

---

# Managing Web-based Information Resources Under Uncertainty: A Probabilistic Approach

---

Asma Omri\* and Mohamed-Nazih Omri

*MARS Research Laboratory, University of Sousse, Tunisia*  
*E-mail: omri.asmaaa@gmail.com; mohamednazih.omri@fsm.rnu.tn*  
*\*Corresponding Author*

Received 30 May 2023; Accepted 13 January 2024;  
Publication 22 February 2024

## Abstract

“Uncertainty” is related to working with inaccurate data, imprecise and incomplete information, and unreliable results that can lead to irrational decisions. Several approaches to managing uncertain data on the Web have been proposed in the literature to resolve this problem. These approaches have failed to find the solution to this problem with accuracy and performance. Our study aims to propose a new probabilistic approach to manage Web information resources in an uncertain large-scale cloud environment. Our approach is based on three main steps: (1) modelling uncertain Web resources, (2) computing HTTP request model, and (3) interpretation and evaluation of uncertain Web resources in a context of classic hypertext navigation. The experimental study shows that the analysis of the execution time necessary for the composition of the services, by our approach, is negligible, compared to that of the other studied approaches. The algorithm that deals with the impact of the variation in the number of nodes, which

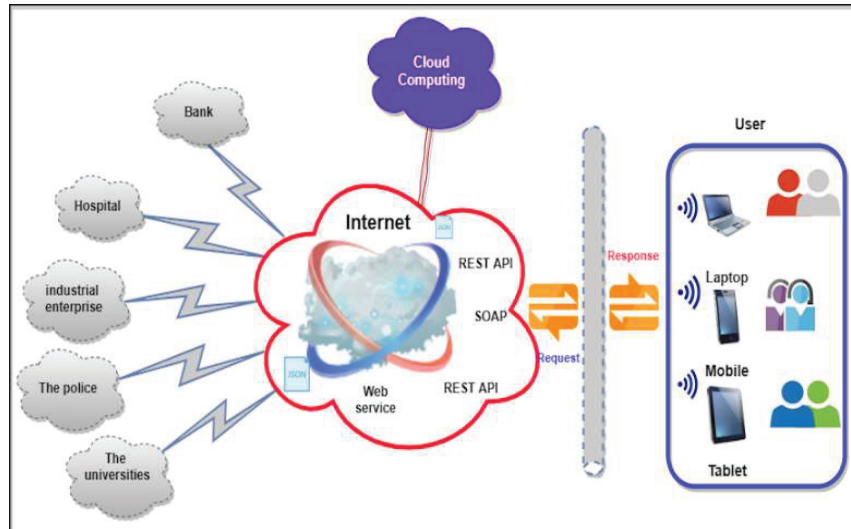
we have proposed, has also been evaluated and checks all the possibilities in polynomial time and can adapt to many possibilities of multiplexing of values.

**Keywords:** Web resources, data management, uncertainty, probabilistic approach.

## 1 Introduction

The current evolution of intelligent objects, with new technologies, has made it possible to develop new perspectives in the field of the Web and expert systems. These modern systems have enabled individuals, organizations and connected objects to produce and publish huge amounts of data to the Web through application programming interfaces (APIs) and public endpoints. This development has enabled users to communicate, receive and share data [1, 2]. The Internet of Things (IoT), for example, is an emerging technology that embodies a groundbreaking resolution enabling the connection of billions of physical devices in the digital world using various networks with heterogeneous objects [3, 4]. Data retrieved from the Internet is not immune to errors and uncertainties, and may contain contradictions resulting from inherently uncertain processes such as data integration and extraction [5]. This can be due to a variety of factors, including but not limited to: human error, technical issues and inconsistencies in data sources. Therefore, it is important to approach data retrieved from the Internet with a critical eye and to verify the accuracy and reliability of the information before using it. These uncertainties arise from different sources, such as minor differences in measurements of physical events or different information provided describing the same entity. Based upon a review of the literature, some scholars [6–9] state that finding solutions to deal with such uncertainty becomes an urgent challenge for the Web community, due to its current state, in order to allow users to select a unique representation for a given entity.

Figure 1 presents a typical scenario that shows the different actors involved in researching the composition of Web services and resources to meet increasing user demand. This scenario shows several users such as banks, police, and hospitals accessing a flexible set of information resources to quickly resize any application. The user of such a system can insert or modify data in the shared databases and, therefore, this data becomes uncertain. Suppose a user tries to find an answer to the following query: “What is the name of the director of the University of Paris?” When browsing the Web



**Figure 1** Example of motivation.

pages, the user enters the name of the person in a search engine, chooses one of the links offered as results by this search engine, and continues to browse until he reaches the answer. This method provides only one representation at a time since the user is forced to select a single representation among several representations corresponding to the same entity. Now consider the same request to a traditional Website (API). The preprocessing process typically filters out results identified as incorrect during the onboarding process and only provides the user with the information deemed correct. However, the main objective is to take into account the uncertain data of the Web which must, therefore, be explicitly modelled and rigorously described. This way, the user can accurately understand and interpret the results returned by these services and use them correctly. The importance of handling the uncertainty of the results returned by Web services becomes important when the output admits multiple possible responses or when the associated probability value is not provided. This study aims to discuss the uncertain probabilistic Web resources by considering how they can be used to represent Web data and the complexity of the evaluation of user queries.

To improve the quality of data produced on the Web, emphasis is placed on uncertainty in the presence of the following problems: (i) evaluation of the interpretation of a composition is manifestly insoluble, and the algorithms

proposed to simplify the possible interpretation present an exponential complexity; and (ii) the number of possible Websites can be extremely large, and hence evaluation of requests becomes difficult, if not impossible. In this paper, we position the problem of managing information resources in large-scale cloud environments of uncertainty, in step with the evolution of new technologies. Therefore, we propose a new model to define, compute, and interpret uncertain Web resources in the context of classical hypertext navigation and evaluation of data queries. The main contributions of this paper are manifold. First, it contributes to research in information systems by modelling concepts of uncertain Web resources, based on a probabilistic XML model, where each resource can be interpreted by several possible representations, with a degree of probability. Based on this, we propose an approach to evaluate user requests. Second, this paper contributes new insights into the literature by proposing a solution allowing Internet users to manage the uncertainty of the data when they navigate in hypertext. Considering this uncertainty has improved the way we deal with the huge amount of information available on the Web. Third, this paper provides new insights into the Web resources by proposing a new language to represent uncertain resources and their implementation as well as a tool for its validation. We propose an algebra for the interpretation and evaluation of data requests in the case of compositions of uncertain resources and an algorithm for the evaluation of requests, recovery, and calculation of the uncertainty associated with responses to user requests.

The remainder of this paper is organized as follows. In Section 2, we discuss the main works presented in the literature. In Section 3, we present the concept of a probabilistic Web resource and define its semantics. We are also introducing the proposed approach and discussing in detail a new programmatic representation of our probabilistic model. Section 4 describes the experiments and the obtained results. Section 5 presents a discussion around the proposed work. In the last section, we summarize the work.

## **2 Related Work**

### **2.1 Uncertain Information**

Information is called uncertain if one does not know whether it is true or false. Elementary information is a proposition or assertion that an event has occurred. It is modelled by a subset of possible values with an uncertainty

marker. This marker can be numeric or linguistic. Markers are always numbers (probability) or symbolic modalities (certain, possible, probable). For example, it is certain that Georges arrives late or the probability that Georges arrives late is 0.7.

