International Journal of Scientific & Engineering Research, Volume 4, Issue 10, October-2013 1614
ISSN 2229-5518

A novel Intelligence Approach to Design a Semantic Description Framework for Semantic Grid Resources.

T. A. Farrag, M. T. Faheem
Department of Electrical Engineering, Faculty of Engineering, Taif University, Saudi Arabia

ABSTRACT—Semantic web services (SWSs) represent the backbone of machine-to-machine interaction in Semantic Grid. The use of the semantic data model makes the information about grid resources not only machine-readable (like the current grid systems) but also machine-understandable. The problem of discovering and selecting the most suitable web service represents a challenge for SWSs to be widely used. Ongoing to test the efficiency of our new proposed Semantic Description Framework for Semantic Grid Resources, a new intelligent approach is introduced in this paper to classify the SWSs. This approach depends on the semantic relations among the different concepts to measure the relative importance value of each concept using a new proposed concepts ranking algorithm (CRA). To apply this approach, a new automatic SWSs classifier (ASWSC) is introduced. ASWSC is a pure semantic based classifier, which does not use the traditional classification techniques (such as text mining, neural, genetic and rough sets). ASWSC is evaluated using experiments which show a high percentage of accuracy and precision that remarks the proposed algorithm over some other non-semantic classifier.

Index Terms— Grid, Semantic, resource description, RDF, web 3.0, machine-understandable

1. INTRODUCTION

The grid can be considered as the base of the modern trends in the field of computing systems. The grid systems suffer from the problem of weakness of the current data models which are used to define the gird resources. These models suffer from the lack of semantics which leads to resources wasting. The lack of semantics yields in misunderstanding of resources information and makes the process of inferring new relations among the grid resources almost not available. The use of the semantic data model makes the information about grid resources not only machine-readable (like the current grid systems) but also machine-understandable. Ongoing to test the efficiency of our new proposed Semantic Description Framework for Semantic Grid Resources, a new approach is introduced in this paper to classify the SWSs. This approach depends on the semantic relations among the different concepts to measure the relative importance value of each concept using a new proposed concepts ranking algorithm (CRA). To apply this approach, a new automatic SWSs classifier (ASWSC) is introduced. ASWSC is a pure semantic based classifier, which does not use any of the traditional classification techniques (such as text mining ,neural, genetic and rough sets). ASWSC is evaluated using empirical experiments which show a high percentage of accuracy and precision that remarks the proposed algorithm.

The rest of this paper is organized as follows: In section 2, a brief background about the developing of the grid computing is introduced. In section 3, we focus on the new trend of the grid computing which is called Semantic Grid. In section 4, the current Grid Resources Description Framework is surveyed. In section 5, we introduce our vision about a proposed new Semantic grid resources description framework.. In section 6, we introduce our prepared testing bed. Section 7 defines the semantic web services . Section 8 introduces the new intelligent SWSs classification approach. In Section 9, we introduces concepts ranking algorithm (CRA). section 10 introduces the automatic semantic web services classifier (ASWSC). In section 11, we introduces empirical experiments to test and verify the proposed algorithm. Section 12 concludes this paper.

2. GRID COMPUTING

Due to the dire need to get over the problem of centralization and the high cost of supercomputers, the Grid Computing appeared in 1990s[1]. The Grid consists of a large number of PCs distributed geographically and communicated together through a special network. The Grid is the infrastructure of the distributed processing model in which the unused processing power of PCs is imposed by the other PCs, which have large processing needs. The research in the field of the grid computing results many new concepts that enrich the computer science such as Internet computing , mobile computing [2]

- This work was supported byTaif University, KSA under grant 2859-434-1.
- The Authors are with the department of the Electrical Engineering, Faculty of Engineering, Taif University, KSA.

and recently cloud computing [3][4]. All of these concepts is derived from the idea of grid computing.

In addition, the unexpected ongoing of the Internet and the huge number of PCs and resources attached to the Internet rises a question "Why we do not apply this huge Computing Power on a large scale?" Although that more 10 billion computers worldwide are connected to the Internet (according to IMS Research.[5]) , researches assert that about 90 percent of the time these computers are on, they are idle—that is, not actively performing computing tasks. The sum of these unused computational powers is 25000 times more powerful than the most powerful supercomputer all over the world .

