

# A Method for Aggregating Ranked Services for Personal Preference Based Selection

Kenneth K Fletcher, University of Massachusetts Boston, USA

## ABSTRACT

Typically, users' service requests, which are similar with varying preferences on non-functional attributes, may result in ranked lists of services that partially meet their needs due to conflicting non-functional attributes. The resultant multiple ranked lists of services that partially satisfies the user's request makes it challenging for the user to choose an optimal service, based on his/her preference. This work proposes a method that aggregates multiple ranked lists of services into a single aggregated ranked list, where top ranked services are selected for the user. Two algorithms are proposed; 1) Rank Aggregation for Complete Lists (RACoL), that aggregates complete ranked lists and 2) Rank Aggregation for Incomplete Lists (RAIL) to aggregate incomplete ranked lists. Examples using real-world airline services to evaluate both algorithms show that the results from both proposed algorithms closely represent the sets of ranked lists better than using alternative approaches. Experiments were also carried out to validate their performance.

## KEYWORDS

Non-Functional Attributes Relationship, Personalized Preference, Rank Aggregation, Service Aggregation, Service Selection, Similarity Measures

## 1. INTRODUCTION

Services technology is well recognized as an easy way to integrate applications without boundaries. Owing to this, most organizations tend to publish their services on the web for easy consumption by the public. This has exponentially increased the number of available services. Different service providers may offer services that are equivalent with respect to their functionality. Therefore, to provide a differentiation among the competing services, non-functional attributes are used; allowing prospective users to choose the services which best suit their preferences.

Usually, a user's personal preference for a service comprises of his/her preference for non-functional attribute(s) that describe the service. These non-functional attributes may be conflicting (Fletcher, Liu, & Tang, 2015; Liu, Fletcher, & Tang, 2012), resulting in ranked list of services that partially meet the user's preference. In an attempt to obtain services that completely meet his/her request, the user may submit similar multiple service requests. Similar requests here refer to requests with the same functionality and trade-offs on non-functional attributes, but with varying preference on those non-functional attributes. These requests may also yield ranked lists that do not fully satisfy the user's preference due to the same conflicting relationship that may exist among non-functional attributes. In such a situation, it becomes challenging for the user to compare the different ranked lists in order to choose the optimal service. This is because, each ranked list may contain huge number of services that makes it time consuming for users to compare them against services in other ranked lists. A naïve way for this comparison will be to merge the different sets of ranked lists into one

DOI: 10.4018/IJWSR.2019040101

list and rank the merged list with respect to the user's search intent. However, this naïve way also has its own challenges: 1) the same service may rank differently in other ranked lists and therefore becomes hard to determine its overall rank; and 2) some of the ranked lists may contain services that do not appear in other ranked lists due to the varying requests, making it difficult to determine the overall rank of such services. In order for the user to obtain an optimal service, based on the ranked lists, there is the need for a method that will produce an aggregated ranked list that addresses the challenges mentioned above.

This paper proposes just such a method and demonstrates its application with specific cases. The proposed method aggregates ranked lists obtained from a user's service request and produces top-k services for user selection. The aggregated ranked list is a compromise among each ranked list. The method considers the user's personalized preferences on non-functional attributes and their trade-offs including linguistic weights on non-functional attributes when searching for services (Fletcher et al., 2015; Liu et al., 2012).

Two algorithms are proposed to compute an aggregated ranked list from each ranked list of services. The first algorithm, Rank Aggregation for Complete Lists (RACoL), aggregates complete ranked lists (i.e. ranked lists given by total orders). There are however instances where the ranked lists to be aggregated come with incomplete ranked lists (incomplete orderings); i.e. some ranked list(s) contain services that do not appear in other ranked list(s). The second algorithm, Rank Aggregation for Incomplete Lists (RAIL) is proposed to aggregate incomplete ranked lists. Examples using real-world services to evaluate the method are presented. Results from the application examples show that compared to alternative approaches, the proposed method closely represents the input ranked lists. Experiments are further carried out to validate the performance of the proposed algorithms.

