ISSN 2007-9737

Automatic Knowledge-Driven Approach for Optimal Service Selection based on ELECTRE III and Quality-Aware Services

Hela Taktak[1,2,*], Khouloud Boukadi[1], Chirine Ghedira-Guegan[3], Michael Mrissa[4], Faiez Gargouri[1]

[1] University of Sfax, MIRACL Laboratory, Tunisia

[2] University of Lyon, LIRIS UMR5205, France

[3] University of Lyon, Iaelyon School of Management, France

[4] University of Primorska, InnoRenew CoE, Slovenia

{hela.taktak1, chirine.ghedira-guegan}@univ-lyon3.fr, khouloud.boukadi@gmail.com, michael.mrissa@innorenew.eu, faiez.gargouri@usf.tn

Abstract. Optimal real-time collection of a variety of environmental parameters from several environmental data sources, still remains a challenge in the selection process. As environmental web services now have access to a wider range of environmental data sources, the quality of these services can vary, even if they offer the same functionality. This competition among providers means that environmental data may differ in quality. Due to this competition, different environmental data sources compete to provide these functionally equivalent services with different levels of quality: the quality of services (QoS), as well as, the quality of the data sources themselves and their data (QoDS). Therefore, we present an approach to satisfy the need of ranking and selecting the optimal services. Our contribution is an automated knowledge-driven approach that relies on the ELECTRE III MCDM (Multi-Criteria Decision-Making) method and on quality-aware service selection, to optimally select services.

Keywords. Optimal service selection, multi-criteria decision-making (MCDM).

1 Introduction

The environmental data comes from diverse observations sent by traditional sensors (e.g., satellites, sensors, etc.), social media platforms (e.g., cell phones, etc.), or cyber-physical systems. Several challenging issues emerged, since, these data sources have different characteristics, such as the used protocols, access techniques, and data formats.

Actually, due to the diverse characteristics of these data sources, the dynamic change of data on the Web and their related quality metrics over time impacts selecting optimal data sources with their related optimal data, which remains a challenge. The access to these data sources is realized through a layer of data services.

Although the services may offer the same functionality, they can vary in terms of non-functional attributes, such as Quality of

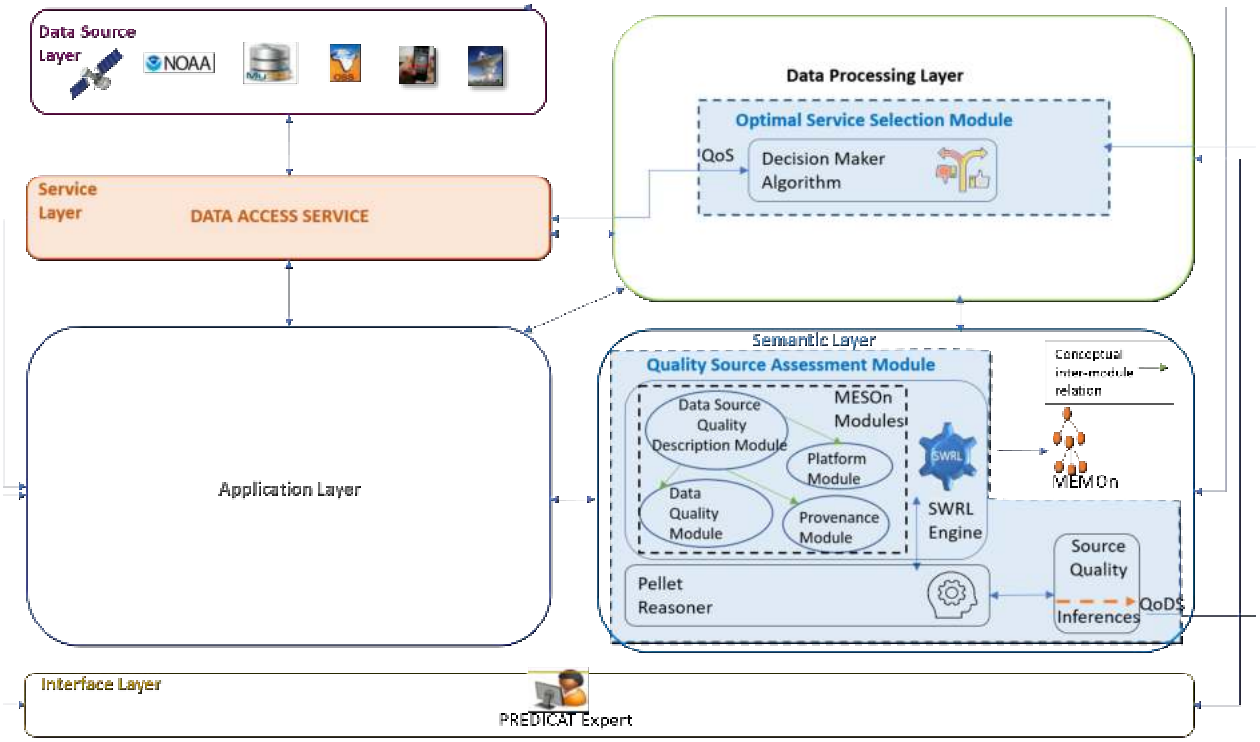

Fig. 1. Overview on the layered architecture of the proposed automatic knowledge-driven solution for optimal service selection

Service (QoS), which includes response time, availability, cost, etc.

For example, when choosing a service, one might prioritize the cheapest option, the fastest option, or perhaps a compromise between the two, that access to several data sources having different quality attributes like trustworthiness, availability, accuracy of the data sources, or also, age and accuracy related to the data itself.

Consequently, many competing services may offer the same concept with different QoS, especially, with a large number of potentially trustworthy services and constantly emerging new services. The primary concern is how to evaluate the quality of environmental data sources and the data they provide.

Otherwise, there is a need to define and explicit the qualities related to the data sources and data. This problem persists as the web environment becomes more dynamic, offering distributed large datasets that require qualification.

The second challenge involves determining the optimal selection of services, which remains a significant issue, specifically, while taking into account the quality related simultaneously to data sources, data and services. Therefore, to tackle these challenges, analyzing competitive qualities and emerging services dynamically requires intelligent analytic techniques.

These techniques provide enhanced decision-making strategies for selecting services. Various approaches have been explored for discovering and selecting web services, primarily relying on ontologies. Examples include OWLS-MX2, WSMO-MX, and SAWSDL-MX2. OWLS-MX2, WSMO-MX, and SAWSDL-MX2 [29, 28, 42].

Although these approaches focus on a better match of the functional or non-functional

ISSN 2007-9737

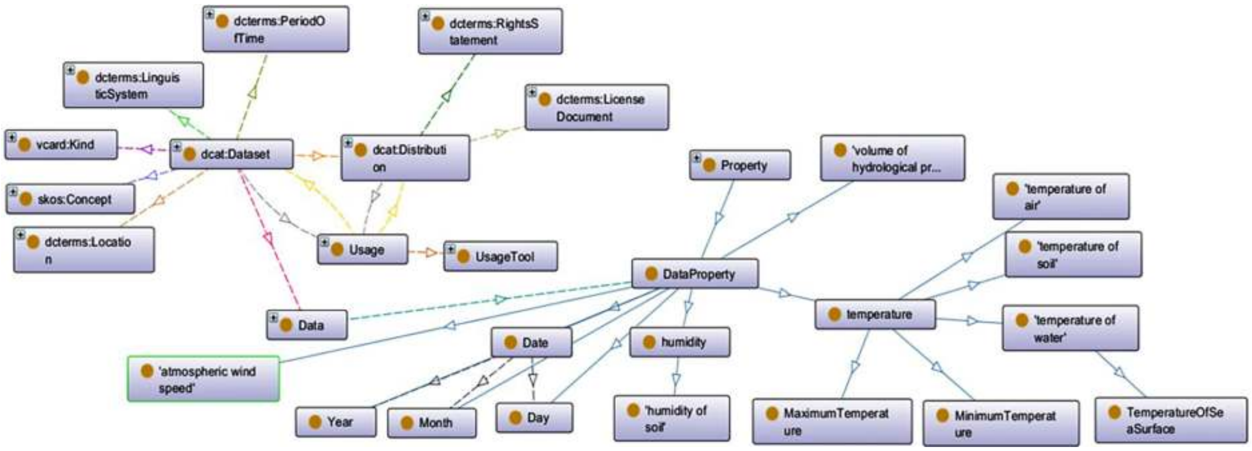

Fig. 2. The data source description module

parameters with the user requirements. They fail to offer a definitive ranking of the optimal selected services, especially when requests involve complex constraints.

These constraints include the quality of the data sources, the data itself, and the quality of services. Several approaches were proposed in the literature, to find the final ranking along with the optimal solutions for the services selection issue related to multi-criteria decision-making (MCDM) techniques. The major advantage of using a multi-criteria method is that it allows modeling the scoring of the optimal solutions, in a more realistic scenario, where a trade-off between conflicting objectives must be resolved.

Several works were proposed to resolve scoring the optimal solutions for the selection problem. Among others, according to [45], ranking approaches such that; AHP, PROMETHEE [14], and ELECTRE [43] are not suitable for directly ranking services due to their high complexity.

For this reason, works such as [37, 36, 53] used the skyline paradigm [12] to search for the optimal dominant services across an important number of services. Skyline solves the selection problem by reducing the search space of services and determines the set of the dominant services based on a Pareto-front.

Nevertheless, it presents two issues: the first one is that its retrieved dominant services are incomparable, without giving any recommendation upon which service to select, thus, causing some confusion in the decision-making process. The second issue, in a large-scale environment, a large number of skyline services could be retrieved with no ranking mechanism.

Therefore, adopting the fuzzy dominance relationship [9] allows us to address both of the stated skyline issues, since it is difficult to classify, re-filter and thus, prune more dominated services in the set of skyline services. To fulfill the aforementioned issues, we propose an automated knowledge-driven solution based on quality-aware selected services. Our value-added contributions can be summarized as follows:

– The first contribution focuses on evaluating the quality of environmental data sources and their inherent data. Our proposal ensures the freshness and reliability of these data sources and their associated data.

  To achieve this, we introduce an ontology that defines quality dimensions and their corresponding inferences for assessing the quality of data sources.

– The second contribution tackles the challenge of optimal service selection with a focus on quality-awareness. We propose an automated knowledge-driven solution for optimal service selection and ranking.

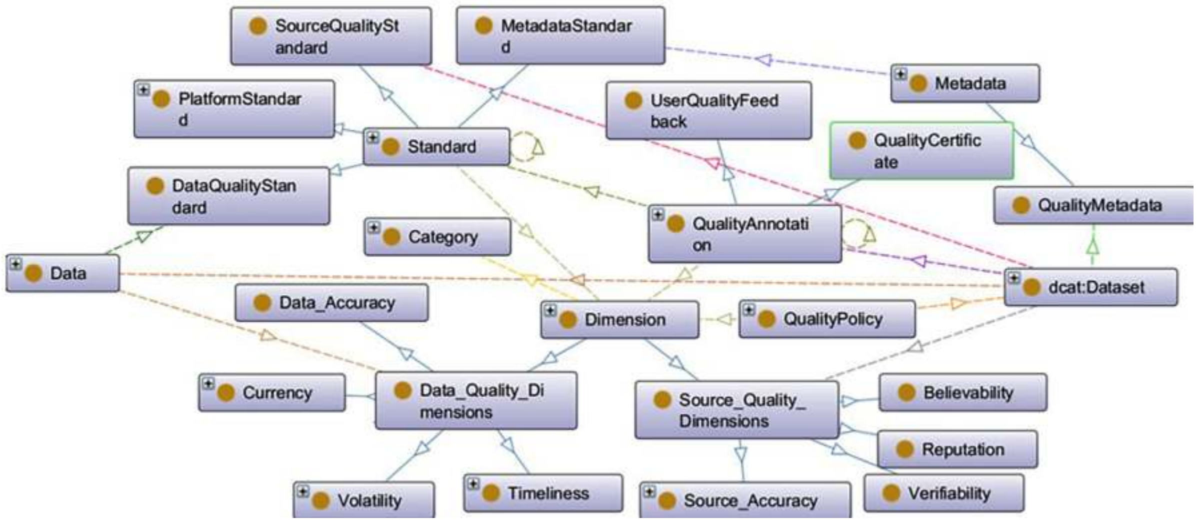

Fig. 3. The data quality module

In addition to considering the Quality of Service (QoS) of candidate services, our approach also takes into account the quality of the data sources (QoDS) as inputs for the skyline operator and the Multi-Criteria Decision-Making (MCDM) technique, specifically ELECTRE III [43].