So, the research in the field of grid computing become affected by the development of the Internet researches. For example, All the modern approaches that deals with the problem of enabling resource sharing of geographically diverse computational resources, use the Web Services. The new trend in the Internet researches is a about the next generation of the web which called web 3.0 or the semantic web. Web 3.0 is a set of different technologies that can enormously change the existed form and technique of the current web functions. With these new technologies, the web would become much smarter and easier for people to use. There is no standard definition of web 3.0 technology; however, this concept is defined by different experts in various ways.

3. Semantic Grid

In 2001, Berners-Lee said, "The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation"[6]. In other words, semantic web is an evolving extension of the current www in which the semantics of information and services on the web is defined, making it possible for the web to understand and satisfy the requests of people and machines to efficiently use the web content [7][8]. So, semantic web aims to convert the web contents from machine readable to machine understandable [10][11][12].

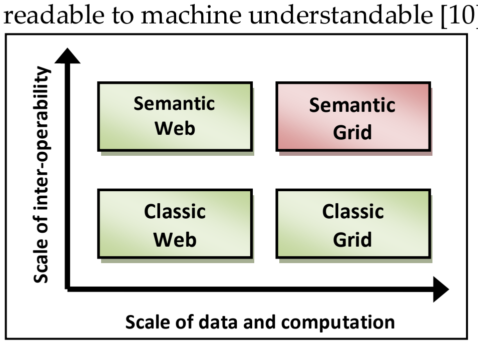

Figure 1: From the current Grid to the semantic grid

As shown in Figure 1, the process of importing the new technologies [13][14][15] of the web 3.0 and the semantic web produces a new enhanced generation of the grid called semantic grid[17]. The Semantic Grid refers to an approach to grid computing in which information, computing resources and services are described using the semantic data model [18][19]. As shown Figure 1, Semantic Grid combines higher inter-operability (of sematic web) with greater computational facilities (of the grid).

The main contributions of using semantic web technologies in the grid context are : (1) redefine the grid resources and services in a semantic manner (2) replace the use of the conventional web services by the new concept of semantic web services. So, there is a need to propose a new framework or upper ontology[20][21] to describe the grid resources. This proposal benefit from the semantic web languages and the modern researches in field of semantic web services[22][23][24]. Figure 2 summarizes the main research areas in the computing systems. This research can be considered as a contribution is fields of data management and resource discovery.

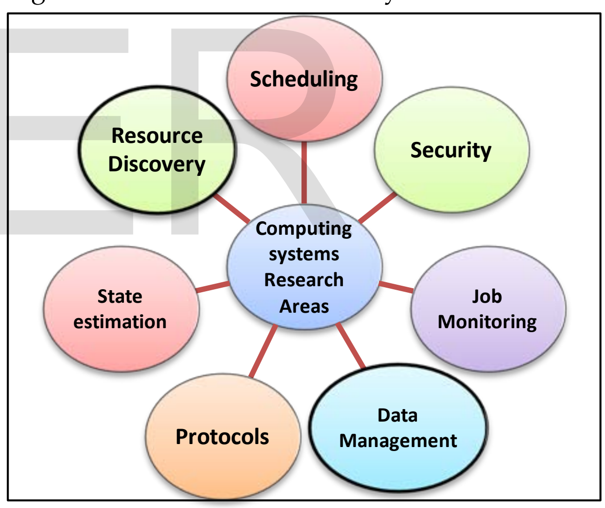

Figure 2: Computing systems Research Areas taxonomy

4. Grid Resources Description Framework

The grid resources description frameworks (GRDFs) can be categorized into two main categories according to the used technologies. They can categorized to semantic and non-semantic frameworks. In non-semantic ones, the resource provider and the resource customers descript the available or required resource in a nonstandard way. For example, the resource provider can say that "I have 32MB free Memory" and on the other hand the resource provider can say that " I need 16MB RAM". In this example the resource matchmaking system cannot infers that the mean of the "free memory" equivalent to "free Ram" except if the system administrator provide this information manually.