## **2.2 Related Work**

The problem of modelling, discovery, invocation, selection, and composition of Web services based on the quality of service and handling of data uncertainty has received considerable attention in the Web services community. For example, to overcome the problem of modelling Web services, several works were carried out to provide reliable solutions. However, studies related to uncertainty problems have not succeeded in proposing reliable and inexpensive solutions, especially in the field of intelligent and expert systems. In this section, we present the main approaches proposed in the literature regarding the problems due to uncertainty in Web services, expert and smart systems and the semantic Web.

## **2.3 Uncertainty Issues on the Standard Web**

In [10], the authors propose an approach to manage the of data returned by Web services with uncertain data. The approach of [11] presents a new classification model that uses the notation provided by different users to filter attacks; it highlights the interest in risk analysis by managing uncertainties, which can help make appropriate decisions on the actions necessary in the event of ignorance of the status of the risk system. The authors of [12] propose a probabilistic approach to manage the uncertainty associated with the interpretation of DaaS services. Another interesting work is that of [13] that proposes a new model with a new algebra to interpret these resources in an uncertain environment by integrating the concept of uncertain representational state transfer (REST)ful resource concept. In the work of [14] and [15], the authors propose a possibilistic approach to solve the problem of interrogating cloud services with possibilities for taking data into account. Two types of invocation have been proposed: a conventional invocation and a possibilistic invocation in addition to a new algebra to solve the problem of composition. The work of [16], has also attempted to create a theoretical framework for the modelling, interpretation, manipulation, and evaluation of uncertain Web data. The approach proposed by [17] offers a solution to

solve the problem of selecting Web services with non-deterministic quality of services (QoS).

## **2.4 Uncertainty in Intelligent and Expert Systems**

In the work presented by [18], the authors propose a model to create a panoramic view of the composition of cloud services sensitive to QoS from an IT intelligent point of view. Another interesting work that also deserves to be cited is that of [19], which implements a recommendation for tourism based on people who share photos, tags and geographic information on social networks. The approach of [20] offers a solution that combines proactive and reactive reasoning in a single architecture that enables intelligent Web services for WoT. This solution highlights the reasoning and decision-making layer for semantic representation and reasoning on complex situations with uncertain information. Another attempt based on fuzzy logic to predict the final grade of students was proposed in [21]. This proposal takes into account the internal and external evaluation scores as well as the scores of a living voice to predict the final grade.

## **2.5 Uncertainty in Semantic Web**

The article in [22] describes a coherent and complete probabilistic framework for the Semantic Web. This model provides a means to represent probabilistic knowledge and provide Web services such as plausible inference and Bayesian learning. To handle uncertainty in the Semantic Web and its applications, many researchers [23] have proposed extending OWL and description logic (DL) formalisms with special mathematical frameworks. Drawing on the results of the literature, they created a fuzzy extension to OWL called Fuzzy OWL. Fuzzy OWL can capture imprecise and vague knowledge. Markov logic has been successfully applied to problems in entity resolution, link prediction, information extraction, and others, and is the basis of the open-source Alchemy system. In this paper [24], they describe the representation of Markov logic and provide an overview of current inference and learning algorithms for it. They started with some basics about first order logic and Markov networks. They developed a series of learning and inference algorithms for this, and applied them successfully in a number of areas. In [25], they proposed a series of higher order probabilistic models to manage Web data and showed that these models allow them to take into account the intrinsic uncertainty of this data, while providing a probabilistic model that allows reasoning about the data. They proposed two types of

analyses, one based on the visual interpretation of heat maps, the other based on the automatic determination of change points by means of a procedure that we introduce. The results obtained with these two analyses partially overlap.

By analyzing the results obtained from the series of criteria developed for different approaches, one can notice the change in performance from one approach to another. The result observed in Table 1 shows that our performance is better in several criteria. The proposed approach, compared to the others, offers the best result thanks to the proposed method for representing uncertain data which is based on probabilistic XML and which is easily implementable on the browser and which generates a better representation of data and queries. The more meaningful and expressive the representation of uncertain data, the more correct the calculation of the degree of uncertainty. According to the results presented in Table 1, we can notice the advantages of the suggested approach in terms of complexity since this model does not allow managing an exponential number of uncertain Web resources. In addition, the proposed programming language and validator allows us to implement real cases and to use hypertext browsers to test our approach. The results obtained allow us to conclude the advantage of probabilistic XML, programming language, algebraic equations and GET algorithms to give more importance to all the uncertain information that characterizes the Web today and to improve the quality of the response.

### 3 Establishment of Model

In this section, we present our three main contributions, around which our approach is articulated: modeling of uncertain resources, proposed HTTP request model, and the proposed composition model of uncertain Web resources.

#### 3.1 Semantic of Uncertain Web Resource

Each *URL* identifies a single resource, which within a server is of the following form:  $\langle A_i, V_{ij} \rangle$ , where  $V_{ij}$  is in the form  $\{(v_{ij}, p_{ij})\}$ . The set of values  $v_{ij}$  expresses at which point  $V_{ij}$  represents  $A_i$  and  $p_{ij}$  is the associated probability value to  $v_{ij}$ . Since several representations of a resource cannot coexist at the same *URL*, these representations are mutually exclusive. Then, the probability associated with each value  $v_{ij}$ , which takes its values in the set  $\{\text{String, Number, Object, Array, False, null}\}$ , is  $p_i \in ]0, 1]$ . So, in our proposed formalism a resource  $R$  is then given by the next definition.

**Table 1** Comparison between some related works

Approach	Type of Service	Data Type	Uncertainty Approach	Complexity	Response Time	Number of Possible Worlds		Navigateur	Parser	Generation of Impossible Cases		Programming Language
						High	Exponential			Not compatible	No	
Pierre [2016]	RESFULL	BD	Probabilistic	High	Medium	High	Exponential	Not compatible	No	No	No	No
Soumaya [2014]	SOAP	BD	Probabilistic	High	Medium	High	Exponential	Not compatible	No	No	No	No
Malki [2015]	SOAP	Semantic	Probabilistic	High	Medium	High	Exponential	Not compatible	No	No	No	No
Soura [2019]	RESFULL	BD	Probabilistic	High	Medium	High	Exponential	Not compatible	No	Yes	No	No
Asma [2018]	SOAP	BD	Possibilistic	High	Medium	High	Exponential	Not compatible	No	No	No	No
Our approach [2020]	RESFULL	BD	XML Probabilistic	Medium	Medium	Medium	Medium	Not compatible	Yes	No	No	Yes



**Definition 3.1.** Let  $URL_R$  be the path to take, in a classic Internet browser, to access the resources  $R$  contained in the Web. Let  $A_i, i \in [1, n]$ , be the set of attributes characterizing the resource  $R$  and  $V_{ij}, j \in [1, m]$ , their values respectively. We define a Web resources  $R$  by the following equation:

$$R = \{U\ RL_R, \langle A_i, V_{ij} \rangle\}. \quad (1)$$

In the context of uncertain data, the semantics of uncertain Web resources include how a resource is represented. In our work, uncertain Web resources adopt the possible semantics of the Web based on the theory of possibilities [12]. Thus, an uncertain resource can have several possible representations which can be potentially and individually interpreted as true. These possibilities can be interpreted as a set of possible worlds  $w_i, i \in [1, n]$ , to which will be associated a degree of probability  $p_i, i \in [1, n]$ , indicating to what extent these possibilities represent the Web resource in question. We call them possible Websites in which the data is considered certain. As we mentioned above, we will adopt the principle of *REST* resources by adding new hypotheses to define the notion of uncertain ones. These assumptions are as follows:

- Due to the principles presented in *REST*, several representations of the same resource *URL* cannot coexist and must, therefore, be mutually exclusive.
- Uncertainty should only affect representations of resources; hence all other data is considered certain in a given possible representation.