The skyline operator is employed to reduce the search space of service candidates before proceeding to the ranking process. As for the ELECTRE III MCDM method, it offers optimal rankings of services based on their qualities [17, 16, 38, 22].

The rest of this paper is structured as follows: Section 2 provides a review of related works focusing on QoS-aware solutions using Multi-Criteria Decision-Making (MCDM) methods for optimal service selection and ranking.

Section 3 presents an overview of the layered architecture of our automated knowledge-driven solution for optimal service selection.

Section 4 elaborates on our definition of quality dimensions related to data sources using our proposed modular source ontology, the Meteorological and Environmental Source (MESOn) ontology, along with its related quality dimensions and modules.

Section 5 presents our proposed approach for the optimal service selection, denoted Bα-DSS (Best α-Dominant Skyline Service). Section 6 details the applicability and the evaluation of our approach through several experiments related to the optimal service selection.

Section 7 presents the threats to validity related to our proposals. Finally, we conclude our findings and outline our future work in Section 8.

2 Related Work

This section overviews the most relevant related works about QoS-aware Web services selection, including MCDM approaches, and some other solutions adopted in the service selection task. In the problem of selecting services based on Quality of Service (QoS), quality dimensions, also known as quality criteria, have always been considered crucial because of their direct influence on the selection of optimal services. Various quality dimensions are linked to data sources, including trustworthiness, accuracy, and timeliness. Quality dimensions related to data provided from the data sources are also considered important in the decision-making problems. Moreover, several quality dimensions are linked to the

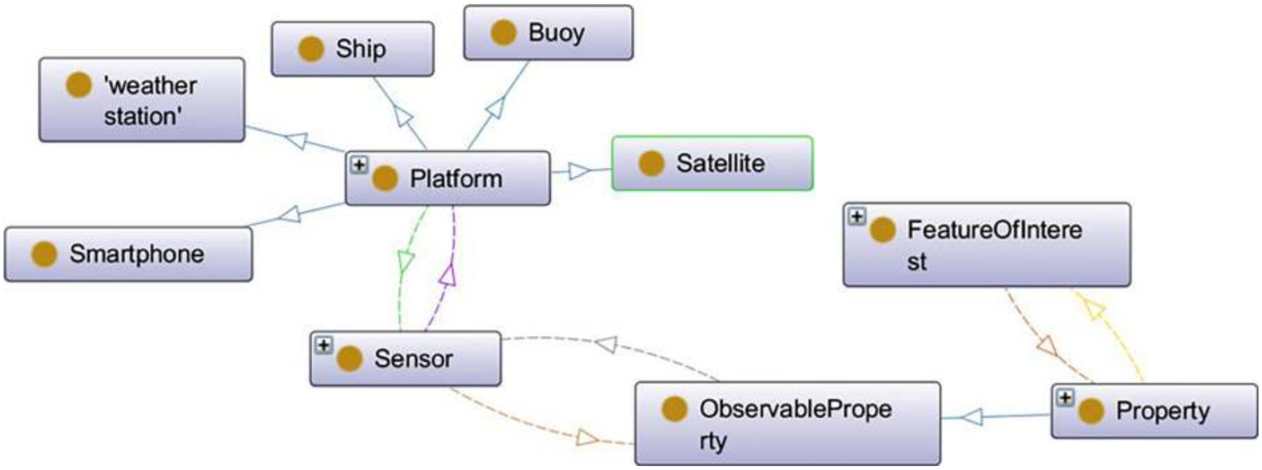

Fig. 4. The platform module

Quality of Service (QoS), including execution time, availability, reliability, reputation, and throughput.

A survey conducted by [8] organizes the building blocks of these quality dimensions into a taxonomy for dataset profiling, including their assessment, summarization, and characterization processes. Dataset profiling, as defined by [8], involves a set of characteristics, both semantic and statistical, that describe a dataset comprehensively, considering the diversity of domains and vocabularies on the Web of data.

However, one of the challenges faced in dataset profiling is computing and interpreting the profiling results. Hence, there is a need for a dedicated solution to reason about and evaluate the dataset used. Ontology stands as a suitable candidate for interpreting and explicitly assessing the quality related to environmental and meteorological data sources. Consequently, to ensure the enrichment of the data sources with quality dimensions, we propose, in this work, the MESOn ontology along with its inferences.

This ontology facilitates dataset profiling and interprets the qualities at both levels: the data source and the data retrieved by the service accessing the data source. Several works dealt with the problem of QoS-aware Web services selection. Authors in [33] and [5] adopted the Linear Programming technique to find the optimal service selection extended with a model evaluating the QoS parameters in [33].

The work in [49] developed a selection algorithm based on QoS evaluation through a QoS evaluation ontology. However, these works consider only a small number of services and QoS parameters, whereas the selection process relies on exponential space complexity. Accordingly, Recent studies concentrate on the skyline algorithm to reduce considerably the important number of services.

Moreover, the application of skyline can be considered as a pre-processing step, since it significantly reduces the search space of the service candidates, and therefore, reduces the computation time when applying the ranking and the selection algorithms.

The skyline concept was firstly introduced in the field of database, by Börzsönyi et al. [12] producing 3408 citations since 2001 (Google Scholar, May 2024). Several algorithms were proposed by Börzsönyi et al. to compute the skyline alternatives built on the Block Nested Loop (BNL) and Divide and Conquer (D&C) [12] algorithms. Other proposed progressive skyline algorithms, which are the Index and Bitmap-based algorithms [47], can output the skyline services without scanning the entire set of the alternatives. Moreover, the Nearest Neighbor (NN) and Branch and Bound Skyline (BBS) algorithms, which rely on the R-tree

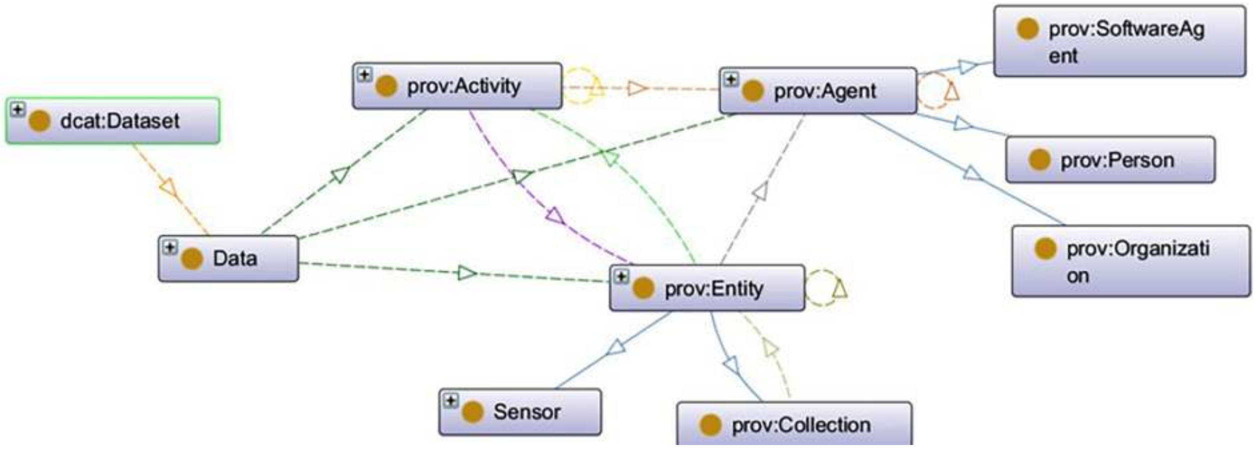

Fig. 5. The provenance module

indexing structure introduced in [39], and that can progressively scan the set of services alternatives.

In order to tackle the problem of large skyline sets, many works such that [54, 9, 17, 16, 38] are proposed, returning K-Representative services to best describe the full skyline set. However, the computation of K-Representative Skyline is a costly problem, since it is based on the multidimensional function. Additionally, the incomparability between service skyline candidates remains an issue in the K-Representative Skyline method.

Therefore, this approach may lack user control over the size of the returned skyline set, especially when dealing with a high number of quality dimensions. Additionally, it does not provide information on the comparative relationship between different skyline service candidates to select the optimal one. Previous studies have relied on the Pareto dominance relationship, as demonstrated in works such as [4, 15, 1]. Furthermore, only a few research works have combined the skyline approach with Multi-Criteria Decision-Making (MCDM) based approaches to solve the QoS-based selection problem and rank the services to select an optimal one. In addition, it is worthy to note, knowledge-driven MCDM methods are only considered, in recent past years. In their work, Dorfeshan et al. [19] introduced a novel data- and knowledge-driven Multi-Criteria Decision-Making (MCDM) method to reduce dependence on expert assessments.

They employed an extended version of the data-driven Decision-Making Trial and Evaluation Laboratory (DEMATEL) method to determine the criteria weights. Additionally, they used the knowledge-driven ELECTRE and VIKOR methods to rank the alternatives.

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method, as described by Zou et al. [56], has been widely utilized as a decision support method in various studies. Its applications include selecting property development locations, cars [57], [6], and mobile applications [26].

In another study, TOPSIS was employed by authors in [52] to optimize service selection on the cloud. Polska et al. [41] developed a web service selection approach based on sensitivity analysis. They compared the Logic Scoring of Preference (LSP) method with other MCDM methods such as Analytic Hierarchy Process (AHP), VIKOR, and TOPSIS.

Sun et al. [46] introduced a fuzzy decision-making framework and MCDM-based approach for cloud service selection. Their work involved the use of a fuzzy ontology to model uncertain relationships between objects in databases for service matching.

ISSN 2007-9737

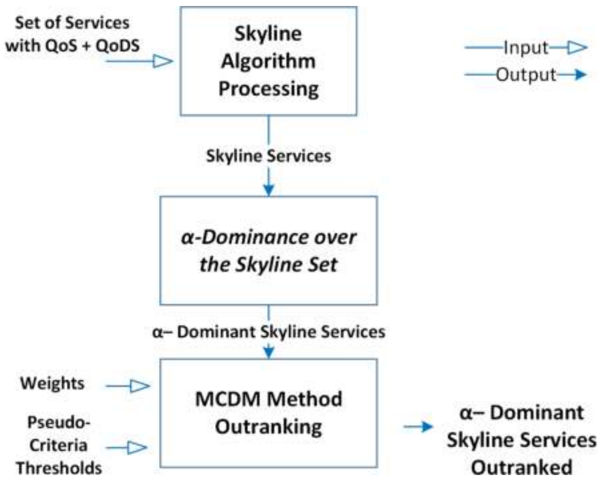

Fig. 6. Bα-DSS steps approach

They employed the AHP method to calculate semantic similarity between concepts and the TOPSIS method for multi-criteria decision-making to rank cloud services. In a related study, Kumar et al. introduced a new framework named CSS-OSSR (Optimal Service Selection and Ranking of Cloud Computing Criteria) [31].

They utilized the TOPSIS method to determine the final ranking of cloud services. Another approach, proposed by Ouadah et al. [37], is called SkyAP-S^3, which integrates skyline with AHP and PROMETHEE to rank services.

Serrai et al. [44] proposed a method that combines skyline with several MCDM (Multi-Criteria Decision-Making) techniques such as SAW, VIKOR, and TOPSIS for service selection and ranking. Xu et al. [51] addressed the QoS-aware service selection problem based on user preferences and fuzzy datasets.

They utilized fuzzy set theory and a fuzzy genetic algorithm to rank web services. In [9], authors proposed the α-dominance principle to rank the Web services based on the quality of services and assigned a fuzzy dominating score to services. However, to our best knowledge, none of these works considered at the same time, the skyline paradigm reducing the services number, the fuzzy degree dominance enabling classifying, re-filtering and pruning more services in the service skyline set, and finally the ranking mechanism.

Although these solutions are a promising direction, efforts are still needed for enforcing and optimizing the quality of solutions. While the existing solutions represent a promising direction, there is still a need for further efforts to enforce and optimize the quality of these solutions.

Our study is related to previous research on the QoS-aware service selection problem, aiming to automatically select optimal services based on their quality dimensions. However, these prior studies overlook the quality of the data sources (QoDS) and fail to account for the dynamically changing service environment.