ISSN 2229-5518

The main idea of the semantic web is to extend the current human-readable web by encoding some of the semantics of resources in a machine-processable form [25]. Moving beyond syntax opens the door to more advanced applications and functionalities on the Web. So, the semantic GRDFs will enhance the way of grid resources description which enable the resources management and matchmaking systems to infer new non-provided relations, unify the ways of resource advertising and resource requesting. The non-semantic GRDFs use Attribute based resource description model and the semantic GRDFs use ontology –based resource description model.

Next in this section, we will survey the most important non-semantic resource management systems:

5. Proposed Semantic Grid Resources Description Frameworks

In order to design a semantic grid, the first step is to design an upper ontology to define any grid aspect. Therefore, a survey about the grid components and the grid workflow should be prepared to count the main concepts and their properties that should be included in this upper ontology. Simply , the grid consists of a number of resources that cooperate to perform the different jobs requested by the grid users. So, The three main concepts in our proposed grid upper ontology are Grid Resource, Grid User and Grid Job as shown in Figure 3 .

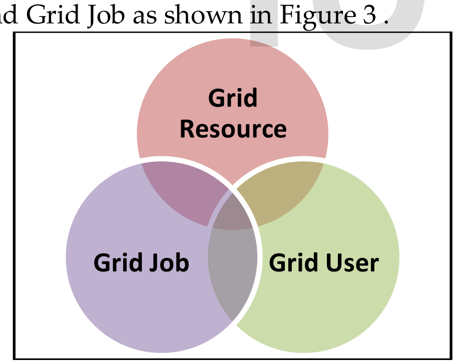

Figure 3: Grid Upper ontology main component

5.1. Semantic Grid Resources Definition

In this paper , we focus on the semantic definition of the grid resources A Grid resource is a Grid entity that provides some capabilities to a consumer. different resources could provide similar capabilities but with different quality of service aspects. The resource capabilities are required to be presented in such a way that a consumer can easily discover a resource or a resource ensemble with needed capabilities. By employing this, we attain the following:

- Grid resource management systems make a large step towards compatibility with the Semantic Web and Grid resources descriptions become web-understandable.
- The resource provider can have maximum freedom to describe resources with different levels of complexity and completeness.
- XML-schema datatypes can be exploited for resource description.
- Complex resource matching is possible based on subsumption relationship.
- A conceptual defnition of resources in a more natural way is possible based on the restriction over the resource attributes, and a semantics level of agreement between resource provider and consumer can be achieved.
- After having a conceptual and flexible resource description, a resource broker can categorize the resource ensembles and devise alternative options.
- Most promisingly, clients can express complex requests in a simple human as well as machine understandable format. Also the system can fulfill more resource requests than without ontology language. For example, in a system without ontology language, a request for *Unix* system fails if the term *Unix* is not specified. Using an ontology this might however be successful.
- Spelling or typing errors in descriptions and requests are prevented by using a controlled vocabulary.

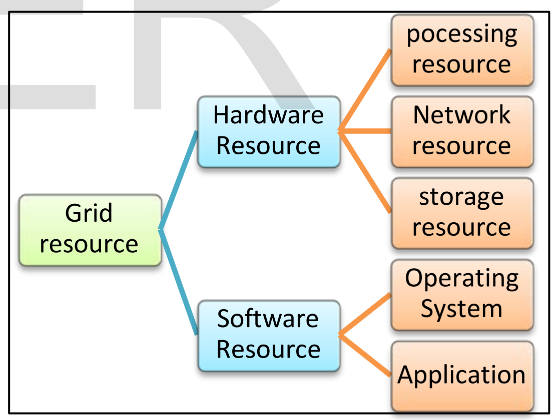

Figure 4: Grid Resource main sub concepts.

Figure 4 shows uncompleted taxonomy of the grid resources sub concepts which should be included in any upper ontology to define and descript and resource in the semantic grid. We select OWL [28] for our current work. OWL is a good language for providing more complex constraints on the types of resources and their properties. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDFS. OWL has three increasingly expressive sublanguages: OWL Lite, OWL DL, and OWL Full.