Therefore, two types of resources (attributes) have become distinguishable: certain and uncertain resources. Certain resources follow the standard JSON (JavaScript Object Notation) model (Figure 2), while the uncertain ones follow our proposed JSON probabilistic model.

In an uncertain environment, an attribute  $A_i$ , which characterizing an  $R^P$ , can take different values  $V_{ij}$ , as a simple or a complex value  $V_{ij}^{Ind}$  or  $V_{ij}^{Mux}$ . When looking for a possible answer (a resource) to a given request, all existing JSON documents will be searched. Each representation, corresponding to a response returned, will be automatically scanned by the control block to find the two nodes *Mux* and *Ind*. An overall degree of probability associated with this response will therefore be calculated, as a function of these last two nodes, indicating to the user the degree of uncertainty of this response. Thus, the user will therefore be warned of the reliability of the response returned by the system. In the next equations, the couple of values  $\langle A_i, V_{ij} \rangle$  represents the set of possible representations of  $R^P$ .

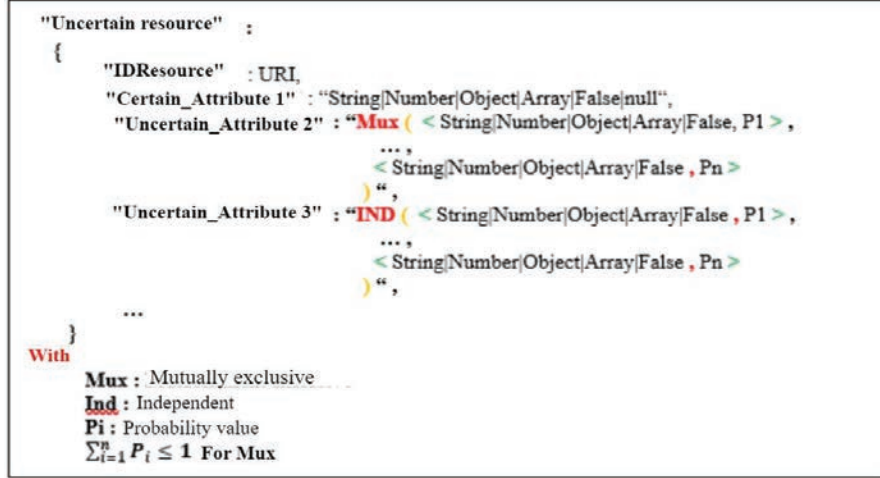


Figure 2 Probabilistic JSON model.

**Definition 3.2.** Let  $A_i, i \in [1, n]$ , be an attribute characterizing an  $R^P$  and  $V_{ij}^{Ind}, j \in [1, m]$ , its possible value, based on *Ind* node, represented by a set of couple of values  $(v_{ij}, p_{ij})$ , we define an independent type  $V_{ij}^{Ind}$  by the following equation:

$$V_{ij}^{Ind} = Ind(\langle v_{i1}, p_{i1} \rangle, \dots, \langle v_{ik}, p_{ik} \rangle). \quad (2)$$

**Definition 3.3.** Let  $A_i, i \in [1, n]$ , be an attribute characterizing an  $R^P$  and  $V_{ij}^{Mux}, j \in [1, m]$ , its possible value, based on *Mux* node, represented by a set of couple of values  $(v_{ij}, p_{ij})$ , we define mutually exclusive type  $V_{ij}^{Mux}$  by the following equation:

$$V_{ij}^{Mux} = Mux(\langle v_{i1}, p_{i1} \rangle, \dots, \langle v_{ik}, p_{ik} \rangle). \quad (3)$$

After having defined the semantics of the two types of nodes *Ind* and *Mux* and given their expressions, we can define the notion of uncertain resource that we adopted in our approach.

**Definition 3.4.** Let  $A_i, i \in [1, n]$ , be an attribute characterizing an  $R^P$  and  $V_{ij}, j \in [1, m]$ , its possible value, based on *Mux* node. Let  $URL_R$  be the path to take, in a classic Internet browser, to access the resources  $R$  contained in the Web. We define an uncertain resource  $R^P$  by the following equation:

$$R^P = \{U\ RL_R, \langle A_i, V_{ij} \rangle\}. \quad (4)$$

### 3.2 Uncertain Web Resource Formalism

To provide a means to manage these uncertain resources, we propose a formalism to represent them physically. In the following section, we present the different main parts of the proposed formalism.

Figure 3 shows the general architecture of our proposed formalism where a resource is represented by a rounded rectangle, colored black, and is characterized by its unique identifier URL. A certain resource is represented by

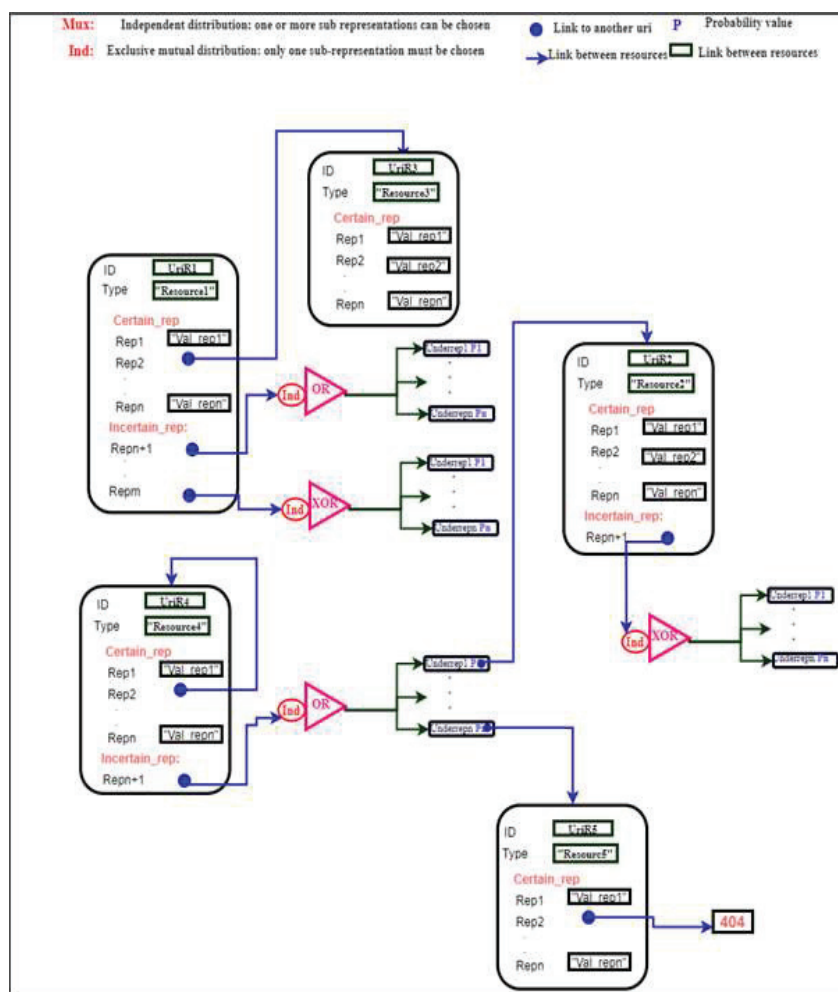


Figure 3 General architecture of the proposed formalism.