Furthermore, QoS may be constantly changing, so it is essential that the service selection and ranking be automatic and knowledge-driven. Furthermore, as far as we know, these studies did not emphasize the importance of providing support for describing and inferring the constantly changing quality attributes of environmental data sources.

3 Overview of the Layered Architecture of the Automatic Knowledge-Driven Solution for the Optimal Service Selection

We present an overview of the layered architecture of our proposed automatic knowledge-driven solution for optimal service selection. This solution is developed within the framework of the PREDICAT (PREDIct natural CATastrophes) project, aimed at predicting natural disasters resulting from climate changes[1].

This architecture encompasses seven layers, namely: (1) data source layer, (2) service layer, (3) application layer, (4) user interface layer, (5) semantic layer, and (6) data processing layer. These layers are illustrated in Figure 1.

Our contribution to the automatic knowledge-driven solution for optimal service selection is twofold. Firstly, it includes the Quality Source Assessment Module within the semantic layer. Secondly, it involves the Optimal Service Selection Module situated in the data processing

[1] sites.google.com/view/predicat/predicat.

ISSN 2007-9737

Table 1. Weight assignment for quality dimensions

Weights	S_ET	S_Av	SO_Acc	QualAssesModule	SO_DTime	QualAssesModule
Assigned Weights$_i$	6	6	4	4	3	1
w_i	0.25	0.25	0.167	0.167	0.125	0.041

layer. These modules are highlighted in blue, in Figure 1.

The data source layer comprises various data sources, including meteorological and environmental observations (EO) from organizations such as NASA, Copernicus, OpenWeather, and others. In the service layer, RESTful (Representational State Transfer - [21]) services are automatically generated to facilitate access to the diverse environmental data sources.

The application layer falls outside the scope of our current work. The user interface layer includes a sophisticated user interface that communicates danger alerts to the experts using the PREDICAT platform.

The semantic layer includes the Meteorological and Environmental Source Ontology (MESOn), which is our first contribution. It also incorporates the Modular Environmental Monitoring Ontology (MEMOn), which comprises a collection of ontological modules addressing various sub-domains within environmental monitoring. MESOn ontology is encompassed in the Quality Source Assessment Module.

This latter evaluates and describes the meteorological and environmental data sources, and their related quality dimensions presented, in section 4. The qualities related to the environmental data sources are captured dynamically.

Moreover, to deduce and analyze the quality dimensions of a specific environmental data source, we employed SWRL (Semantic Web Rule Language) rules along with the Pellet reasoner.

As a result, we obtain the Quality of Data Sources (QoDS) inferences that will be used by the Optimal Service Selection Module and, specifically by our decision-maker algorithm (i.e., Bα-DSS), to select optimally an access service to the assessed data source. These QoDS are queried through the SWRL rules, from the MESOn ontology.

This paper focuses only on the semantics applied for the quality of source assessment through the MESOn ontology. Furthermore, semantics were used in data source profiling according to reasoning on two levels: the quality of the data source itself and the quality of the data returned by the service accessing that data source.

The data processing layer includes our second contribution, which is the Bα-DSS. It deals with the process of selection of the optimal services, through a ranking mechanism. To this end, the data processing layer encompasses the Optimal Service Selection Module.

This module, responsible for identifying optimal service candidates for service composition, employs successive analytical methods to eliminate less important services (i.e., dominated services). Additionally, it utilizes the multi-criteria decision-making method, ELECTRE III, to rank service alternatives.

Furthermore, this module encompasses our proposed decision-maker algorithm referred to as Bα-DSS, that implements the already mentioned analytical methods, taking into consideration the assigned preferences to the criteria (i.e., the criteria related to the quality dimensions QoDS and QoS) specified by experts of the PREDICAT platform.

We provide, in the following, further details on the ontology-based quality assessment to represent the MESOn ontology with its related quality dimensions and modules.

4 Ontology-Based Quality Assessment

The aim of the ontology-based quality assessment process is to assess the quality of both the data sources and the data they provide. The proposed MESOn ontology relies on the Quality Source Assessment Module, encompassed in the semantic layer.

Algorithm 1: Calculate Skyline Services

Input: List of Services S, List of Criteria $\mathrm{ListCrit}$
Output: List of Skyline Services Sky
Function ComputeBNLSkyline;
foreach p *in* S **do**
  if $(\mathrm{Sky} = \emptyset)$ **then**
    $\mathrm{Sky} \leftarrow \{p\}$;
  foreach q in S **do**
    $\mathrm{Res} \leftarrow \mathrm{ComparisonFct}(p,\ q,\ \mathrm{ListCrit})$;
    if $(\mathrm{Res} > 0)$ **then**
      $\mathrm{Sky} \leftarrow \mathrm{Sky} \cup \{p\}$;
      $S \leftarrow S - \{q\}$;
    else if $(\mathrm{Res} < 0)$ **then**
      $\mathrm{Sky} \leftarrow \mathrm{Sky} - \{p\}$;
return Sky;
End Function

We detail, in the following, at first, the selected quality dimensions adequate to our requirements describing the data sources and their data quality (QoDS), in addition to the Quality of Service (QoS). Then, we present how to calculate them. Finally, we highlight the semantic aspects through the Quality Assessment Module, which incorporates the MESOn ontology and its related modules.

4.1 Quality Dimensions

Quality dimensions, includes both quality related to data sources and quality related to data. Quality dimensions are commonly conceived as a multidimensional construct, where each dimension represents quality-related characteristic as a multidimensional construct; such as accuracy, timeliness, completeness, relevancy, objectivity, believability, understandability, consistency, and conciseness [3].

Furthermore, Quality dimensions are often grouped into categories known as quality categories. Each quality category comprises one or more computed quality metrics, whose values serve as indicators of quality. According to [27], there are 127 data quality dimensions identified in the literature.

Considering the objectives of our study for the Quality Source Assessment Module, we have selected specific quality dimensions, including source accuracy, and trustworthiness for the data source, and volatility, currency, and timeliness for the data.

4.2 Computing the Quality Dimensions

In order to evaluate the quality dimensions, we describe in the following, our proposal through a formal approach to compute the different retained quality dimensions. Source Accuracy : refers to the percentage of provided values that are consistent with the given gold standard, as described in the literature [7]:

$$\mathrm{SourceAccuracy} = \frac{NG}{NT}, \tag{1}$$

where:
$NG =$ Number of instances of data flagged as good.

$NT =$ Number of total values.

Volatility describes the time period during which information remains valid in the real world, as in [25].

It is the length of time, where data remains valid, as in [7, 40]. Currency, concerns how promptly data are updated with respect to changes occurring in the real world in [7]:

$$\mathrm{Currency} = \mathrm{Age} + \mathrm{DeliveryTime} - \mathrm{InputTime}, \tag{2}$$

where:

– DeliveryTime: Indicates the time when the data are delivered to the user.

– InputTime: Denotes the time when the data are received by the system.

– Age: Represents the age of the data when first received by the system.

– Timeliness: refers to the suitability of the data age for the specific task [48]:

$$\mathrm{Timeliness} = \max\left(0,\ 1 - \frac{\mathrm{Currency}}{\mathrm{Volatility}}\right). \tag{3}$$

– Trustworthiness: The trustworthiness category consists of three dimensions: believability, reputation, and verifiability.

Algorithm 2: Calculation of α-Dominant Skyline Services

Input: α: Degree of dominance
ϵ: ϵ-value
λ: λ-value
Sky: List of Skyline Services
Output: α_Sky: List of α-Dominant Skyline Services
α_Sky $\leftarrow \emptyset$
Function
Computeα_DominantSkyServices(α, ϵ, λ, Sky):
  foreach Element in Sky **do**
    deg $\leftarrow$ Compute_Degree_Service(Element, NextElement)
    remove(Element)
    if deg $\geq \alpha$ **then**
      α_Sky $\leftarrow$ List_Of_α_Dominant_Services()
  return α_Sky
End Function.

– Believability: Refers to the extent to which data are considered true, real, and credible [7].

– Verifiability: Refers to the degree and ease with which the information can be checked for correctness [11, 7].

– Reputation: is a judgment made by a user to determine the integrity of a source. It can be associated with a data publisher, a person, organization, group of people or community of practice, or it can be a characteristic of a dataset [48, 7].

Due to the correlation of believability, verifiability, and reputation, and for simplification reason, we chose to treat the trustworthiness as a block.

Many authors dealt with trustworthiness by proposing different ways of calculation. We opted to assess trust using two approaches:

Models and tools. For models, we employed the 7Ws Model [23], which involves answering seven questions and then calculating a score between 0 and 7 based on the responses.

More information about these questions is provided in the subsequent sub-section 4.3.2.

4.3 MESOn: A Source Ontology with Quality Dimensions

In order to evaluate the quality related to data sources and their inherent data, to our best knowledge, there is no ontology dedicated to explicit environmental and meteorological data sources and to assess their quality, in order to interpret and exploit this assessment. We tended to use ontology, owing to the fact, to define a shared conceptualization of our problem related to the assessment of the data source qualities.

Hence, we chose to design a data source ontology by reusing some fragments from other ontologies and vocabularies. In this context, we have analyzed the existing ontologies and vocabularies. We examined the available ontologies and vocabularies, and then introduced our MESOn ontology, which includes quality dimensions associated with meteorological data sources.

We incorporated fragments from validated ontologies such as the Dataset Quality Ontology (daQ) [18], Data Quality Vocabulary (DQV) [2], Data Catalogue Vocabulary (DCAT) [34], Data Usage Vocabulary (DUV) [20], PROV-O ontology [32], and SOSA/SSN Ontology.

Consequently, our proposed modular ontology is stable and the reused fragments respect the W3C standards. We present, in the following, the main modules encompassed in our proposed MESOn ontology. Then, we detail how to use inferences to reason on the assessment of the quality of the data sources, in MESOn ontology.

4.3.1 The Source Ontology Modules

We adopted a modular approach, a recognized best practice for developing high-quality ontologies, which facilitates easier maintenance and promotes reusability. Therefore, MESOn is constituted of four modules, detailed in the following.

– The Data Source Description Module Figure 2 details the Data Source Description Module with its related classes.This module focuses on describing the data source, detailing the dataset (dcat:Dataset) and its characteristics.

ISSN 2007-9737

Table 2. Reasoning time for the MESOn ontology

Data Source	Reasoning Time (ms)
Copernicus	141
NASA	150
OpenWeather	110
NOAA	159
CHIRPS	133
GPCP	139
UCSB Climate Hazard Center	147
OSS	127
HWSD	130

These characteristics include the time period (dcterms:PeriodOfTime), observation locations (dcterms:Location), linguistic system used (dcterms:LinguisticSystem), and the various types of data it contains (vcard:Kind).

This module contains, also, information about the form of the dataset (i.e., the dcat:Distribution). For instance, it describes the type of dataset (e.g., XML dataset, Web service, database, etc.) and its specific data properties such as the URL, username, and password.

Additionally, this module provides information about how the dataset is used, the tools that manipulate it, and the required license for its usage. All these characteristics related to the description of the data source, provide information on what is the format of the dataset and which is the tool to open it.

– The Data Quality Module Figure 3 describes the Data Quality Module with its related classes. This module elaborates on various quality characteristics, encompassing quality dimensions, standards, certificates, quality policies, and user feedback on quality.

  Its primary objective is to evaluate the quality of meteorological and environmental data sources and their associated data, utilizing the SourceQuality and DataQuality classes. We have represented the calculated quality dimensions based on the details provided in sub-section 4.2.

– The Provenance Module Figure 5 depicts the Provenance Module with its related classes. This module reuses fragments from the provenance ontology (i.e., PROV-o). The provenance module provides information about the data lineage, indicating the origins of a data unit. Its main concepts include Entity, Activity, and Agent.

  The Agent class represents the entity responsible for carrying out activities. Agents can be categorized as SoftwareAgent, Person, or Organization. The Activity class illustrates the activities involved in generating the data. These activities are performed by agents and entities.

  The Entity class showcases entities involved with data units. As depicted in figure 5, Sensor is an Entity and Collection is a class, which includes a group of entities (e.g., Sensor Network).

– Platform Module Figure 4 represents the Platform Module The module includes descriptions of platforms capturing meteorological and environmental observations (e.g., temperature) and the sensors they host (e.g., smartphones and satellites).