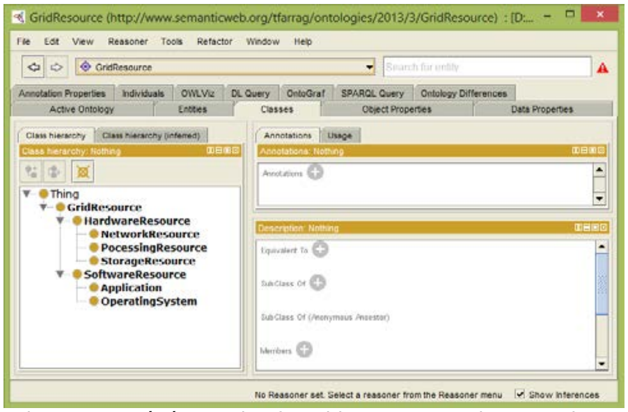

Figure 5: Protégé snapshot for grid resource ontology creation

We use Protégé [29] to write our grid upper ontology as shown in Figure 5. Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. Protégé enables users to build ontologies for the Semantic Web, in particular in the W3C's OWL. An OWL ontology may include descriptions of classes, properties and their instances. Given such an ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms. Figure 6 shows a part of the OWL file that defines the contents of the Grid resource ontology.

```
<?xml version="1.0"?>

<Ontology xmlns="http://www.w3.org/2002/07/owl#"
    xml:base="http://www.semanticweb.org/tfarrag/ontologies/2013/3/GridResource"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
ontologyIRI="http://www.semanticweb.org/tfarrag/ontologies/2013/3/GridResource">
  <Prefix name="" IRI="http://www.w3.org/2002/07/owl#"/>
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
  <Declaration>
    <Class IRI="#Application"/>
  </Declaration>
  <Declaration>
    <Class IRI="#GridResource"/>
  </Declaration>
  <Declaration>
    <Class IRI="#HardwareResource"/>
  </Declaration>
  <Declaration>
    <Class IRI="#NetworkResource"/>
  </Declaration>
  <Declaration>
    <Class IRI="#OperatingSystem"/>
  </Declaration>
  <Declaration>
    <Class IRI="#PocessingResource"/>
  </Declaration>
  <Declaration>
    <Class IRI="#SoftwareResource"/>
  </Declaration>
  <Declaration>
    <Class IRI="#StorageResource"/>
  </Declaration>
  <SubClassOf>
    <Class IRI="#Application"/>
    <Class IRI="#SoftwareResource"/>
  </SubClassOf>
  <SubClassOf>
    <Class IRI="#HardwareResource"/>
    <Class IRI="#GridResource"/>
  </SubClassOf>
  <SubClassOf>
    <Class IRI="#NetworkResource"/>
    <Class IRI="#HardwareResource"/>
  </SubClassOf>
  <SubClassOf>
    <Class IRI="#OperatingSystem"/>
    <Class IRI="#SoftwareResource"/>
  </SubClassOf>
  <SubClassOf>
    <Class IRI="#PocessingResource"/>
    <Class IRI="#HardwareResource"/>
  </SubClassOf>
  <SubClassOf>
    <Class IRI="#SoftwareResource"/>
    <Class IRI="#GridResource"/>
  </SubClassOf>
  <SubClassOf>
    <Class IRI="#StorageResource"/>
    <Class IRI="#HardwareResource"/>
  </SubClassOf>
:
</Ontology>
```

Figure 6: part of OWL file of Grid Resource Ontology

6. Testing Bed

In order to test our proposals, we construct a small Grid consist of 6 PCs. Alchemi is used to implement this testing grid. However, there is a distinct lack of service-oriented architecture-based grid computing software in this space. To overcome this limitation, a Windows-based desktop grid computing framework called Alchemi is developed and implemented on the Microsoft .NET Platform. The Microsoft .NET Framework is the state of the art development platform for Windows and offers a number of features which can be leveraged for enabling a computational desktop grid environment on Windows-class machines. Next, we present the main features of Alchemi and the main terminologies that are used in it.

6.1. Alchemi

Alchemi is conceived with the aim of making grid construction and development of grid software as easy as possible without sacrificing flexibility, scalability, reliability and extensibility. The key features supported by Alchemi are:

- Internet-based clustering [30][31] of Windows-based desktop computers.
- Dedicated or non-dedicated (voluntary) execution by individual nodes.
- Object-oriented grid application programming model (fine-grained abstraction).
- file-based grid job model (coarse-grained abstraction) for grid-enabling legacy applications.

- web services interface supporting the job model for interoperability with custom grid middleware e.g. for creating a global, cross-platform grid environment via a custom resource broker component.