## 1.1. Motivating Example

As a motivating example, consider a user whose search intent is to find a cheap nonstop flight from New York to London that gets to London in the shortest possible time. It is obvious that this user is concerned about three non-functional attributes, price, stops, and duration. In addition, it can be inferred from the requirement that the user is not willing to trade-off any of these non-functional attributes. The user's request functionality and trade-off strategy can be written as follows:

Functionality: Flight from New York to London

Trade-off Strategy: Price AND Stops AND Duration (Fletcher et al., 2015)

A "cheap" flight for this user is any flight under \$1,150 and the phrase "shortest possible time" to describe the user's preference on duration non-functional attribute refers to any flight that gets to London in under 7 hours. Preferences for each non-functional attribute for his/her initial request is shown in Table 1.

For the sake of this example, let's assume the user's requirement for nonstop criteria is satisfied but his/her requirement for price and duration were not met. For this reason, the user relaxes his/her nonstop and duration criteria and then performs the search again using modified preferences on the non-functional attributes from the initial request (see first modified request in Table 1). Again, assuming the user's stop criteria was met but neither requirements for price nor duration were met. Since none of the flights fully satisfies the user's preferences he/she decides to modify the preferences on the non-functional attributes and try the search one more time (see second modified request in Table 1). It is assumed that ranked list of flights generated from this request, satisfies the price and stops non-functional attributes but not duration.

After a third search, let's assume that the user realizes that his/her search intent cannot be completely satisfied. However, he/she needs to choose a flight based on the previously three ranked lists of flights obtained so far. It becomes necessary to find the optimal flight with respect to the user's requests. This can be achieved by aggregating the three ranked lists of flights, to produce an aggregated ranked list. The aggregated ranked list is a compromise between the previously three ranked lists and closely represents the user's search intent.

21 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:

[www.igi-global.com/article/a-method-for-aggregating-ranked-services-for-personal-preference-based-selection/226287?camid=4v1](http://www.igi-global.com/article/a-method-for-aggregating-ranked-services-for-personal-preference-based-selection/226287?camid=4v1)

This title is available in InfoSci-Journals, InfoSci-Journal Disciplines Business, Administration, and Management, InfoSci-Journal Disciplines Computer Science, Security, and Information Technology, InfoSci-Networking, Mobile Applications, and Web Technologies eJournal Collection, InfoSci-Digital Marketing, E-Business, and E-Services eJournal Collection. Recommend this product to your librarian:

[www.igi-global.com/e-resources/library-recommendation/?id=2](http://www.igi-global.com/e-resources/library-recommendation/?id=2)

## Related Content

---

### Multi-Level Web Service Clustering to Bootstrap the Web Service Discovery and Selection

Banage T. G. S. Kumara, Incheon Paik and Koswatte R. C. Koswatte (2019).

*Innovative Solutions and Applications of Web Services Technology* (pp. 44-68).

[www.igi-global.com/chapter/multi-level-web-service-clustering-to-bootstrap-the-web-service-discovery-and-selection/214831?camid=4v1a](http://www.igi-global.com/chapter/multi-level-web-service-clustering-to-bootstrap-the-web-service-discovery-and-selection/214831?camid=4v1a)

### A Framework for Situation-Aware Adaptation of Service-Based Applications

Ioannis Patiniotiakis, Nikos Papageorgiou, Yiannis Verginadis, Dimitris Apostolou and Gregoris Mentzas (2013). *Adaptive Web Services for Modular and Reusable Software Development: Tactics and Solutions* (pp. 253-262).

[www.igi-global.com/chapter/framework-situation-aware-adaptation-service/69477?camid=4v1a](http://www.igi-global.com/chapter/framework-situation-aware-adaptation-service/69477?camid=4v1a)

## XML Data Binding for C++ Using Metadata

Szabolcs Payrits and Péter Dornbach (2009). *International Journal of Web Services Research* (pp. 18-34).

[www.igi-global.com/article/xml-data-binding-using-metadata/34104?camid=4v1a](http://www.igi-global.com/article/xml-data-binding-using-metadata/34104?camid=4v1a)

## Automatic Dynamic Web Service Composition Using AND/OR Directed Graphs

Hajar Elmaghraoui, Laila Benhlime and Dalila Chiadmi (2019). *International Journal of Web Services Research* (pp. 29-43).

[www.igi-global.com/article/automatic-dynamic-web-service-composition-using-andor-directed-graphs/231448?camid=4v1a](http://www.igi-global.com/article/automatic-dynamic-web-service-composition-using-andor-directed-graphs/231448?camid=4v1a)