  Each sensor tracks an observable property and its feature of interest. For example, if we consider air temperature as the observation required, measured by an iPhone, the platform would be a smartphone represented by an individual named “iPhone 9-IMEI 35-207776-824955-0”.

This platform contains a sensor represented by the individual “Bosch SensortecBMA253”. The observable property is “Air Temperature” and its feature of interest is “Earth Temperature”.

In the next sub-section, we detail how to use inferences to reason on the proposed data source MESOn ontology, in order to assess the quality of the EO data sources and their related data (QoDS).

4.3.2 Inferences

Our proposed inferences are related to the source accuracy, currency, volatility, timeliness, and trustworthiness. Source Accuracy: The quality of the source accuracy can be computed along two cases. The first one is when the Quality Control Levels are provided with the data observations.

Therefore, the source accuracy is deduced from the accuracy of all items of the data source. In this case, a coefficient for each Quality Control Level is attributed. If the level 1 of Quality control is checked then, a coefficient of 0.5 is attributed.

For the level 2, the coefficient is 0.75 and for the level 3, the coefficient is 1. We adopted these coefficients according to a gradual logic, which correspond to our requirements. The source accuracy in this case is calculated as following:

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y) ^
numberOfInstances(?x,?nins) ^
numberOfQCLevel1(?x,?nqc1) ^
numberOfQCLevel2(?x,?nqc2) ^
numberOfQCLevel3(?x,?nqc3) ^
swrlm:eval(?res,"((nqc1 * 0.5 + nqc2*
0.75 + nqc3)/nins)",?nqc1, ?nqc2,
?nqc3, ?nins)->accuracy_value(?y,?res)
```

The second case is when no quality control annotations are provided with the observations, in the data source. Therefore, we proposed in the procedures of quality controls to assign one of these flags: (Good, Inconsistent, Doubtful, Erroneous, Missing Data) for each observation. Subsequently, the source accuracy is computed, as defined in sub-section 4.2, in Eq. 1.

```
dcat:Dataset(?x)^
Source_Accuracy(?dim)^
hasSourceQualityDimensions(?x,?dim)^
numberOfFlagCorrect(?x,?ncf) ^
numberOfInstances(?x,?nins) ^
swrlm:eval(?res,"(ncf/nins)",?ncf,
?nins)->accuracy_value(?dim,?res)
```

– Currency: We have adopted the following rule to compute the quality of Currency:

– Currency = Age + DeliveryTime - InputTime [40, 7] which can be translated in our case as following:

– Currency = (CurrentDate - Max (Date_Dataset)) + (CurrentDate - LastModification).

Currency is computed according to two cases in SWRL: The first case is dedicated to assign the value of currency when it is greater than 0.

```
dcat:Dataset(?x) ^ Currency(?y)^
hasDataQualityDimensions(?x,?y)^
terms:PeriodOfTime(?p)^
terms:temporal( ?x,?p)^
temporal:add ( ?currentDate, "now",0,
"Days") ^ end( ?p,?e)^
dataset_modified( ?x,?date_modified)^
temporal:duration( ?duration,
?currentDate,?date_modified,"Days")^
temporal:duration ( ?d, ?currentDate,
?e, "Days")^
swrlb: add (?currency, ?d,?duration)^
swrlb: greaterThan (?currency, 0)->
currency_value (?y, ?currency)
```

The second case is applied, when the obtained currency value is equal to 0 or negative. Subsequently, we assign 0 instead.

```
dcat:Dataset(?x) ^ Currency(?y) ^
hasDataQualityDimensions(?x, ?y) ^
terms:PeriodOfTime(?p) ^ terms:
temporal (?x, ?p) ^
temporal:add(?currentDate,
"now", 0, "Days") ^ end(?p, ?e) ^
dataset_modified(?x,
?date_modified) ^
temporal:duration(?duration,
?currentDate, ?date_modified,
"Days") ^ temporal:duration(?d,
?currentDate, ?e, "Days") ^
swrlb:add(?currency, ?d,
?duration) ^
swrlb:lessThanOrEqual(?currency, 0)->
currency_value(?y, "0.0"^xsd:double)
```

– Volatility: Volatility determines the length of the time data remains valid. We considered the following rule to check the volatility of data:

– Volatility = Currentdate $<$ (LastModification + accuralPeriod).

By applying this rule, we are able to compute the duration between the current date and the date of the last modification plus the accuracy, to get the remaining validity period. This computation is achieved following two rules: The first case when the obtained volatility value is positive.

```
dcat:Dataset(?x) ^ Volatility(?y)^
hasDataQualityDimensions(?x, ?y)^
temporal:add(?currentdate, "now", 0,
"Days")^dataset_modified(?x,?datmod)^
dataset_accuralPeriodicity(?x,?ap)^
temporal:add(?datadd, ?datmod, ?ap,
"Days")^temporal:duration(?duration,
?datadd,?currentdate,"Days")^
temporal:before(?currentdate,?datadd)
->volatility_value(?y,?duration)
```

The second case is when the volatility is less or equal to 0. We assign 1 as a value in order to avoid division by 0 in the timeliness rule.

```
dcat:Dataset(?x) ^ Volatility(?y) ^
hasDataQualityDimensions(?x, ?y) ^
temporal:add(?currentdate,"now", 0,
"Days")^dataset_modified(?x,?datmod)^
dataset_accuralPeriodicity(?x, ?ap)^
temporal:add(?datadd,?datmod,?ap,
"Days" )^temporal:duration(?duration,
?datadd, ?currentdate, "Days") ^
temporal:notBefore(?currentdate,
?datadd) -> volatility_value(?y,
"1.0" ^^xsd:double)
```

– Timeliness: Timeliness determines how current the data are for the task at hand [7]. We considered the following rule for the data:

– Timeliness = Max(0, 1 - (Currency / Volatility) [7, 40].

– To compute the Timeliness, we considered two SWRL rules depending on the cases:

– The first case is when the result of timeliness is less than 0. Therefore, the value must be equal to 0. This issue was resolved according to the following rule:

```
dcat:Dataset(?x) ^ Timeliness(?y) ^
Currency(?xc) ^ Volatility(?xv) ^
hasDataQualityDimensions(?x, ?y) ^
hasDataQualityDimensions(?x, ?xc) ^
hasDataQualityDimensions(?x, ?xv) ^
currency_value(?xc, ?c) ^
volatility_value(?xv, ?v) ^
swrlm:eval(?z, "(c / v)", ?c, ?v) ^
swrlb:subtract(?t, 1, ?z) ^
swrlb:lessThan(?t, 0) ->
timeliness_value(?y, 0)
```

The second case is when the value of timeliness is greater than 0. The obtained value is taken. The rule is as follows:

```
Dataset(?x) ^ Timeliness(?y) ^
hasDataQualityDimensions(?x,?y) ^
currency_value(?x, ?c) ^
volatility_value(?x, ?v) ^
swrlm:eval(?z, "(c / v)", ?c, ?v) ^
swrlb:subtract(?t, 1, ?z) ^
swrlb:greaterThanOrEqual(?t, 0) ->
timeliness_value(?y, ?t)
```

Trustworthiness: To reason on the trustworthiness, we used the 7Ws Model [23], consisting on replying to 7 questions. The rationale behind using this model is that the provenance information related to the assessment of the trustworthiness can be identified by answering the seven questions, detailed in the following. We, therefore, created the inferences rules related to the questions of the 7Ws Model.

To compute a score ranging from 0 to 7, we base it on the answers provided. The questions are the following: We check for each question, if it is answered, by assigning a boolean value to each question: 1 as a score if the question is answered (true) or 0 in the opposite case (false).

– What is the name of the author or organization that created the dataset?

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
dqv:inCategory(?y, ?z) ^
author_b(?x, ?b) ^
swrlb:equal(?b, true) ->
score_author(?z,"1.0"^^xsd:double)
```

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
dqv:inCategory(?y, ?z) ^
author_b(?x, ?b) ^
swrlb:equal(?b, false) ->
score_author(?z,"0.0"^^xsd:double)
```

– What is the data?

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
dqv:inCategory(?y, ?z) ^
data_b(?x, ?b) ^
swrlb:equal(?b, true) ->
score_whatis(?z,"1.0"^^xsd:double)
```

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
dqv:inCategory(?y, ?z) ^
data_b(?x, ?b) ^
swrlb:equal(?b, false) ->
score_whatis(?z,"0.0"^^xsd:double)
```

– Which instruments were used to collect the dataset?

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)
dqv:inCategory(?y, ?z) ^
instruments_b(?x, ?b) ^
swrlb:equal(?b, true) ->
score_instruments(?z,"1.0"
^^xsd:double)
```

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
dqv:inCategory(?y, ?z) ^
instruments_b(?x, ?b) ^
swrlb:equal(?b, false) ->
score_instruments(?z,"0.0"
^^xsd:double)
```

– What events led to the collection of the dataset and how was it collected?

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
dqv:inCategory(?y, ?z) ^
collected_b(?x, ?b) ^
swrlb:equal(?b, true) ->
score_how (?z,"1.0"^^xsd:double)
```

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
dqv:inCategory(?y, ?z) ^
collected_b(?x, ?b) ^
swrlb:equal(?b, false) ->
score_how (?z,"0.0"^^xsd:double)
```

– Why the dataset is created?

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
dqv:inCategory(?y, ?z) ^
reason_b(?x, ?b) ^
swrlb:equal(?b, true) ->
score_why (?z, "1.0"^^xsd:double)
```

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
dqv:inCategory(?y, ?z) ^
reason_b(?x, ?b) ^
swrlb:equal(?b, false) ->
score_why (?z, "0.0"^^xsd:double)
```

– When was it collected?

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
dqv:inCategory(?y, ?z) ^
when_b(?x, ?b) ^
swrlb:equal(?b, true) ->
score_when (?z, "1.0"^^xsd:double)
```

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y) ^
dqv:inCategory(?y, ?z) ^
when_b(?x, ?b) ^
swrlb:equal(?b, false) ->
score_when (?z, "0.0"^^xsd:double)
```

– Where was it collected?

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
dqv:inCategory(?y, ?z) ^
where_b(?x, ?b) ^
swrlb:equal(?b, true) ->
score_where (?z,"1.0"^^xsd:double)
```

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
```

```
dqv:inCategory(?y, ?z) ^
where_b(?x, ?b) ^
swrlb:equal(?b, false) ->
score_where (?z,"0.0"^^xsd:double)
```

After replying to the questions, we calculate the overall score which is the ratio between the answered questions and the total questions number by applying the following rule:

```
dcat:Dataset(?x) ^
hasSourceQualityDimensions(?x,?y)^
dqv:inCategory(?y, ?z) ^
score_why(?z,?s3)^score_where(?z,?s6)^
score_whatis (?z, ?s2) ^
score_when(?z,?s5)^score_how(?z,?s4)^
score_author(?z,?s1)^
score_instruments(?z, ?s7) ^
swrlm:eval (?res,"((s1+s2+s3+s4+s5+s6
+s7)/7)", ?s1, ?s2, ?s3, ?s4, ?s5,
?s6, ?s7) ->
trustworthiness_value (?z, ?res)
```

5 Optimal Service Selection

The automatic knowledge-driven solution for optimal service selection aims at selecting the most appropriate services based on the quality of the data sources, the data itself, and the services.

Our automatic knowledge-driven solution for optimal service selection relies on the data processing and the semantic layers, accordingly, as depicted in Figure 1. To achieve this, we focused our proposed approach, on the one hand, on analytical filtering techniques to reduce the search space of services (i.e., skyline and α-dominance), and on the other hand, on an outranking method, in order to select the optimal ranked service, for a given concept.

The analytical filtering techniques are encompassed in the Optimal Service Selection Module, in the data processing layer. In this section, we present, at first, the skyline approach, a formalization of our concepts using the α-dominance principle, which is based on a dominance relationship combined with the fuzzy sets theory. Second, considering the aforementioned solutions, we hence, base our optimal service selection solution on the Best_α-Dominant_Skyline_Service (Bα-DSS) approach, presented hereafter.

5.1 Background on the adopted Skyline and Fuzzy Sets

As aforementioned in the introduction section, two mechanisms were used: The skyline and the fuzzy sets theory.

A) Skyline: The skyline operator allows retrieving all non-dominated and best alternatives based on a crisp multi-criteria comparison. According to the Pareto sense. One service dominates another, if and only if, it is at least as good as the other in all criteria and better in at least one criterion.

– **Definition 1. (Pareto Dominance)**

Let S = $(S_1, S_2, ..., S_n)$ be a set of n-dimensional services which are functionally similar. The N dimensions are the number of the considered quality criteria.

Let S_i and S_j two services of S. S_i dominates S_j, in Pareto sense, if and only if, S_i is better or equal to S_j in all dimensions and (strictly) better than S_j in at least one dimension.

We assume that a greater value is preferable in each criterion to maximize and a smaller value is preferable in each criterion to minimize. Each service S_i is characterized by a vector $Q(S_i) = (q_1(S_i), ..., q_d(S_i))$ where $q_\iota(S_i)$ denotes the value of the ι-th quality criteria related to the service S_i:

$$\forall \iota \in [1, d], q_\iota(S_i) \geq q_\iota(S_j) \wedge \exists \kappa \in [1, d], \\ q_\kappa(S_i) \succ q_\kappa(S_j). \tag{4}$$

Since the comparison of the quality criteria related to data sources and services are susceptible to the uncertainty, we introduce in the following the fuzzy sets theory.

Algorithm 3: Pseudo-Algorithm for Best α-Dominant Skyline Service (Bα-DSS)