According to Alchemi, Grids are constructed using three types of distributed components (or nodes). They are named according to their roles with respect to a grid application.:

1. **Manager** :The Manager manages the execution of grid applications and provides services associated with managing thread execution. It is deployed as an executable. An optional sub-component of the Manager is the Cross Platform Manager, which is deployed as a web service.
2. **Executor**: The Executor executes individual grid threads and provides services associated with executing threads. It is deployed as an executable. An Executor can be configured to be dedicated (meaning the Manager initiates thread execution directly) or non-dedicated (meaning that thread execution is initiated by the Executor on a volunteer basis via a screen saver or some other user-defined options.)
3. **Owner** : The Owner owns an application and provides services associated with the ownership of an application (and its constituent threads). The Owner is implicitly created by the Alchemi API.

6.2. Testing Grid Configuration

As mention before, our testing grid consist of 6 PCs. We configure one PC as a grid manager and the rest PCs as executor as shown in Figure 7.

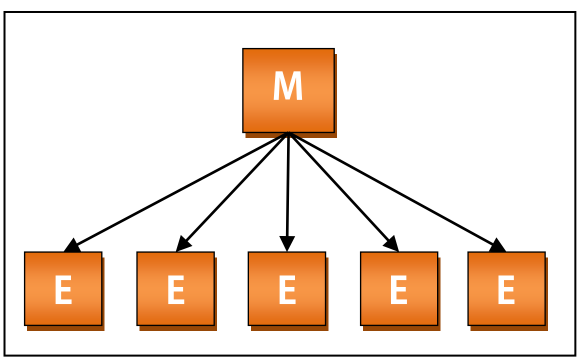

Figure 7: Testing Grid Configuration

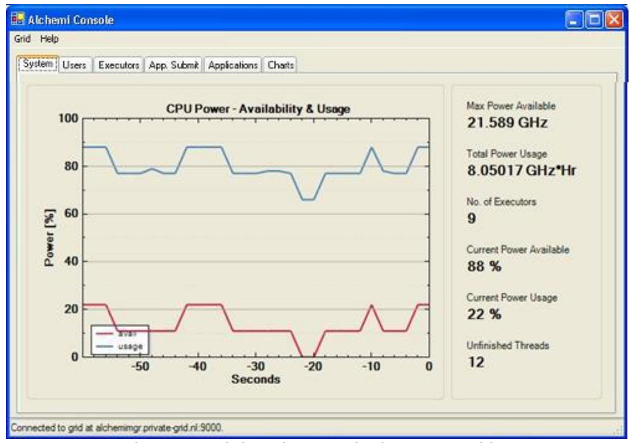

Figure 8: Alchemi Console (System tab)

We use the Alchemi console application to monitor the process of grid establishing as shown in Figure 8. We use some of applications examples provided by Alchemi developer to ensure the work of the testing grid as shown in Figure 9.

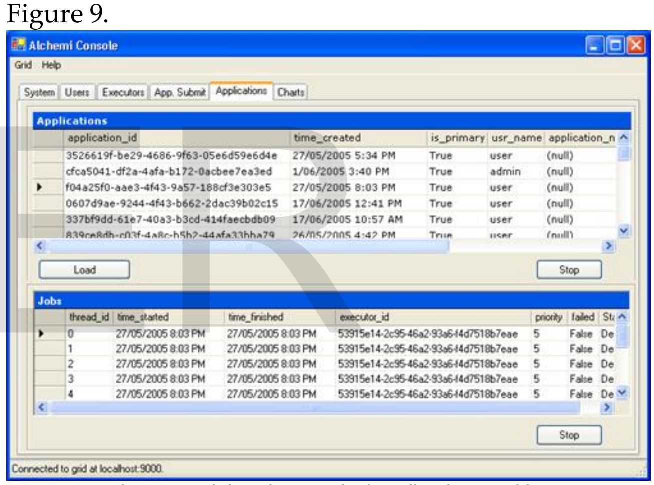

Figure 9: Alchemi Console (Applications tab)

7. Semantic Web Services

Semantic Web services (SWSs) is the a modern research topic and represent one of the most important resources in the sematic grid. SWSs are constructed when using the semantic web technologies in addition to traditional web services technologies to describe the web services. As a semantic grid resource, SWSs use markups that make data machine-readable in a detailed and sophisticated way (as compared with human-readable HTML, which is usually not easily "understood" by computer programs).