a simple rectangle of green color and an uncertain representation represented by a large blue dot. The symbol  $P$  represents the uncertainty value associated with a representation, it represents the probability value, in the interval  $]0,1[$ , associated with this value. These URL links are represented by fairly large dots and blue arrows which can indicate two different types of links: (i) a link to a new certain or uncertain resource, (ii) to an operator of type *Mux* or *Ind*. These two operators indicate the two types of uncertainties, which we have introduced and taken into consideration in our formalism, which were presented previously (see Equations (2) and (3)). In the following section, we will present the definitions of the degrees of probability associated with these two types of resource uncertainty values.

### 3.3 Independent and Mutually Exclusive Nodes

In XML probabilistic representation we have two types of nodes, namely the independent type *Ind* and the mutually exclusive type *Mux*. For the first type, if a resource property has multiple possible values, it can appear in separate representations or in the same representation. The second type of uncertainty requires the appearance of its values in separate representations. For the sake of simplicity in this approach, we consider that every possible representation of a given resource is represented according to the same model.

#### 3.3.1 Definitions of the probability measures

Next, we suppose that the choice of a possible Web resource  $w_i$ , for any children, is independent of any other choice of children of  $v$  or other distributive nodes. When looking for a possible answer (a resource) to a given request, all existing JSON documents will be searched. Each representation, corresponding to a returned response, will be automatically scanned, by the control block, to find the different forms of the data, at the node levels. The possible forms are three in number and are provided by the literature in the field: (i) the first form corresponds to data of type *Ind*, (ii) the second form corresponds to data of type *Mux* and (iii) the third and last form corresponds to data of type *Ind* and *Mux* at the same time. So, depending on the form found of the data, a degree of probability  $P^{Ind}$ , or  $P^{Mux}$ , or  $P^{Ind} * P^{Mux}$ , associated with this response will therefore be calculated and provided to the user indicating the uncertainty of this response, so that they will be warned about the reliability of the response received. These probability measures are given by the following definitions.

**Definition 3.5.** Let  $C_i$  be a subset of the children of a node  $v$  and  $P^v(w_i)$  be the probability of choosing a possible Web resource  $w_i$  for any children in  $C_i$ , of the node  $v$  of type *Ind*. We define the probability  $P^{Ind}(C_i, v)$ , of choosing a subset  $C_i$  of the children of the node  $v$ , by the following equation:

$$P^{Ind}(C_i, v) = \prod_{w_i \in C} P^v(w_i) \prod_{w_i \in \overline{C_i}} (1 - P^v(w_i)) \quad (5)$$

where  $\overline{C_i}, i \in [1, n]$  denotes all the children of the node  $v$  who are not in  $C_i$ .

**Definition 3.6.** Let  $C_i$  be a subset of the children of a node  $v$  and  $P^v(w_i)$  be the probability of choosing a possible Web resource  $w_i$  for any children, in  $C_i$ , of the node  $v$  of type *Mux*. We define the probability  $P^{Mux}(C_i, v)$ , of choosing a subset  $C_i$  of the children of the node  $v$ , by the following equation:

$$P^{Mux}(C_i, v) = P^v(W_i) \quad (6)$$

A node  $v$  of type *Mux* specifies the probabilities  $p^v(w_1), \dots, p^v(w_k)$  for its children  $w_1, \dots, w_k$ , respectively. The node  $v$  chooses at most one child  $w_i$  with the probability  $p^v(w_i)$ ; this choice is independent of the other distributive nodes.

We require that  $\sum_{i=1}^K P^v(w_i) \leq 1$ . The probability that  $v$  does not choose any of their children is  $1 - \sum_{i=1}^K P^v(w_i)$ . Its operating principle can be summarized based on the logical mathematical operation XOR.

### 3.4 Proposed Probabilistic JSON Model

We defined the resources and their associated data in terms of the JSON data model. However, these resources are always considered abstract entities. Before they can be communicated to a client over an HTTP connection, they must be serialized to a textual representation that can be included as an entity in an HTTP message body. As explained earlier, we rely on the standard JSON principle in this process. Hence, we must make some modifications that will make it applicable to the proposed model. The proposed probabilistic JSON model is illustrated in Figure 4.

In our approach,  $GET^P$  does not define a new *HTTP* method, since it is only a more complex notation of a standard *GET* with specific headers. The *GET* method acts as a standard *GET* with a specific *HTTP* header. We then define an *Uncertainty-Processing: 1* used to inform the user that we can accept the uncertain Web resource processing and that the response we send to it is not certain, and a *Degree-Uncertainty: 0.2* specifies the degree of

```

< Probabilistic JSON > ::= <value>|< probabilistic Value >

< probabilistic Value > ::= <object> | <array> | <boolean> | <string> | <number> |
<null>|< probabilistic object > | < probabilistic array > | < probabilistic boolean
> | < probabilistic string > | < probabilistic number >

<Array> ::= "[" [<probabilistic value >] {" , " < probabilistic value > } * "]"

<Object> ::= "{" [<probabilistic property >] {" , " < probabilistic property > } * "}"

< probabilistic Property > ::= <string> ":" Mux[Ind " (" "<" <value>, probability
value" " {" , " "<"<value>, probability value"> } * ")"

```

**Figure 4** Proposed portion of probabilistic JSON parser.

```

HTTP/1.1 200 OK
Date: Thu, 11 Jan 2007 14:00:36 GMT
Server: Apache/2.0.54 (Debian GNU/Linux) DAV/2 SVN/1.1.4
Uncertainty-Processing: 1
Degree-Uncertainty : 0.2

```

**Figure 5** Sample portion of a proposed probabilistic header.

uncertainty or accuracy of the sent response. It is a good practice to specify an ad hoc specific header to comply with *HTTP* standard. Figure 5 illustrates an example of a probabilistic header of a response returned by uncertain Web resources.

### 3.5 Proposed HTTP Request Template

To manage uncertain Web resources, it is necessary for the client who is querying an uncertain resource to understand this resource. To respect the principles of the Web and give the customer the opportunity to become

aware of the concept of uncertainty, we rely on content negotiation to serve our uncertain resources. This content negotiation is a mechanism provided by *HTTP* that serves different versions of the same resource representation (i.e., the same uniform resource identifier (*URL*)) to fit clients. Through this content negotiation, a client can notify the server of their intention to communicate with several uncertain resources. This communication can generate an enrichment of these initially conventional resources. The *HTTP* = 1.1 protocol defines eight methods, four of which define the interface, namely *POST* to create a resource, *GET* for the recovery, *PUT* for the change or modification of a state and *DELETE* to delete a resource [26]. In our model, we only focus on the *GET* method while differentiating traditional *GET* requests from uncertain requests. For this, we propose the  $GET^P$  notation, which describes a *GET* request from an uncertain conscious client (i.e., using specific headers to ask for uncertain representations).