```
Input: List of Services S, α (default value is 0.7)
  Output: Best-ranked service Best_Ranked_Service
Function Best_α_DominantSkyService
SKY ← ∅
foreach element in S do
    Sky ← ComputeBNLSkyline();
    α_Sky ← Computeα-DominantSkyServices();
    foreach element in α_Sky do
        Best_Ranked_Service ← ELECTRE III()
return Best_Ranked_Service // The 1st-ranked service
  from the set of ranked services.;
End Function.
```

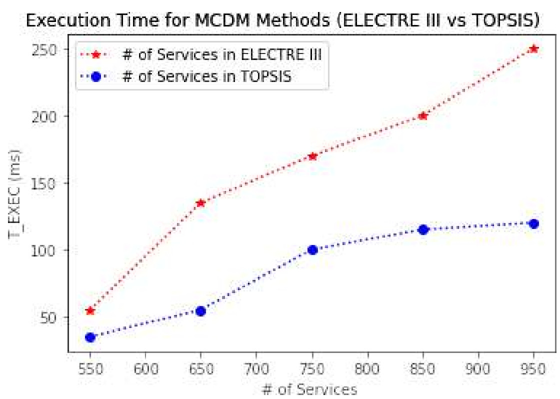

Fig. 7. Execution time for ELECTRE III and TOPSIS methods before applying skyline and α-dominance

A) Fuzzy Sets: We introduce in this part, the fuzzy sets theory and the fuzzy dominance (i.e., α-dominance). Fuzzy sets theory was first introduced in 1965, by Zadeh [55].

The usefulness of the fuzzy sets theory consists of representing vague and uncertain data. The fuzzy logic models uncertain systems to reason and help the decision-making process, when precise information is lacking.

Zadeh defines a fuzzy set as a group of objects with a range of membership grades. The rationale is that an object can belong to a set partially. Moreover, the set can be defined by a generalized membership function that assigns a degree of membership to each object, typically ranging from zero to one.

In the context of skyline computation, fuzzy sets were used to express fuzzy dominance degrees. However, there is no information available on the comparison relationship between candidates in the skyline set of services. Thus, as a first effort, the proposed querying syntax is extended with user-defined fuzzy comparators is SQLf in [13].

Another study in [30], demonstrated the effectiveness of fuzzified Pareto dominance and its application in Evolutionary Multi-Objective Optimization. In order to determine a graded dominance relationship between the different services, we define below the fuzzy dominance relationship based on a specific comparison function that expresses a graded inequality of the type “strongly greater than”.

– **Definition 2. (Fuzzy Dominance)**

Given two services S_i, S_j $\in$ S, we define the fuzzy dominance, as stated in [9], to express the degree to which a service S_i dominates a service S_j as:

$$\deg_{\mu_{\epsilon,\lambda}}(S_i \succ S_j) = \frac{\sum_{\iota=1}^{d} \mu_{\epsilon,\lambda} q_\iota((S_i),\, q_\iota(S_j))}{d}, \quad (5)$$

where $\mu_{\epsilon,\lambda}$ is a membership monotone comparison function that expresses the extent to which $q_\iota(S_i)$ is significantly greater or lesser than $q_\iota(S_j)$. The membership function $\mu_{\epsilon,\lambda}$ can be defined absolutely way (i.e., in terms of $x - y$) as follows:

$$\mu_{\epsilon,\lambda}(x,y) = \begin{cases} 0 & \text{if } x - y \le \epsilon, \\ 1 & \text{if } x - y \ge \lambda + \epsilon, \\ \dfrac{x-y-\epsilon}{\lambda} & \text{otherwise,} \end{cases} \quad (6)$$

where $\lambda > 0$, i.e., $\succ$ gives more grade information than the idea of “strictly greater” and ϵ must be ≥ 0. The semantics of $\mu_\succ$ are given in the following way: x is not significantly greater than y, when $(x - y) < \epsilon$. And x is significantly greater than y, when $(x - y) > \lambda + \epsilon$.

And x is greater than y to some extent, when $\epsilon < (x - y) < \lambda + \epsilon$. λ and ϵ are subjective parameters that are user-defined and domain-specific. They represent the semantics of the gradual relation μ within a particular domain for a specific user, as defined in [10].

– **Definition 3. (α-Dominance):**

In this section, we introduce the concept of the α-dominant service skyline, which is based on the notion of α-dominance, representing a graded form of dominance. For two services S_i, S_j $\in$ S and $\alpha \in$ [0, 1], we state that S_i α-dominates S_j (or S_i dominates S_j to a degree α) in the context of μ_ϵ, λ, denoted as $S_i \succ \mu_{\epsilon,\lambda}{}^{\alpha} S_j$, if and only if deg $\mu_{\epsilon,\lambda}$ ($S_i \succ S_j) \geq \alpha$. Otherwise, the α-dominance eliminates all the services with a degree below to the fixed α degree value.

5.2 Best_α-Dominant_ Skyline_Service (Bα-DSS) Approach

Figure 6 illustrates the three primary steps of our proposed approach for effectively selecting the optimal service from the initial set of services.

Our proposed Bα-DSS approach is enacted by the decision-maker algorithm encompassed in the Optimal Service Selection Module, within the data processing layer of the PREDICAT platform. The main objectives of our proposed approach, ensured by the decision-maker are as follows:

1. Reduce the overall number of services to decrease the search space and computational time using the skyline algorithm.
2. Support the comparison between alternatives in the retrieved dominant skyline set by applying a fuzzy degree of dominance to eliminate skyline services that do not meet the specified degree of dominance.
3. Apply an outranking mechanism to the compared set of α-dominant skyline services based on a multi-criteria decision-making method (MCDM).

To do so, Bα-DSS is composed of three main steps: the first one is performing the skyline algorithm, which is processed upon a group of functionally similar services, each characterized by various Quality of Service (QoS) and Quality of Data Source (QoDS) criteria, to retrieve the dominant set of skyline services.

This set serves as the input for the next step, which involves applying an additional filter based on the definition of a fuzzy degree of dominance. It consists on performing the fuzzy dominance principle through the α-dominance degree, computing the set of the α-dominant skyline services.

Consequently, the skyline services with a degree of dominance lower than the specified threshold are discarded. Otherwise, the combined use of skyline and α-dominance effectively eliminates dominated services. The skyline and the α-dominance help eliminate dominated services. This reduces the search space for ranking services. It is particularly useful when dealing with a large number of services.

The α-dominant skyline services are the output of the second step of our approach, that will be the input for the next step. Finally, the third step involves an outranking mechanism applied to the set of compared α-dominant skyline services, based on multi-criteria decision-making (MCDM) method. The output is an ordered set of the α-dominant skyline services.

The first outranked service will be selected as the optimal service responding to a fixed requested functional concept. We give, in the following, more details about the three steps related to our proposed Bα-DSS approach.

A) Skyline-based Services Filtering We considered six quantitative quality dimensions: Execution Time $(\mathrm{S\_ET})$, Availability $(\mathrm{S\_Av})$, and Cost $(\mathrm{S\_Cost})$, which are related to the quality of service (QoS), and Accuracy $(\mathrm{SO\_Acc})$, Trustworthiness $(\mathrm{SO\_Trust})$, and Data Timeliness $(\mathrm{SO\_DTime})$, which are related to the quality of data sources (QoDS).

ISSN 2007-9737

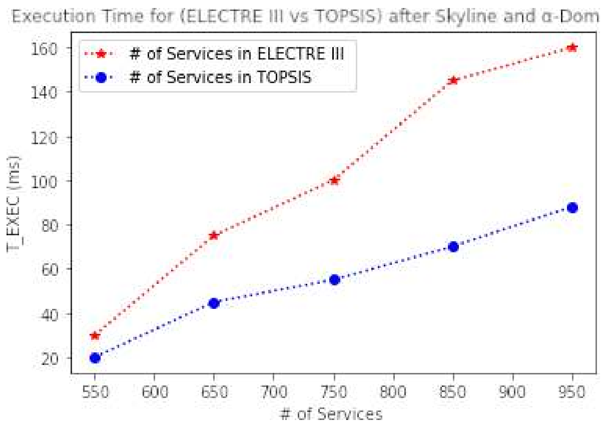

Fig. 8. Execution time for ELECTRE III and TOPSIS methods after applying skyline and α-dominance

We employed the BNL (Block-Nested- Loops) skyline algorithm to compute the set of dominant services due to its popularity and ease of use. The function ComparisonFct(p, q, ListCrit) in Algorithm 1 compares the two services p and q pairwise, considering all the criteria listed in ListCrit.

This function returns a count of the maximum number of criteria for a given service. Furthermore, the main function ComputeBNLSkyline, in Algorithm 1, retrieves the set of all the dominant services in the Sky list.

B) α-Dominant-based Services Filtering as aforementioned, we operate in a fuzzy environment, since the incomparability between the skyline service candidates remains an issue.

As a result, this second step identifies all skyline services that meet the condition of having a fuzzy dominance degree greater than or equal to the specified threshold. To compute the α-Dominant Skyline Services, our proposed algorithm 2 uses the previously detailed functions (e.g., Eq. 5 and Eq. 6). Changes in the α parameter affect the size of the resulting α-dominant skyline services.

Increasing (or decreasing) α includes (or excludes) services with lower-quality compromises. In our study, we varied the α parameter while fixing its value at 0.7. Even if the set of α-dominant skyline services may contain services with a bad compromise, they will be classified and outranked in the MCDM ELECTRE III method.

Moreover, adjusting the values of λ and ϵ enables the retention of services with a satisfactory compromise between the QoS attributes. As λ and ϵ are subjective parameters, we experimented with different values and ultimately set them to 0.2 and 0.1, respectively. These values consistently returned α-dominant services with a desirable compromise between the QoS attributes.

Furthermore, we advocate to use the α degree to 0.7, ϵ to 0.1, and λ to 0.2. These parameters yielded favorable results concerning the α-Dominant Skyline Services set, which demonstrates a satisfactory compromise between QoS attributes.

Additionally, the algorithm computes the dominance degree for each service and verifies whether the dominance degree between every pair of services is greater than or equal to the predefined α-dominance threshold. Subsequently, the algorithm retains all services with a dominance degree greater than or equal to 0.7, constituting the α-dominant skyline services set.

The next section addresses the remaining issue of ranking the α-dominant skyline services. It introduces a ranking mechanism using the ELECTRE III method, to produce an ordered services set, helping in the selection process of the optimal services. We detail in the following, the ranking mechanism.

C) ELECTRE III-based Services Outranking Different versions of ELECTRE were developed (e.g., ELECTRE I, II, III, IV, and TRI). To address the ranking problem of candidate services effectively, we opted for the ELECTRE III method (Roy, 1990) [43]. ELECTRE III was chosen for its capability to handle inaccurate, imprecise, and uncertain comparisons.

The method is based on pseudo-criteria, which act as thresholds, accommodating the uncertainty and ambiguity inherent in calculations and performance evaluations. These thresholds enable fuzzy comparisons, allowing the method to

ISSN 2007-9737

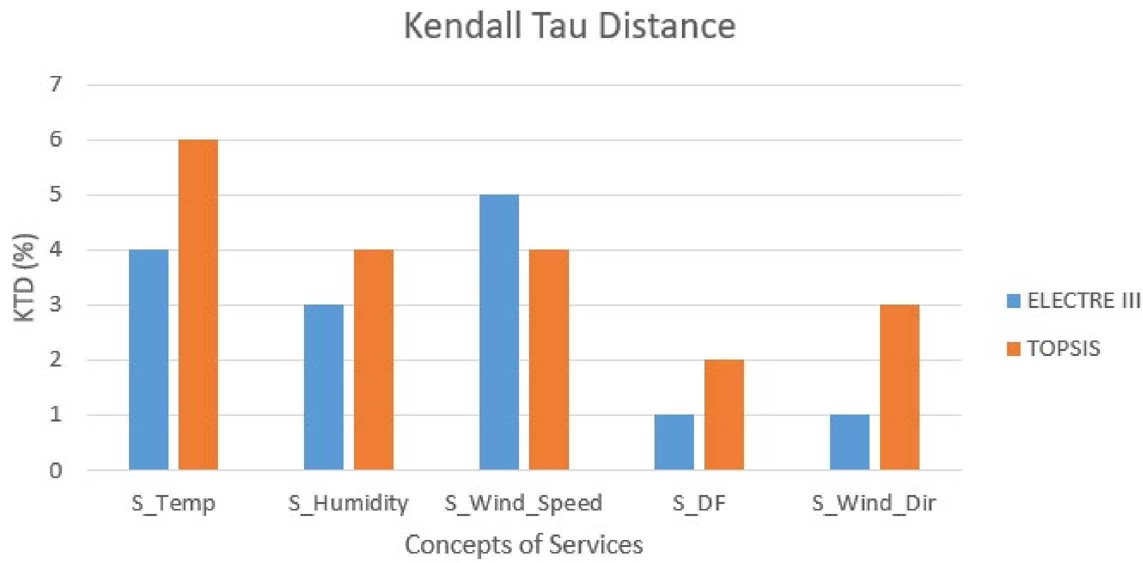

Fig. 9. Kendall Tau Distance (KTD) over ELECTRE III and TOPSIS rankings

draw conclusions based on the set of α-dominant skyline services, which are then ranked.

By exploiting knowledge derived from the quality dimensions (i.e., the criteria in ELECTRE III) of the QoDS inferred from the MESOn ontology, fuzzy comparisons are made, producing valuable decisions. In addition to outline the decision maker's preferences, ELECTRE III assigns weights and pseudo thresholds to each quality criterion.