8. An Intelligent Approach for SWSs Classification

It is possible to state that most of the SWSs that belong to the same group often use a specific set of ontology concepts extensively and frequently. This set of concepts represents a part of a huge number of concepts that are expected to be found in any SWSs system. Therefore, our new intelligent

ISSN 2229-5518

classification approach is based on measuring the relative importance value of each one of the concepts that used by the SWSs in each group. Later in this paper, concepts ranking algorithm (CRA) provides the details of this process. According to the new classification approach, to select the best group of services for a new SWS (in other words, to classify a new SWS), we should at first compute the summation of relative importance values of the new SWSs concepts for each group. Then, the group that gets the maximum summation value represents the recommended group for the new SWS. The details of this process are provided later in this paper using a new proposed automatic semantic web services classifier (ASWSC).

9. Concepts Ranking Algorithm (CRA)

The ranking value is a number that represents the importance of a specific concept in a certain group. In our philosophy, the importance of a concept (Rank) depends on three points: (1) the number of services that use this concept (2) the existence and the importance of the concept parent (super concept) (3) the existence and the importance of the concept children (sub concepts). In other words, the ranking value is derived from three components: self, parent and children components that should be calculated to get the total ranking value.

CRA Algorithm

INPUT

A set of pre-classified semantic web services considered as algorithm training set

OUTPUT

The Rank of each concept in each group concept list GCL

Terminology

- **GS:** list of services in the group
- **GCL:** list of <concept , number_of_usage> pairs used by a group
- **N:** number of services that use the concept
- **Max_N** : maximum number_of_usage in GCL
- **P_S:** percentage that represents the importance of the concept in the group (self-ranking component)
- **C_P:** the parent concept of Concept (C)
- **P_p:** percentage that represents the importance of **C_P** in the group (parent-ranking component)
- **SCL :** list of sub-concepts of a specified concept
- **M :** total number of Concept children (sub-concepts)
- **S:** number of sub-concepts of a specific concept that exist in the group
- **$\sum_{SCL} P_s$**: summation of Ps of the concepts that belong to SCL
- **P_C:** percentage that represents the Children-ranking component
- **R** : the total ranking value of a specific concept

STEPS

```
** Concepts Frequency distribution in a Group**
foreach( service S in GS)
    foreach (Concept C in (S_inputs U S_outputs))
        if C ∈ GCL then
            update the pair <C,N> by increment N
        else
            add the pair <C,1> to GCL
        end if
    next
next
*** The concept self-ranking component***
foreach (Concept C in GCL)
    P_S = N/ Max_N
    extend <C,N> pair to <C,N, P_S>
next
*** The concept parent-ranking component***
foreach (Concept C in GCL)
    C_P= getParentConcept(C)
    if CP ∈ GCL then
        P_p = getSelfRank(C_P)
    else
        P_p =0
    end if
    extend <C,N, P_S> to <C,N, P_S, P_p >
next
*** the concept Children-ranking component***
foreach (Concept C in GCL)
    SCL= getSubConcepts(C)
    M=getCount(SCL)
    S=0
    ∑_SCL P_s=0
    foreach (Concept C' in SCL)
        if C' ∈ GCL then
            ∑_SCL P_s=∑_SCL P_s+ getSelfRank(C')
            S=S+1
        end if
    next
** Calculating the total ranking Value***
    P_C=(∑_SCL P_s /S) * (S/M)
    R= W_S* P_S+ W_P* P_P + W_C* P_C
    extend <C,N,P_S, P_P> to <C,N,P_S,P_P,P_C,M ,S,R>
next
```

Figure 10: CRA algorithm

9.1. Ranking Algorithm

Concepts ranking algorithm (CRA) is proposed based on the previously mentioned philosophy. Its process starts with preparing a list of concepts used by each group of services (GCL). Then, CRA determines the concepts frequency distribution inside the group. This distribution represents the number of services (N) that use each concept. Then, the

ISSN 2229-5518

highest value in this frequency distribution is defined (Max_N). There are four stages to calculate the ranking value of a specific concept:

