 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.

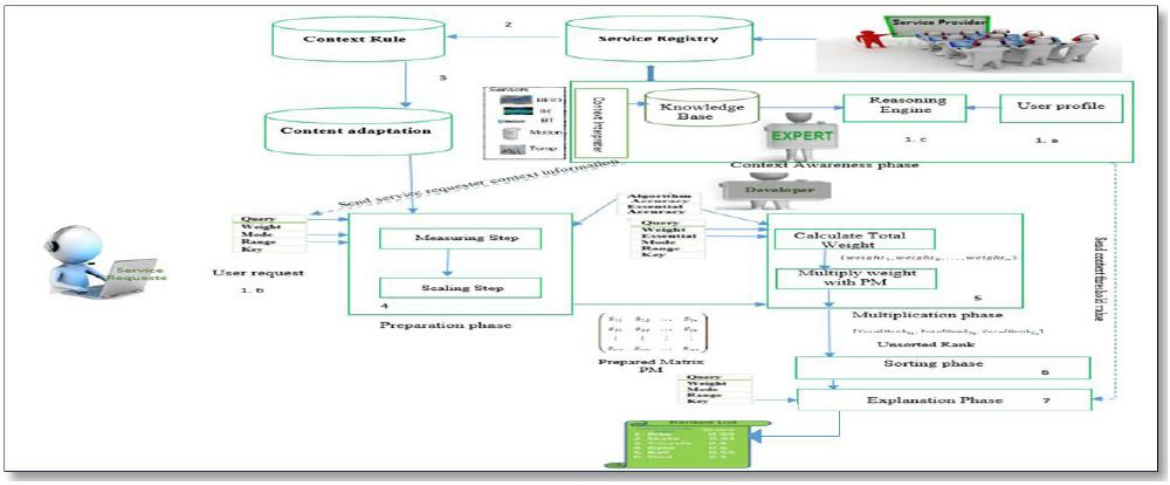


Figure 1. Framework of formal approatch for matching and ranking context depended services.

Definition 14: a query is defined as q=(f̂, ç, Ḱ, Ess,AlgAcc, ụ) where f̂ is a non-context query , ç is a context query. Ḱ is a key feature. Ess is Essential Accuracyand AlgAcc is Algorithm Accuracy that are determined by the provider. ụ is the user profile.

Definition 15: The non-context query is defined as f̂ = (f̂f ,Ṁ, ,ẅ) where f̂f is non-context features requrment. Ṁ: (x ∈ {best mode, Exact mode}) (y∈ {f̂f })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∈ { f̂}) is a function that assign weights to the feature of the query.

Definition 16: The context query is defined as ç = (,Ṁ, ,ẅ, Ê) where is a service request context features requirement . Ṁ and ẅ are defined by a domain expert as in the not -context query. Ê is the Essential weight for SRC requrments.

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 work.

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 [8] [21] 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 SR. 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:

1) Perform a pairwise comparison between the attributes of query and available services associated with its semantics for each feature.
2) Consider the following parts of the user request: query, mode, range, and algorithm accuracy.
3) Normalize the results of the pairwise comparison to a common scale.
4) 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_throw 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: This phase is responsible for generating good explanations for the ranked results. For example, services can be grouped by their context information. Each group is labeled with a label explaining the characteristics of the services. 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 [22] 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." The explanation phase uses the following as inputs: the sorted rank list and context threshold value and key features. With these inputs 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". Then, the services in each list are evaluated based on key values and produce four lists: 1) "Services suit the context and key"; 2) "Different services suit the context" (services that satisfy the context requirement but do not fulfill the key requirements); 3) "Different services suit the key" (services that satisfy the key requirement but do not fulfill the context requirements); 4) "Services did not suit the context and key".

This phase performs the following tasks: 1) Apply the context threshold values. 2) Apply the key

requirements and 3) categorize the ranking result and each category are labeled with a label explaining the characteristics of the services.

VI. CASE STUDY

To evaluate the success of the proposed architecture, we present a real-world case study on (KASP) and show how to rank context universities This section presents the KASP and results.

kASP

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 perceived need of the country and economy [23]. 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 [24]. 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 [24].

Some of 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.

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.

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 or 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.