It serves as a decision-maker to select the best compromise among all considered service alternatives and their criteria. The method is based on pairwise comparisons of alternatives, considering the extent to which evaluations of the alternatives and preference weights confirm or contradict the dominance relationship between pairwise alternatives.

In our case, the quality criterion j can be one of six quality dimensions (i.e., QoS: $\mathrm{S\_ET}$ (service execution time), $\mathrm{S\_Av}$ (service availability), $\mathrm{S\_Cost}$ (service cost), QoDS: $\mathrm{SO\_Acc}$ (source accuracy), $\mathrm{SO\_Trust}$ (source trustworthiness), $\mathrm{SO\_DTime}$ (source data Wtimeliness).

For each criterion, we defined three different pseudo-criteria: The preference threshold (p), the indifference threshold (q), and the veto threshold (v). Experts must specify values related to these thresholds for each criterion, ensuring that $(v \geq p \geq q)$, and assign an importance weight (w_j) for each criterion j, as depicted in Table 1.

PREDICAT experts assigned the important weights for $\mathrm{S\_ET}$, $\mathrm{S\_Av}$, $\mathrm{S\_Cost}$, $\mathrm{SO\_Acc}$, $\mathrm{SO\_Trust}$, and $\mathrm{SO\_DTime}$. We then normalized the criteria weights using the Weighted Arithmetic Mean, as shown in Eq. 7. This normalization ensures that the sum of the weights is equal to 1.

The Weighted Arithmetic Mean is calculated using the following formula:

$$x'_w = \frac{\sum_{i=1}^{n}(w_i x_i)}{\sum_{i=1}^{n}(w_i)}, \tag{7}$$

where :

$x_w =$ is the weighted mean,
$w_i =$ is the allocated weighted value, and
$x_i =$ is the observed value of each criterion.

ELECTRE III method encompasses several steps such that:

1. Estimation of concordance indices,

ISSN 2007-9737

Table 3. Parameter configurations for the penalty-based GA

Attribute	Value (Condition)
The Size of Population	100
Initial Population	Solutions randomly generated
Probability of Crossover	0.8
Probability of Mutation	0.1
Termination condition	No enhancement observed in the optimal individual for 30 consecutive generations

2. Estimation of discordance indices,
3. Estimation of credibility scores,
4. Performing distillation procedures, and
5. Performing the complete ranking.

The relation a outranks b, denoted aSb, is asserted by measuring the concordance and the discordance indices. In our case, a and b are the pairwise alternatives of services to be compared. We unrolled the first step of computing the concordance index by comparing the performance alternatives over each criterion individually.

This comparison is weighted, and the formula for $c_j(a,b)$ is given by Eq. 10. For example; $c_j(a,b)$ is the concordance index computed for both services a and b, which are S_1 and S_2 respectively, and which are responding to the same functional concept (e.g., temperature):

$$C(a,b) = \frac{1}{W}\sum_{j=1}^{d} w_j c_j(a,b), \tag{8}$$

where j: criterion, d: the number of the used criteria, w: the used weight corresponding to its criterion from table 1, (a and b) are the services, where:

$$W = \sum_{j=1}^{d} w_j. \tag{9}$$

And following these cases:

$$c_j(a,b) = \begin{cases} 1 \quad \text{if} \quad g_j(a) + q_j(g_j(a)) \geq g_j(b), \\ 0 \quad \text{if} \quad g_j(a) + p_j(g_j(a)) \leq g_j(b), \\ \dfrac{g_j(a) - g_j(b) + p_j(g_j(a))}{p_j(g_j(a)) - q_j(g_j(a))} \text{ otherwise,} \end{cases} \tag{10}$$

where: $g_j(a)$ and $g_j(b)$ correspond respectively, to the performance retrieved values of the quality dimension j of the services a (i.e., S_1) and b (i.e., S_2), respectively, which are responding to the same functional concept (e.g., temperature):

$p_j(g_j(a))$ corresponds to the assigned preference threshold to the performance value of the quality dimension j for the service alternative a and $q_j(g_j(a)$ corresponds to the assigned indifference threshold to the performance value of the quality dimension j for the service alternative a. The first case, when $c_j(a,b) = 1$, means that alternative a is at least as good as alternative b, with the possibility of being better, by a margin equal to the indifference threshold for criterion j.

In the second case, if $c_j(a,b) = 0$, the alternative a is considered not better than alternative b for criterion j. Otherwise, the relationship is between these two extremes.

Then, we unrolled the second step, which consists of computing the discordance index for each pair (i.e., pairwise) of alternatives a and b, for each criterion j, according to Eq. 11. The discordance index expresses the extent to which the concordance index is weakened in the

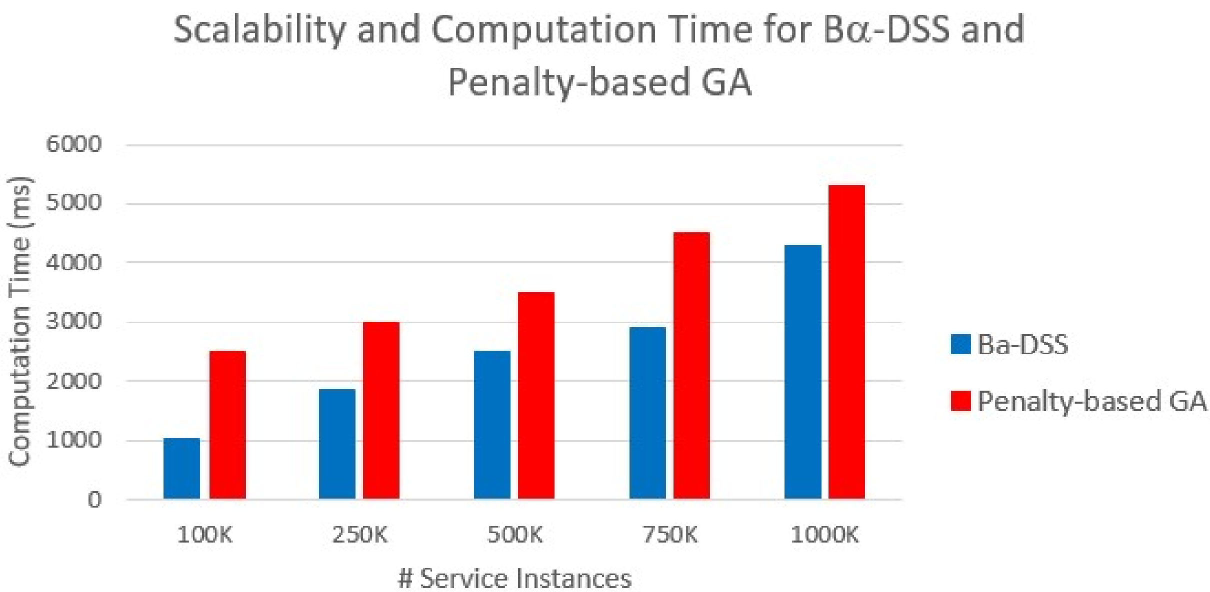

Fig. 10. Scalability and computation time (ms) for Bα-DSS and the penalty-based GA

outranking relations. It verifies the case where an alternative a (i.e., S_1) is worse than b (i.e., S_2).

It is based on the veto (v) threshold. The veto threshold for criterion j is the value from which to refuse any credibility favoring the outranking of the alternative a by alternative b, even if all the other criteria are in concordance with this outranking:

$$D_j(a,b) = \begin{cases} 1 \quad \text{if} \quad g_j(b) \geq g_j(a) + v_j(g_j(a)), \\ 0 \quad \text{if} \quad g_j(b) \leq g_j(a) + p_j(g_j(a)), \\ \dfrac{g_j(b) - g_j(a) - p_j(g_j(a))}{v_j(g_j(a)) - p_j(g_j(a))} \; \text{otherwise} \, , \end{cases} \tag{11}$$

where: j is a criterion, $g_j(a)$ and $g_j(b)$ correspond to the performance retrieved values of the quality dimension j of the services a and b, respectively. $p_j(g_j(a)$ corresponds to the assigned preference threshold to the performance value of the quality dimension j for the service alternative a. $v_j(g_j(a)$ corresponds to the assigned veto threshold to the performance value of the quality dimension j for the service alternative a.

Additionally, in the third step, we calculate the credibility score based on the concordance and discordance indices. This score indicates the degree of credibility of the outranking, depending on two scenarios. The first case occurs when no veto threshold is applied, as described in Eq. 12:

$$S(a,b) = C(a,b) \text{ if } D_j(a,b) \leq C(a,b), \forall j, \tag{12}$$

where: $S(a,b)$ is the outranking relation between the services alternatives a and b (i.e., S_1 and S_2 respectively). The second case when the level of discordance increases above a threshold value, the degree of outranking is determined by the concordance index with a reduction according to the discordance index when no veto threshold is applied, following the Eq. 13:

$$S(a,b) = C(a,b) \prod_{j \in \psi(a,b)} \frac{1 - D_j(a,b)}{1 - C(a,b)}, \tag{13}$$

where $\psi(a,b)$ represents the set of criteria for which the discordance index $D_j(a,b)$ is less than the concordance index $c_j(a,b)$. Then, as a fourth step, we performed the distillation procedures. To do so,

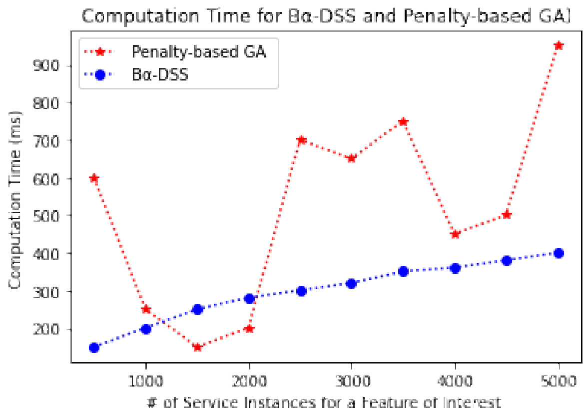

Fig. 11. Computation time (ms) for the Bα-DSS and the penalty-based GA

two iterative processes are generated to obtain two different complete pre-orders.

The first pre-order is descendant (Descendant Distillation), which selects the best alternatives initially and proceeds to the worst. The second pre-order is ascendant (Ascendant Distillation), which selects the worst alternatives initially and proceeds to the best.

Finally, depending on the resulting distillation procedures, we generated a complete ranking of the services alternatives, which are in our case, the α-dominant skyline services. The complete ranking is retrieved by the combination of the previously resulted distillation procedures.

Algorithm 3 provides details on the optimal service selected from the set of α-dominant skyline services after applying the ELECTRE III MCDM method for ranking. This algorithm returns the best-ranked service.

6 Implementation and Evaluation

Below, we outline the implementation and evaluation details of our proposed knowledge-driven solution, focusing on quality-aware service selection for optimal service ranking. We relied on the Protégé-OWL development environment for the reasoning on the quality of the data sources through the SWRL rules. Next, to rank and select the optimal QoS-aware services, we implemented the Bα-DSS method using Java.

The dataset used in Bα-DSS initially consisted of 6 sets of services (concepts). Each set consisted of 500 functionally equivalent services with different QoS attributes corresponding to a given concept. To select the optimal services for a given concept, we used the ELECTRE III MCDM method implemented in Java.

Assigning weights to each quality dimension is a prerequisite for the ELECTRE III method. The assigned weights for each quality dimension (QoS and QoDS) are shown in Table 1. These weights were normalized using the Weighted Arithmetic Mean formula Eq. 7.

6.1 Metrics for Evaluation

In this section, we present three experiments conducted to evaluate and analyze: (1) the reasoning time for data source quality in the MESOn ontology, (2) the effectiveness of our proposed Bα-DSS approach, comparing the relevance of ELECTRE III MCDM ranking results with those of the TOPSIS MCDM method, (3) the complexity assessement of our proposed Bα-DSS approach compared with Penalty-based GA, in terms of the execution time, and (4) the scalability of the Bα-DSS approach compared with Penalty-based GA by varying the dataset size of the candidate services.

6.2 Experiment 1: Reasoning Time for Data Source Quality

We assessed the quality of data sources using SWRL rules for semantic reasoning, as discussed in sub-section 4.3.2. By executing various SWRL rule queries across different environmental data sources, we evaluated the time required for data source quality reasoning. These queries were conducted using the Pellet reasoner within the Protégé 5.5.0 ontology editor. The results of this evaluation are presented in Table 2.

For instance, the reasoning time for Copernicus and NASA data sources was found to be 141 ms and 150 ms, respectively, which are reasonable durations. The execution time of SWRL queries for all data sources remained consistently low and reasonable. Therefore, the quality of data sources cannot change within this timeframe. As a result, semantic reasoning is unlikely to lead to the misselection of inappropriate data sources by the data processing layer in real-time scenarios.

6.3 Experiment 2: Execution Time and Ranking Performance of Bα-DSS

This experiment serves two purposes: (i) Evaluating the execution time of the ELECTRE III MCDM method compared with the TOPSIS MCDM method [35], with and without applying the skyline operator and the α-dominance. (ii) Assessing the ranking performance of the ELECTRE III compared with the TOPSIS, using the Kendall Tau Distance (KTD) [50].