The next definition gives the quantity of the *GET* request from an uncertain resource.

**Definition 3.7.** Let  $R^P$  be an uncertain resource deployed on  $URL_{R^P}$ . Let  $A_i, i \in [1, n]$ , be the set of attributes characterizing an  $R^P$  and  $V_{ij}, j \in [1, m]$ , its value respectively. We define a *GET* request from an uncertain resource, noted  $GET^P(URL_{R^P})$ , by the following equation:

$$GET^P(URL_{R^P}) = \langle A_i, V_{ij} \rangle. \quad (7)$$

Figure 6 illustrates an example of a probabilistic GET.

### 3.6 Uncertain Web Resource Composition Template

Since several uncertain resources can be linked to other resources that can also be uncertain, we propose a new model to generate uncertain compositions of resources. Let  $G$  be a composition of  $R_i$  Web resources, where each resource  $R_i$  is identified by a  $URL_i$  address, and each representation  $R_i$  is characterized by a set of possible representations of the resource  $PW_{R_i}$ . In this composition, each possible representation leads to the generation of a new possible Web PW. Since the probability of this possible Web is derived from the probabilities of its representations involved in this calculation, we should incorporate the representations of the involved resources; we suppose that our resources are independent.

**Definition 3.8.** Let  $P^{Mux}(C_i, v)$  and  $P^{Ind}(C_i, v)$  be the mutually exclusive and the independent probability, of choosing a subset  $C_i$  of the children of

```

{
  "ID": "Https://",
  "Type": "Author",
  "Last name": "Victor",
  "First name": "Hugo",
  "Age": Ind (<45,0,2>, <60, 0,8>),
  "City": "Paris",
  "Nationality": Mux (<"French",0,4>, <"English",0,2>, <"American", 0,2>)
}

```

**Figure 6** Example of a probabilistic GET response.

the node  $v$ , respectively. The resulting probability  $PW_i$  is then given by the following equation:

$$PW_i(C_i, v) = P^{Mux}(C_i, v) * P^{Ind}(C_i, v) \quad (8)$$

where  $\bar{C}$  comprises all children of  $v$  who are not in  $C_i$ :

Based on this scenario, we generate all the possible Webs that are derived from the involved resources, Figure 7. In this figure, the “*IdP*” attribute and the “*City*” resource are uncertain. For the “*IdP*” attribute in both possible Webs,  $PW_8$  and  $PW_9$  represent the unknown part of this attribute. Technically speaking, these *URLs* do not point to anything. Each probability is calculated using the formula in the previous section. For example, the possible Web probabilities for  $PW_4$  and  $PW_6$  is  $\text{prob}(W_4) = \text{prob}(09200) * \text{prob}(08200) * \text{prob}(\text{Canada}) = 0.6 * 0.4 * 0.2 = 0.048$ .

### 3.6.1 General P-GET algorithm

We propose general algorithm of P-GET called the *P-GET Algorithm* (Algorithm 1) that responds to any request and comprises five blocks. The first block represents the certain or uncertain client part. The second block introduces the control block that receives the user’s request. It mainly acts as an



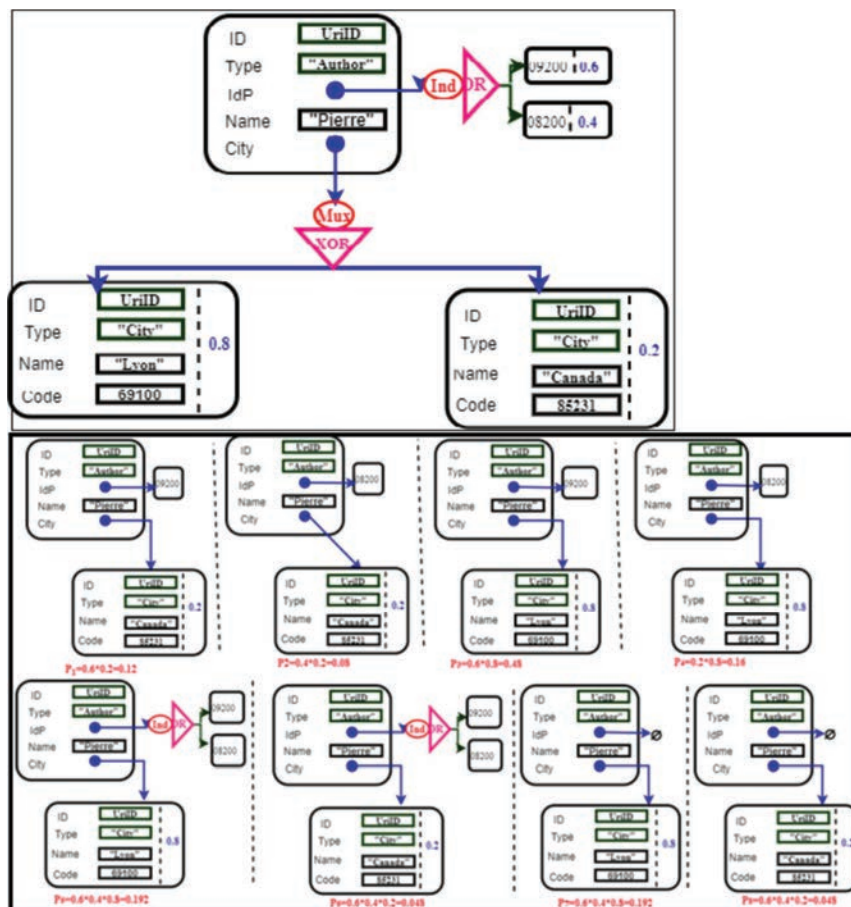


Figure 7 Example of interpreting composition services.

intermediary by sending the request between the user and the server. The third one looks for the representation of the resource requested by the client and retrieves the request sent by the control block and transmits it to the next block to generate all the possible Web resources, and then evaluates the query in each resource, which represents the role played in the fourth block. As for the last block, it takes all the answers obtained by the different resources as inputs. It aggregates these responses and yields a single output. We could, therefore, send the response to the server that subsequently transmits it to the control block. This proposed algorithm comprises four parts that are explained through blocks, as presented in Figure 8.

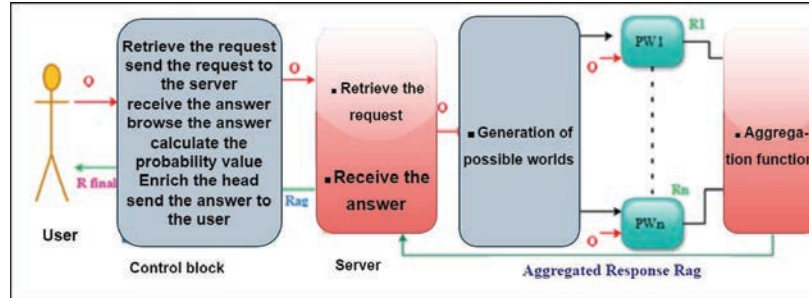


Figure 8 General architecture of the P-GET method.