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 and non-context. 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 [5]. The data was organized in Comma-Separated Value (CSV) sheet format. Context information for Saudi Scholar was extracted from MoE.

Table 1 shows the features, their data types and semantics.

TABLE 1 . The features, their data types and semantics.

Feature	Data type	Semantic
Subject	string	EB
Program Title	string	EB
University	string	EB
City	string	EB
Duration	numeric	LB
Mode of study	string	EB
Entry qualifications	string	EB
qualifications	string	EB
English requirement	numeric	LB
Fee	numeric	LB
Accredited universities	string	EB

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. Table 2 below illustrates the relationship between features and their categories, as well as between identities and their features.

TABLE 2 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.
Unranked feature	Program description and email.	

For the sake of simplicity, we have included only eleven records ordered by league table ranking 2016. The records shown in tables 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 3. The eleven services and their context and non-context feature

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	-

Case 1: Identity has been set to "scholar student" Here, in this case, we demonstrate how Ahmad, future postgraduate student, utilize our framework to find the suitable institutions and course to apply to in the UK. He starts by building his profile and identify as shown in tables 4, and he submits the request query as in table 5, the title has a significant weight and the duration is set to best mode. University and city features have not considered weights, therefore, he is willing to accept any values for these features.

TABLE 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 5 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			

From the user profile, the system can infer the list of accredited universities based on user nationality. From identity, it can define the set of features and context rules that relate to them. Since he is a scholar student, the system inferences the following: 1) The mode of study will be full-time because they are not permitted to enroll in the part-time or online program. 2)Also, the system can infer entry requirement from student's qualification and predict the preferred qualification type. 3) 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 the context features are set to essential features both have insignificant weights and the mode for English requirement is set to exact better. Table 6 shows the query for context features and context rule.

Table 6. 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: First, the system will apply context rule, Therefor: 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 7.

Table 7 .Ranking the result based on context and non-context features case1.

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

From table 7, If we notice the first four services S3, S8, S10 and S4, we found the context features were met and the service ranked based on context features English requirement and qualification. Although S3 and S4 have, the same title value but algorithm preferred S3 over 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, the algorithm classify those services into two groups: 1) 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. 2)The second group, called services did not suit the context, includes the rest of the services S4, S6, S11 and S9. 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 ranked based on context features and take 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 over 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.

CONCLUSION AND FUTURE WORK

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

We mainly extended X-algorithm from semantic-based and user-centric [2] to context- centric. 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 reduce 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.

We are using explanation techniques to categorize services based on context information. By providing labels to explain to the user why these services are presented to them.

An implementation of a case study in the KASP is presented. 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.

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

As future work, we plan to incorporate trustworthiness features to rank and provide reliable services based on user preferences.

ACKNOWLEDGEMENT

This project 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). The authors also, acknowledge with thanks Science and Technology Unit, King Abdulaziz University for technical support”

REFERENCES

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

[2] Alsaig, V. Alagar, M. Mohammad, and W. Alhalabi, “A user-centric semantic-based algorithm for ranking services: design and analysis,” SOCA, pp. 1–20, Sep. 2016.

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

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

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

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

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

[8] 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.

[9] Schmidt, M. Beigl, and H. Gellersen, “There is more to Context than Location,” Computers and Graphics, vol. 23, pp. 893–901, 1998.

[10] Bettini et al., “A survey of context modelling and reasoning techniques,” Pervasive and Mobile Computing, vol. 6, no. 2, pp. 161–180, Apr. 2010.

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

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

[13] [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] H. Tianfield, “Context-aware service discovery in pervasive environments: A survey,” International Transactions on Systems Science and Applications, vol. 7, no. 3/4, pp. 314–338, 2011.

[15] H. Cao et al., “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.

[16] 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.

[17] 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.

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

[19] 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.

[20] W. Mansoor et al., "Adaptation in context-aware pervasive information systems: the SECAS project," International Journal of Pervasive Computing and Communications, vol. 3, no. 4, pp. 400–425, 2008.

[21] J. Berri and R. Benlamri, "Context-Aware Mobile Search Engine," Next Generation Search Engines: Advanced Models for Information Retrieval: Advanced Models for Information Retrieval, p. 371, 2012.

[22] 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-Adap Inter, vol. 22, no. 4–5, pp. 317–355, Mar. 2012.

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

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

★★★