We employed TOPSIS due to its ability to identify the best α-dominant skyline service alternatives by minimizing the distance to the positive ideal solution (i.e., service) and maximizing the distance to the negative-ideal solution. TOPSIS was used for benchmarking and ranking purposes according to [22].

The initial set of services was expanded to 950 services. The search space was reduced to 500 services by applying the skyline operator and the α-dominance.

As a result of using the skyline and the α-dominance methods, we observed a reduction in the execution time of both the ELECTRE III and TOPSIS, as depicted in figures 7 and 8.

These results demonstrate that employing the skyline and the α-dominance methods is crucial for pruning the dominated services before performing the ranking step through the MCDM method.

Reducing the search space of the services allows us to operate only on the most relevant services, simplifying the selection process.

To assess the rankings produced by the ELECTRE III and TOPSIS MCDM methods, we enlisted the help of environmental experts from the Observatory of Sahara and Sahel (OSS), our socio-economic partner in the PREDICAT project. We proposed 500 ratings of the service candidates (i.e., the 1^{st}-ranked services) to these experts. They were divided into four groups, with each group evaluating approximately 125 ranked service alternatives.

A cross-validation process was then conducted among the different groups. We measured the Kendall Tau Distance (KTD) coefficients between the services ranked by the experts and those ranked by ELECTRE III and TOPSIS. Our analysis revealed that the KTD rankings produced by ELECTRE III outperformed those produced by TOPSIS.

Specifically, for the majority of the concepts (i.e., Temperature, Humidity, Wind_Speed, Drought_Factor, and Wind_Direction), the KTD measures for ELECTRE III were lower than those for TOPSIS, as shown in Figure 9. A decrease in the KTD measure indicates that the ranked lists produced by ELECTRE III are more similar to those proposed by the experts.

6.4 Experiment 3: Complexity Assessment of the Optimal Service Selection Using the Bα-DSS

The aim of this experiment is to evaluate the execution time of the Bα-DSS method for the selection of the optimal services compared with the Penalty-based Genetic Algorithm (GA) approach. The Genetic Algorithm (GA) generates a population of solutions, typically using random initialization, which are then evaluated based on a fitness function.

We employed the Penalty-based GA approach proposed in [24], which penalizes infeasible solutions that violate constraints. Table 3 outlines the parameter settings for the penalty-based GA, which were determined through experimentation on randomly generated test problems.

In our context, each chromosome in the GA represents an executable service composition. An executable service is formed by replacing each gene of the chromosome. Each gene in the chromosome corresponds to an index pointing to an array of potential concrete services that can fulfill a given concept.

We measured the computational time for the following approaches: the Penalty-based GA and the Bα-DSS. For all the tests, we used the six quality dimensions (QoS and QoDS), cited above (in sub-Section 5.2, A). We varied the number of the candidate instance services for each concept.

For about 10 concrete services, the computation time of the Penalty-based GA tends to be linear, and almost constant. Then, we noticed, that the computational time for the Penalty-based GA increases exponentially as the number of service instances grows. Otherwise, the computational time in the GA rises exponentially, as the number of feasible candidate services and the number of concepts grow.

We, then, compared the computational time of the Bα-DSS approach with Penalty-based GA. We noticed that the computational time is narrowed significantly, as the number of service instances for a given concept increases, in the Bα-DSS approach. The use of the skyline operator and the α-dominance methods reduces the search space and saves time in the outranking process of the dominant solutions. Figure 11 depicts the necessary computational time for the Bα-DSS and the Penalty-based GA approaches.

6.5 Experiment 4: Scalability Assessement of the Optimal Service Selection Using the Bα-DSS

This experiment aims to evaluate the scalability of the Bα-DSS method for the selection of the optimal services, compared with the Penalty-based Genetic Algorithm (GA) approach. We varied a collection of the dataset of the service candidates for the selection process. This collection comprises datasets of varying sizes: 100K, 250K, 500K, 750K, and 1000K.

Figure 10 depicts the varied dataset size of the candidate services and the necessary computation time for the execution of the Bα-DSS and the Penalty-based GA. According to the results, we noticed that as the size of the dataset of the candidate services increases, the computational time decreases, with the application of our Bα-DSS approach.

Therefrom, the computation time decreases due to the pruning process of the candidate services that are not likely to be part of the optimal solutions of QoS services, thanks to the skyline and the α-dominance methods, which allowed to gain/save time on the selection process overall compared to the GA one.

Therefrom, the information overload issue related to the evolution of services and the need for context-specific selection, are addressed through our knowledge-driven solution (Bα-DSS) acting as a decision-maker and ensuring recommendations by filtering irrelevant services.

7 Threats to Validity

The final results of our proposal garnered significant attention from experts, as they have the potential to reduce considerably the initial set of services and the response-time, thanks to our (Bα-DSS) approach. Moreover, experts from the OSS conducted several evaluative tests, as described in Section 6, to evaluate the outcomes of our proposed automated knowledge-driven approach for optimal selection of services. In addition, using our framework, experts can apply weights based on the actual circumstances. The proposed framework is currently in a prototype stage, developed to meet the requirements specified by PREDICAT experts.

When evaluating the performance and quality of our framework, it is essential to consider the threats to the validity of the findings. Specifically, we need to assess the potential inaccuracies in the framework's outcomes, i.e., the relationship between the framework's results and reality. If the number of services significantly increases, the MESOn ontology may no longer provide adequate and timely responses for quality assessment. This could also affect the availability of information on service quality.

8 Conclusion and Future Work

This paper proposes a novel approach that combines (i) a dedicated ontology to define and assess data sources quality dimensions along with their associated inferences, with (ii) ELECTRE III MCDM method performing fuzzy outranking, to optimize the selection of services participating in service composition.

Additionally, our approach considers (iii) the knowledge related to the quality levels of both services (QoS) and environmental data sources (QoDS) in the outranking process.

To evaluate our framework, we conducted a series of experiments in collaboration with experts from OSS. Through these experiments, we assessed the effectiveness and relevance of our proposed approach. Our findings indicate that our framework offers:

1. Reasonable reasoning time for assessing data source quality, ensuring that data source quality cannot change within this timeframe.
2. Reduction of the execution time of the ELECTRE III method through the application of the skyline and the α-dominance methods. Furthermore, our results demonstrate that the ELECTRE III MCDM method outperforms the TOPSIS MCDM method in the ranking process and selection of optimal services.
3. A reduction of the computational time of the Bα-DSS approach compared with Penalty-based GA, as the number of service instances for a given concept increases.
4. The scalability analysis along with a variation of the dataset size of the services candidates showed a decrease of the computational time due to the pruning process of the irrelevant services.

As a future research, we intend to rely on the application of the reinforcement learning algorithms to select optimal candidates services. Furthermore, as part of the quality of services, we want to improve our framework with business non-functional qualities (e.g., consequences for variations, failure reporting, etc.), as future work.

However, since these QoS are not computable, we can rely on a subjective approach that allows evaluating these QoS, based on experts' ratings and feedback.

Acknowledgments

The authors acknowledge the support of the European Commission for funding the InnoRenew project (Grant Agreement #739574) under the Horizon 2020 Widespread-Teaming program, as well as the support of the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Regional Development Fund).

This work was financially supported by the “PHC Utique” program of the French Ministry of Foreign Affairs and Ministry of higher education and research and the Tunisian Ministry of higher education and scientific research in the CMCU project number 17G1122.

The authors acknowledge the European Commission for funding the InnoRenew CoE project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming program and the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Union of the European regional Development Fund).

References

1. **Abourezq, M., Idrissi, A. (2014).** Introduction of an outranking method in the cloud computing research and selection system based on the skyline. Proceedings of the IEEE 18th International Conference on Research Challenges in Information Science, pp. 1–12. DOI: 10.1109/RCIS.2014.6861067.
2. **Albertoni, R., Isaac, A. (2015).** Data quality vocabulary (DQV). W3C Interest Group Note. www.w3.org/TR/vocab-dqv/.
3. **Albertoni, R., Isaac, A. (2020).** Introducing the data quality vocabulary (DQV). Semantic Web, Vol. 12, No. 1, pp. 81–97. DOI: 10.3233/sw-200382.
4. **Alrifai, M., Skoutas, D., Risse, T. (2010).** Selecting skyline services for QoS-based web service composition. Proceedings of the 19th international conference on World wide web, pp. 11–20. DOI: 10.1145/1772690.1772693.

5. **Ardagna, D., Pernici, B. (2007).** Adaptive service composition in flexible processes. IEEE Transactions on software engineering, Vol. 33, No. 6, pp. 369–384.