## 4 Experimental results and evaluation

### 4.1 Dataset and Experimental Setup

We set up 16 RESTful services in an uncertain database of 100MB size and storing synthetic data on students, teachers, and universities. The total size of the population manipulated within this database is 4000 tuples. The used data details are presented in Table 2 which summarize experimental parameters setting. We have conducted four experimental studies under sets of data as mentioned in Table 2. The various experiments were carried out on a computer with a Windows 7 operating system, an Intel Core i5 processor and a capacity of 6GB of RAM. To be able to analyze our approach and measure its performance, we considered execution time as the main measure.

### 4.2 Implementation, Evaluation and Analyses of Results

In this section, we discuss and analyze the results of the various experiments carried out to evaluate our approach based on REST principles; therefore, it is possible to use any HTTP client to access our uncertain resources. To ensure the applicability of our approach and allow its integration with other approaches, we implement our P-GET algorithm as RESTful services where calls to these services are made via POST and GET to retrieve a user-friendly description of the service. Our algorithm transforms a request into a list of concepts to extract while moving from one resource to another, thus creating the path through the tree of different possibilities.

#### Experimental study 1

In the first experiment, we examine the effect of varying the number of services over execution time. In this series of experiments, we measure the

**Algorithm 1:** P-GET algorithm

---

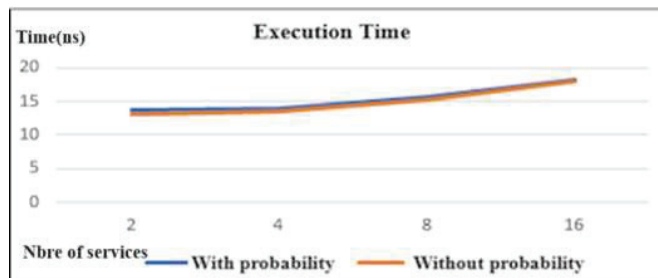
```

Require: URL
Ensure: Prob GET Case General
Begin
/* Step1: Sending the request */
(Cont. Serv) ← Request (Req). Receive a GET Request.
/* Step 2: Generation of the requested representations */
Server ← Request
PWI ← Generate all possible worlds.
for  $\forall pw_i$  in PW do
    | Evaluate the query
    | Receive the response from PWI:  $R_i$ 
end
Aggregate-Function ←  $R_i$ 's responses.
Server ← Aggregate-Response;
Control-block:Rag ← Aggregate-Response;
Get the answer
/* Step 3: Calculating degree of probability */
/* Browse Rag. */
if  $\exists Mux$  in Rag then
    |  $P \leftarrow P \text{ XOR } P^{Mux}$  /* with  $P \text{ Mux}$  calculated by the equation 5 */
    else
    end
if  $\exists Ind$  in Rag then
    |  $P \leftarrow P * P^{Ind}$  /* with  $P \text{ Ind}$  calculated by the equation 6 */
    end
end
/* End Rag */
Concatenate (all probability values)
Header ← Add (value of P)
Add indication (the header contains an uncertainty)
User ← "Final answer"
end
    
```

---

**Table 2** Experimental parameters setting

Experimentations	Values			
	Number of Services	Nodes $Ind$	Nodes $Mux$	Size of data (tuples)
First series	16	512	9	4000
Second series	16	512	9	4000
Third series	40	512	9	4000



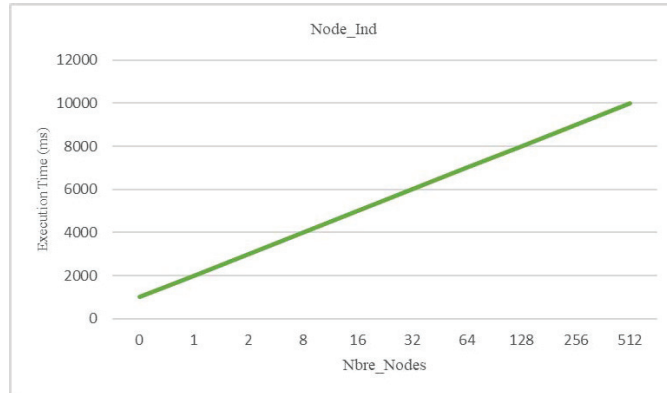
**Figure 9** Performance in terms of execution time with and without probability.

execution time required to compose services with and without probability calculation (see Figure 9). These experiments can assess the cost incurred by calculating the probability that the composition made to respond to a user request.

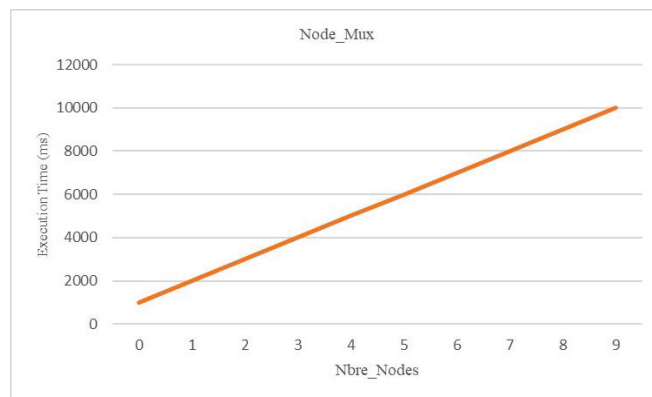
These experiments reveal that the time required to return responses to a request is estimated at 13.72 ms, which is a negligible value. However, when the number of services is two, the required execution time, set by the certain approach, is 13.04 ms; this time is almost the necessary time taken by the main classical information retrieval systems. The response time, given by our approach, has a linear behavior depending on the number of resources on the Web. This confirms the robustness of our solution and its convergence in a reasonable time even when this number increases considerably. Our approach therefore gives better results, since it makes it possible to manage the uncertainty of the data without requiring more time. In contrast, the GET probabilistic algorithm allows you to browse multiple JSON files to find the representation of a resource requested by the user.

### Experimental study 2

This experiments are carried out on a computer with a Windows 11 operating system, an Intel Core i7 processor and a capacity of 6 GB of RAM. To be able to analyze our approach and measure its performance, we considered execution time as the main measure. In this second study, we performed a set of tests to assess the impact of varying the number of nodes of type *Ind* and type *Mux*, assuming that there are no parameters in the path of the request. For the first type of node *Ind*, for which we have considered 512 nodes that we have made vary by  $2 * n$  nodes starting from  $n = 1$  node until reaching 512 nodes, our algorithm succeeded in recovering all the possibilities of response to the request. For each set of  $n$  possible responses (see Figure 10),



**Figure 10** Response time by varying number of nodes of *Ind* type.



**Figure 11** Response time by varying number of nodes of *Mux* type.

there is a generation of  $2 * n$  possible responses to retrieve. To improve the performance of our algorithm, we have made some modifications to it to treat only one possibility without generating the others. The results, illustrated in Figure 10, show that the execution time is proportional to the number of possibilities (1 possibility = 1000 ms) and that the execution time, of the order of  $10^{+04}$  ms for 512 nodes, necessary for the processing of the request, remains reasonable and can adapt to many possibilities of nodes of type *Ind*. For the second type of node *Mux*, for which we considered 9 nodes and we varied this number, our algorithm gives results, illustrated in Figure 11, which also show that the execution time, of the order of  $10^{+04}$  ms for 9 nodes, is

also proportional to the number of possibilities and reasonable and can thus adapt to many possibilities of type nodes *Mux*.