6. **Balioti, V., Tzimopoulos, C., Evangelides, C. (2018).** Multi-criteria decision making using topsis method under fuzzy environment. Application in Spillway Selection. EWaS3 2018, Vol. 2, No. 11, pp. 637. DOI: 10.3390/proceedings2110637.

7. **Batini, C., Scannapieco, M. (2016).** Data and information quality: Dimensions, principles and techniques. DOI: 10.1007/978-3-319-24106-7.

8. **Ben-Ellefi, M., Bellahsene, Z., Breslin, J. G., Demidova, E., Dietze, S., Szymański, J., Todorov, K. (2018).** RDF dataset profiling – a survey of features, methods, vocabularies and applications. Semantic Web, Vol. 9, No. 5, pp. 677–705. DOI: 10.3233/sw-180294.

9. **Benouaret, K., Benslimane, D., Hadjali, A. (2011).** On the use of fuzzy dominance for computing service skyline based on QoS. Proceedings of the IEEE International Conference on Web Services, pp. 540–547. DOI: 10.1109/ICWS.2011.93.

10. **Benouaret, K., Benslimane, D., Hadjali, A., Barhamgi, M., Maamar, Z., Sheng, Q. Z. (2014).** Web service compositions with fuzzy preferences: A graded dominance relationship-based approach. ACM Transactions on Internet Technology, Vol. 13, No. 4, pp. 1–33. DOI: 10.1145/2576231.

11. **Bizer, C. (2007).** Quality-driven information filtering in the context of web-based information systems. Freie Universität Berlin. DOI: 10.17169/REFUBIUM-14260.

12. **Borzsony, S., Kossmann, D., Stocker, K. (2001).** The skyline operator. Proceedings of the 17th International Conference on Data Engineering, pp. 421–430. DOI: 10.1109/icde.2001.914855.

13. **Bosc, P., Pivert, O. (1995).** SQLf: A relational database language for fuzzy querying. IEEE Transactions on Fuzzy Systems, Vol. 3, No. 1, pp. 1–17. DOI: 10.1109/91.366566.

14. **Brans, J. P. (1982).** L'ingénierie de la décision: l'élaboration d'instruments d'aide a la décision. UniversitéLaval, Faculté des sciences de l'administration.

15. **Chen, L. (2014).** Ensuring reliability and QoS optimizing for web service composition. Proceedings of the 10th International Conference on Computational Intelligence and Security, pp. 510–513. DOI: 10.1109/cis.2014.86.

16. **Chouiref, Z., Belkhir, A., Benouaret, K., Hadjali, A. (2016).** A fuzzy framework for efficient user-centric web service selection. Applied Soft Computing, Vol. 41, pp. 51–65. DOI: 10.1016/j.asoc.2015.12.011.

17. **Dai, G., Zhu, Q. (2013).** Using skyline and dominance relationship for web services ranking. Journal of Computational Information Systems, Vol. 9, pp. 3977–3984. DOI: 10.12733/jcis5895.

18. **Debattista, J., Lange, C., Auer, S. (2014).** daQ, an ontology for dataset quality information. CEUR Workshop Proceedings, Vol. 1184, pp. 1–8.

19. **Dorfeshan, Y., Tavakkoli-Moghaddam, R., Jolai, F., Mousavi, S. M. (2021).** A new data-driven and knowledge-driven multi-criteria decision-making method. Journal of AI and Data Mining, Vol. 9, No. 4. DOI: 10.22044/jadm.2021.10803.2218.

20. **Farias-Lóscio, B., Stephan, E. G., Purohit, S. (2016).** Data on the web best practices: Dataset usage vocabulary. W3C Working Group Note. www.w3.org/TR/vocab-duv/.

21. **Fielding, R. T. (2000).** Architectural styles and the design of network-based software architectures. University of California, Irvine.

22. **Forestal, R. L., Pi, S. (2021).** A hybrid approach based on ELECTRE III-genetic algorithm and TOPSIS method for selection of optimal COVID-19 vaccines. Journal of

Multi-Criteria Decision Analysis, Vol. 29, No. 1-2, pp. 80–91. DOI: 10.1002/mcda.1772.

23. **Frank, M., Walker, J. (2016).** User centred methods for measuring the value of open data. The Journal of Community Informatics, Vol. 12, No. 2. DOI: 10.15353/joci.v12i2.3221.
24. **Grati, R., Boukadi, K., Ben-Abdallah, H. (2014).** QoS based resource allocation and service selection in the cloud. Proceedings of the 11th International Conference on e-Business, pp. 249–256.
25. **Jarke, M., Jeusfeld, M. A., Quix, C., Vassiliadis, P. (1999).** Architecture and quality in data warehouses: An extended repository approach. Information Systems, Vol. 24, No. 3, pp. 229–253. DOI: 10.1016/s0306-4379(99)00017-4.
26. **Jauhari, A., Mufarroha, F. A. (2020).** Smart mobile application for decision support systems on determination of resident in dormitory. Jurnal Ilmiah Kursor, Vol. 10, No. 3. DOI: 10.21107/kursor.v10i3.236.
27. **Jayawardene, V., Sadiq, S., Indulska, M. (2015).** An analysis of data quality dimensions. Technical Report, The University of Queensland.
28. **Kaufer, F., Klusch, M. (2006).** WSMO-MX: A logic programming based hybrid service matchmaker. Proceedings of the European Conference on Web Services, pp. 161–170. DOI: 10.1109/ECOWS.2006.39.
29. **Klusch, M., Fries, B., Sycara, K. (2009).** OWLS-MX: A hybrid semantic web service matchmaker for OWL-S services. Journal of Web Semantics, Vol. 7, No. 2, pp. 121–133. DOI: 10.1016/j.websem.2008.10.001.
30. **Koppen, M., Vicente-Garcia, R. (2004).** A fuzzy scheme for the ranking of multivariate data and its application. Proceedings of the IEEE Annual Meeting of the Fuzzy Information, Vol. 1, pp. 140–145. DOI: 10.1109/NAFIPS.2004.1336266.
31. **Kumar, R. R., Kumari, B., Kumar, C. (2020).** CCS-OSSR: A framework based on hybrid MCDM for optimal service selection and ranking of cloud computing services. Cluster Computing, Vol. 24, No. 2, pp. 867–883. DOI: 10.1007/s10586-020-03166-3.
32. **Lebo, T., Sahoo, S., McGuinness, D. (2012).** PROV-O: The PROV ontology. W3C Recommendation. www.w3.org/TR/prov-o/.
33. **Liangzhao, Z., Benatallah, B., Ngu, A. H. H., Dumas, M., Kalagnanam, J., Chang, H. (2004).** QoS-aware middleware for web services composition. IEEE Transactions on Software Engineering, Vol. 30, No. 5, pp. 311–327. DOI: 10.1109/TSE.2004.11.
34. **Maali, F., Erickson, J., Archer, P. (2014).** Data catalog vocabulary (DCAT). W3C recommendation. www.w3.org/TR/vocab-dcat-3/.
35. **Olson, D. L. (2004).** Comparison of weights in TOPSIS models. Mathematical and Computer Modelling, Vol. 40, No. 7-8, pp. 721–727. DOI: 10.1016/j.mcm.2004.10.003.
36. **Ouadah, A., Benouaret, K., Hadjali, A., Nader, F. (2015).** Combining skyline and multi-criteria decision methods to enhance web services selection. Proceedings of the 12th International Symposium on Programming and Systems (ISPS), pp. 1–8. DOI: 10.1109/ISPS.2015.7244975.
37. **Ouadah, A., Benouaret, K., Hadjali, A., Nader, F. (2015).** SkyAP-S3: A hybrid approach for efficient skyline services selection. Proceedings of the IEEE 8th International Conference on Service-Oriented Computing and Applications, pp. 18–25. DOI: 10.1109/SOCA.2015.22.
38. **Ouadah, A., Hadjali, A., Nader, F. (2018).** A hybrid MCDM framework for efficient web services selection based on QoS. Proceedings of the International Conference on Applied Smart Systems, pp. 1–6. DOI: 10.1109/icass.2018.8652037.
39. **Papadias, D., Tao, Y., Fu, G., Seeger, B. (2003).** An optimal and progressive algorithm for skyline queries. Proceedings of the

ACM SIGMOD International Conference on Management of Data, pp. 467–478. DOI: 10.1145/872757.872814.

40. **Pipino, L. L., Lee, Y. W., Wang, R. Y. (2002).** Data quality assessment. Communications of the ACM, Vol. 45, No. 4, pp. 211–218. DOI: 10.1145/505248.506010.

41. **Polska, O. V., Kudermetov, R. K., Shkarupylo, V. V. (2021).** An approach web service selection by quality criteria based on sensitivity analysis of MCDM methods. Radio Electronics, Computer Science, Control, , No. 2, pp. 133–143. DOI: 10.15588/1607-3274-2021-2-14.

42. **Rodriguez-Mier, P., Pedrinaci, C., Lama, M., Mucientes, M. (2016).** An integrated semantic web service discovery and composition framework. IEEE Transactions on Services Computing, Vol. 9, No. 4, pp. 537–550. DOI: 10.1109/tsc.2015.2402679.

43. **Roy, B. (1991).** The outranking approach and the foundations of electre methods. Theory and Decision, Vol. 31, No. 1, pp. 49–73. DOI: 10.1007/bf00134132.

44. **Serrai, W., Abdelli, A., Mokdad, L., Hammal, Y. (2017).** Towards an efficient and a more accurate web service selection using MCDM methods. Journal of Computational Science, Vol. 22, pp. 253–267. DOI: 10.1016/j.jocs.2017.05.024.

45. **Serrai, W., Abdelli, A., Mokdad, L., Serrai, A. (2018).** How to deal with QoS value constraints in MCDM based web service selection. Concurrency and Computation: Practice and Experience, Vol. 31, No. 24. DOI: 10.1002/cpe.4512.

46. **Sun, L., Ma, J., Zhang, Y., Dong, H., Hussain, F. K. (2016).** Cloud-FuSeR: Fuzzy ontology and MCDM based cloud service selection. Future Generation Computer Systems, Vol. 57, pp. 42–55. DOI: 10.1016/j.future.2015.11.025.

47. **Tan, K. L., Eng, P. K., Ooi, B. C. (2001).** Efficient progressive skyline computation. Proceedings of the 27th International Conference on Very Large Data Bases, pp. 301–310.

48. **Wang, R. Y., Strong, D. M. (1996).** Beyond accuracy: What data quality means to data consumers. Journal of Management Information Systems, Vol. 12, No. 4, pp. 5–33. DOI: 10.1080/07421222.1996.11518099.

49. **Wang, X., Vitvar, T., Kerrigan, M., Toma, I. (2006).** A QoS-aware selection model for semantic web services. Lecture Notes in Computer Science, pp. 390–401. DOI: 10.1007/11948148_32.

50. **Wauthier, F., Jordan, M., Jojic, N. (2013).** Efficient ranking from pairwise comparisons. Proceedings of the 30th International Conference on Machine Learning, Vol. 28, No. 3, pp. 109–117.

51. **Xu, J., Guo, L., Zhang, R., Zhang, Y., Hu, H., Wang, F., Pei, Z. (2017).** Towards fuzzy QoS driven service selection with user requirements. Proceedings of the International Conference on Progress in Informatics and Computing, pp. 230–234. DOI: 10.1109/PIC.2017.8359548.

52. **Youssef, A. E. (2020).** An integrated MCDM approach for cloud service selection based on TOPSIS and BWM. IEEE Access, Vol. 8, pp. 71851–71865. DOI: 10.1109/ACCESS.2020.2987111.

53. **Yu, Q., Bouguettaya, A. (2010).** Computing service skylines over sets of services. Proceedings of the IEEE International Conference on Web Services, pp. 481–488. DOI: 10.1109/ICWS.2010.48.

54. **Yu, Q., Bouguettaya, A. (2010).** Foundations for Efficient Web Service Selection. Springer US. DOI: 10.1007/978-1-4419-0314-3.

55. **Zadeh, L. A. (1965).** Fuzzy sets. Information and Control, Vol. 8, No. 3, pp. 338–353. DOI: 10.1016/s0019-9958(65)90241-x.

56. **Zou, H., Zhang, L., Yang, F., Zhao, Y. (2010).** A web service composition algorithmic method based on topsis supporting multiple

decision-makers. Proceedings of the 6th World Congress on Services, pp. 158–159. DOI: 10.1109/services.2010.110.

57. **Zulqarnain, R., Saeed, M., Ahmad, N., Dayan, F., Ahmad, B. (2020).** Application of topsis method for decision making. International Journal of Scientific Research in Mathematical and Statistical Sciences, Vol. 7, No. 2, pp. 76–81.

Article received on 07/01/2023; accepted on 03/03/2024.
** Corresponding author is Hela Taktak.*