### Experimental study 3

These experiments are carried out on a computer with a Windows 7 operating system, an Intel Core i5 processor and a capacity of 6 GB of RAM. To be able to analyze our approach and measure its performance, we considered execution time as the main measure. The third experimental study concerns two series of experiments to evaluate the impact of the variation in the number of services on the execution time, on the one hand, and to evaluate the impact of varying data size on query execution time, on the other. In the two series of experiments, we have, in a first case, carried out the tests without considering the associated degrees of probability and, in a second case, with taking these degrees into account. Regarding the first set of tests, we considered 40 different services, which we varied in steps of five services (see Figure 12). The results provided by our algorithm show that they are better with taking into account the uncertainty of the data than those without taking into account this uncertainty with a very reasonable execution time, of the order of  $5^{+09}$  ns for 40 services, since it is slightly higher than that without taking into account for uncertainty. For the second set of tests, we considered a test database of size 4000, which we varied in steps of 1000 (see Figure 13). The results provided by our algorithm show that the execution time, of the order of  $2.25^{+09}$  ns for a size of 4000, is better taking into account the uncertainty of the data than without taking this uncertainty into account, while remaining reasonable.



Figure 12 Response time by varying number of services with/without uncertainty.



Figure 13 Response time by varying size of data with/without uncertainty.

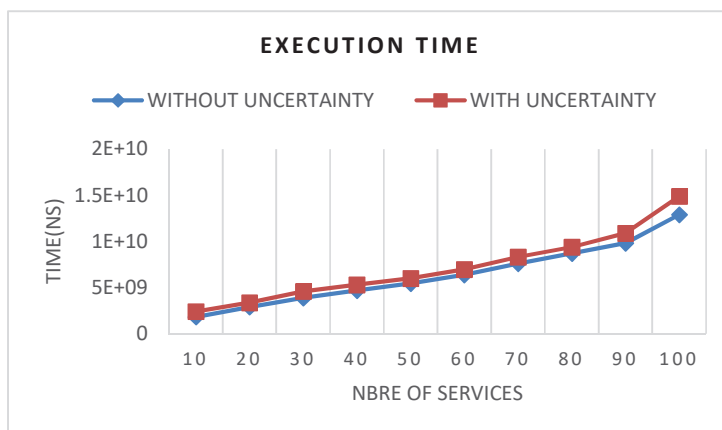


Figure 14 Response time by varying number of services with/without uncertainty.

#### Experimental study 4

These experiments are carried out on a computer with a Windows 11 operating system, an Intel Core i7 processor and a capacity of 8GB of RAM. To be able to analyze our approach and measure its performance, we considered execution time as the main measure. Figure 14 shows the execution time results of the two approaches by varying the number of services involved to answer a request. The result of this experiment showed that the time taken to calculate these values is negligible. This experience allows us to evaluate the cost incurred by calculating the probability of our composition algebra. For this reason, we have set up 100 Web services on an uncertain database storing

synthetic data on products, consumers, and salespeople. The execution time (response time to a query) given by our approach is negligible; we notice a small increase from the number of services 90 but the difference still remains acceptable. In any case, the time required for the composition of the Web services of our approach is negligible compared to approach 2. We can deduce from these results that our approach gives better results, because it allows us to manage the uncertainty of the data and does not increase the response time too much.

## 5 Discussion

While Web resources provide a rigorous fundamental basis for modeling the relationships between data and users, the data extracted from these resources remains noisy and lacking in quality. The main limitations of the approaches proposed in the current literature are the complexity in determining the degree of uncertainty relating to each resource. This uncertainty reduces the quality of responses sent to users and of the information and data stored on the Web, which form the basis of information retrieval systems. Our main objective was to introduce: (i) a new type of Web resources, both certain and uncertain, (ii) a new programming language, and (iii) an analyzer and the necessary algorithms for handling these resources. We have proposed a new concept of probabilistic REST Web resources and a modification of the method of retrieving GET information. The main first contribution consists of the new modeling proposed for probabilistic Web resources characterized by its simplistic structure, its efficiency in managing uncertainty and its low effect on response time and complexity.

The second contribution being the definition of new precise and automatic rules to answer the queries presented as a set of algebraic equations. In our approach, the two types of nodes we proposed allowed us to manage a set of possible Websites that are independent of each other.

The third contribution is that of the proposal for a new module, namely the probabilistic JSON parson. Our approach is characterized by hypertext links and to implement them on a browser, we have developed a probabilistic programming language and an analyzer for the validation phase. Through this last module, we have proposed new headers for the HTTP protocol making it possible to signal, to users, the uncertainty of resources and, at the same time, determine the degree of uncertainty associated with each of these resources.



## 6 Conclusion

The proposed approach represents a solution that allows Internet users to manage the uncertainty of data while simultaneously navigating on hypertext. Theoretically, we recognized the need to develop a solution for managing and processing data uncertainty during referencing and browsing Web resources. Our approach makes it possible not only to represent uncertain Web resources but also to consider uncertain resources as specific resources that could have several possible probabilistic representations.

We developed an analyzer to validate the proposed probabilistic model and help developers use this approach on the HTTP browser. We also proposed an algebraic approach for the interpretation and evaluation of the data requests in compositions of uncertain resources based on the model of uncertain Web resources. In this context, we have defined data requests as Web resource paths and proposed algorithms to evaluate requests within uncertain resources. Our algorithm enables the recovery and calculation of the uncertainty associated with a response to a given request.

## References

- [1] Y. Hammal, K. Salah Mansour, A. Abdelli, L. Mokdad, 'Formal techniques for consistency checking of orchestrations of semantic Web services'. *Journal of Computational Science*, 44: 101-165, 2020. <https://doi.org/10.1016/j.jocs.2020.101165>.
- [2] C. Hu, X. Wu, B. Li, 'A Framework for Trustworthy Web Service Composition and Optimization'. *IEEE Access*, 8: 73508–73522, 2020. doi: 10.1109/ACCESS.2020.2984648.
- [3] C. Guo, J. Jia, Y. Jie, C. Z. Liu, K. R. Choo, 'Enabling secure cross modal retrieval over encrypted heterogeneous IoT databases with collective matrix factorization'. *IEEE Internet Things Journal*. 7: 3104–3113, 2020. doi: 10.1109/JIOT.2020.2964412.
- [4] S.P. Jaikar, R.L. Kamatchi, 'A survey of messaging protocols for IoT systems'. *International Journal of Advanced in Management, Technology and Engineering Sciences*. 8: 510–514, 2018. doi: 16.10089.IJAMTES.2018.V8I01.15.20552.
- [5] A. Omri, K. Benouaret, M. N. Omri, D. Benslimane, 'Toward a new model of indexing big uncertain data'. In R. Chbeir, A. Kawtrakul, W. I. Grosky, and A. Ouni (Eds.), *Proceedings of the 9th International*

- Conference on Management of Digital EcoSystems, MEDES, Bangkok, Thailand, 2017.* <https://doi.org/10.1145/3167020.3167034>.
- [6] D. Benslimane, Q.Z. Sheng, M. Barhamgi, H. Prade, ‘The uncertain Web: Concepts, challenges, and current solutions’. *ACM Transactions on Internet Technology*, 1:1–6, 2016. doi:10.1145/2847252.
- [7] F. Chen, R. Dou, M. Li, H. Wu, ‘A flexible QoS-aware Web service composition method by multi-objective optimization in cloud manufacturing’. *Computers & Industrial Engineering*. 99: 423–431, 2016. <https://doi.org/10.1016/j.cie.2015.12.018>.
- [8] A. L. Lemos, F. Daniel, B. Benatallah, ‘Web service composition: A survey of techniques and tools’. *ACM Computing Surveys*, 48: 1–41, 2016. <https://doi.org/10.1145/2831270>.
- [9] A. Malki, D. Benslimane, S.M. Benslimane, M. Barhamgi, M. Malki, P. Ghodous, K. Drira, ‘Data services with uncertain and correlated semantics’. *World Wide Web*, 19(1): 157–175, 2016. <https://doi.org/10.1007/s11280-014-0317-x> doi: 10.1007/s11280-014-0317-x.
- [10] Amdouni, S., Barhamgi, M., Benslimane, D., and Faiz, R. (2014). Handling uncertainty in data services composition. In IEEE International Conference on Services Computing, SCC 2014, Anchorage, AK, USA, June 27 – July 2, 2014 (pp. 653–660). IEEE Computer Society. URL: <https://doi.org/10.1109/SCC.2014.91>. doi: 10.1109/SCC.2014.91.
- [11] Filali, F. Z., and Yagoubi, B. (2015). Classifying and filtering users by similarity measures for trust management in cloud environment. *Scalable Computing: Practice and Experience*, 16, 289–302. URL: <http://www.scpe.org/index.php/scpe/article/view/1102>.
- [12] Malki, A., Barhamgi, M., Benslimane, S. M., Benslimane, D., and Malki, M. (2015). Composing data services with uncertain semantics. *IEEE Trans. Knowl. Data Eng.*, 27, 936–949. URL: <https://doi.org/10.1109/TKDE.2014.2359661>. doi: 10.1109/TKDE.2014.2359661.
- [13] Pierre, D. V., Michaël, M., and Djamal, B. (2016). Modeling and composing uncertain Web resources. In H. Sack, G. Rizzo, N. Steinmetz, D. Mladenic, S. Auer, and C. Lange (Eds.), *The Semantic Web* (pp. 327–341). Cham: Springer International Publishing.
- [14] Omri, A., Benouaret, K., Omri, M. N., and Benslimane, D. (2016). Querying data services in an uncertain environment: A possibilistic-based approach. In K. Yétongnon, A. Dipanda, R. Chbeir, G. D. Pietro, and L. Gallo (Eds.), *12th International Conference on Signal-Image Technology & Internet-Based Systems, SITIS 2016, Naples, Italy,*

- November 28 – December 1, 2016 (pp. 246–251). IEEE Computer Society. URL: <https://doi.org/10.1109/SITIS.2016.47>. doi: 10.1109/SITIS.2016.47.
- [15] Omri, A., Benouaret, K., Omri, M. N., and Benslimane, D. (2017). Toward a new model of indexing big uncertain data. In R. Chbeir, A. Kawtrakul, W. I. Grosky, and A. Ouni (Eds.), *Proceedings of the 9th International Conference on Management of Digital EcoSystems, MEDES 2017, Bangkok, Thailand, November 07-10, 2017* (pp. 93–98). ACM. URL: <https://doi.org/10.1145/3167020.3167034>. doi: 10.1145/3167020.316703.
- [16] Boulaares, S., Omri, A., Sassi, S., and Benslimane, D. (2018). A probabilistic approach: A model for the uncertain representation and navigation of uncertain Web resources. In G. S. di Baja, L. Gallo, K. Yétongnon, A. Dipanda, M. C. Santana, and R. Chbeir (Eds.), *14th International Conference on Signal-Image Technology & Internet-Based Systems, SITIS 2018, Las Palmas de Gran Canaria, Spain, November 26–29, 2018* (pp. 24–31). IEEE. URL: <https://doi.org/10.1109/SITIS.2018.00015>. doi: 10.1109/SITIS.2018.00015.
- [17] Abdelhak, E., Fethallah, H., and Mohammed, M. (2019). Qos uncertainty handling for an efficient Web service selection. In *Proceedings of the 9th International Conference on Information Systems and Technologies, ICIST 2019, Cairo, Egypt, March 24-26, 2019* (pp. 17:1–17:7). ACM. URL: <https://doi.org/10.1145/3361570.3361592>. doi: 10.1145/3361570.3361592.
- [18] She, Q., Wei, X., Nie, G., and Chen, D. (2019). Qos-aware cloud service composition: A systematic mapping study from the perspective of computational intelligence. *Expert Syst. Appl.*, 138. URL: <https://doi.org/10.1016/j.eswa.2019.07.021>. doi: 10.1016/j.eswa.2019.07.021.
- [19] Abdulwadood, A. S., Mahmood, M. A. B., and Salim, D. T. (2020). Uncertain data reduction based on demographic analysis for tourist place recommendations. *International Journal of Innovation, Creativity and Change*, 11.
- [20] Sekkal, N., Benslimane, S. M., Mrissa, M., Park, C. Y., and Boudaa, B. (2020). Proactive and reactive context reasoning architecture for smart Web services. *International Journal of Data Mining, Modelling and Management*, 12, 1–27. URL: <https://ideas.repec.org/a/ids/ijdm/v12y2020i1p1-27.html>.
- [21] Veeraiyan, R., and Ramakrishnan, S. (2020). Fuzzy logic based ontological modelling for student academic performance prediction in pervasive

- environments. The International journal of analytical and experimental modal analysis. URL: <http://ijaema.com/gallery/271-january-3311.pdf>.
- [22] Paulo Cesar G. da Costa, Kathryn B. Laskey, Kenneth J. Laskey (2008) PR-OWL: A Bayesian Ontology Language for the Semantic Web. Uncertainty Reasoning for the Semantic Web I. 978-3-540-89765-1F. doi: 10.1007/978-3-540-89765-1\_6.
- [23] Stoilos, G., Simou, N., Stamou, G., and Kollias, S. (2006). Uncertainty and the semantic Web. *IEEE Intelligent Systems*, 21(5), 84–87.
- [24] Eckhardt, A., Horváth, T., Maruščák, D., Novotný, R., & Vojtáš, P. (2008). Uncertainty issues and algorithms in automating process connecting Web and user. In *Uncertainty Reasoning for the Semantic Web I: ISWC International Workshops, URSW 2005–2007, Revised Selected and Invited Papers* (pp. 207–223). Springer Berlin Heidelberg.
- [25] Ceolin, D., van Hage, W. R., Fokkink, W. J., and Schreiber, G. (2011, October). Estimating Uncertainty of Categorical Web Data. In *URSW* (pp. 15–26).
- [26] Leonard, R., and Sam, R. (2007). *Restful Web Services*. (1st ed.). O'Reilly.

## Biographies



**Asma Omri** received her Ph.D. in computer science from the University of Claude Bernard Lyon1, France in 2018. Her areas of interest include uncertainty, indexing, information retrieval, web, web services, among others.



**Mohamed Nazih Omri** is a professor in computer science at the University of Sousse, Tunisia. He is a member of MARS (Modeling of Automated Reasoning Systems) Research Laboratory. His group conducts research on information retrieval, data base, knowledge base and web services.