1. To calculate the self-component of the concept ranking value (PS), the ratio between N and Max_N is calculated.
2. To calculate the parent component (PP), the parent concept is defined. If it belongs to GCL then PP equals to PS of this parent concept, otherwise PP equals to zero.
3. To calculate the children component (Pc), the full list of the sub-concepts (SCL) is prepared. Then, the total number of sub concepts in this list (M) and the number of sub concepts, which belongs to GCL (S), are computed. Then, the summation of Ps of the concepts that belongs to SCL is computed ($\sum_{SCL} P_s$) as follows:

$$P_C = \frac{\sum_{SCL} P_s}{S} * \frac{S}{M} \quad (1)$$

4. Use the previously computed values to find the total ranking value according to the following equation:

$$\text{Ranking Value} = P_S * W_S + P_p * W_p + P_C * W_C \quad (2)$$

Where WS, WP and WC are the weights of self, parent and children component of the ranking value in order. The values of these weights will be determined empirically. The important notification about the equation (1) is the ratio (S/M), which represents how much the concept benefits from its children component. The maximum benefit occurs when S equals M. The detailed steps of CRA are presented in Figure 10

10. ASWSC: Automatic Semantic web services Classifier

The system contains predefined groups of services where a new registered SWS should be classified into one of these groups. This classification occurs by calculating certain values; each of them measures the relation between the new SWS and a specific group. Before starting the classification process, a list of concepts used by each group of services (GCL) is prepared. Each concept in this list is ranked to show its relative importance within the group using CRA. To calculate how strong a new SWS (S) is related to a group (G), the concepts used by (S) are defined then the summation of their ranking values in (G) is calculated. This value represents the measurement of the relation between the (S) and (G). This process is repeated for each group. The maximum calculated value refers to the best classification choice. ASWSC does not use any of the traditional classification techniques, which are based on neural, genetic or rough sets. ASWSC algorithm is an automatic algorithm used to classify SWSs and it depends on a novel concepts ranking algorithm (CRA). Figure 11 shows the detailed steps of the ASWSC algorithm.

```
Classification Algorithm
INPUT
    An unclassified semantic web service (S)
OUTPUT
    Discover a recommended group ResultGroup (G)
Terminology
Groups: list of available groups of services
GCL: list of <concept, Rank> pairs used by a group
SerCL: list of concepts used by inputs and outputs of the service that will be
    classified
SerGR: measurement of the strength of the relation between the unclassified
    service and the tested group
Max_SerGR: maximum value of SerGR
getRank: a function that accepts two parameters one of them is a GCL and the
    other is a concept (C) and returns with the ranking value of the concept C
    in GCL if exists otherwise it returns with zero.
ResultGroup: the name of the recommended Group.
STEPS
    SerCL = getConcepts(S)
    Max_SerGR=0
    ResultGroup ="no group"
    foreach (Group G in Groups)
        GCL= getConceptsList (G)
        SerGR=0
        foreach (Concept C in SerCL)
            SerGR = SerGR + getRank (GCL,C)
        next
        if (SerGR > Max_SerGR) then
            Max_SerGR = SerGR
            ResultGroup =G
        end if
    next
```

Figure 11: the classification algorithm

11. Empirical Experiments

11.1. Objectives

There are three objectives of the testing process, the first one is to prove the validity of using the parent and children components in addition to the self-component in our proposed concepts ranking algorithm (CRA). This is tested by using ASWSC to perform the classification with and without using the parent and children components. In other words, we compare the case of using self-component only (WS =1, WP =0, WC = 0) with the case of using all the ranking components with equal weights (WS =WP = WC= 0.33). The second objective is to find out by empirical trials the best values for ranking equation weights (WS, WP and WC). The third objective is to compare the accuracy of ASWSC with the other web services classifiers.

11.2. Performance criteria

The most common way to measure the performance of a multi-class classifier is to divide its task into multiple binary classification tasks [33],[34] in which the classifier has only two decisions, whether to accept the service in the class (group) or to reject it. There are some values that can be calculated to measure the binary classifier performance (Where the values of A, B, C and D are defined in Table 1):

$$\text{Precision (P)} = \frac{\text{Correct Classes Found}}{\text{Total Classes Found}} = \frac{A}{A+B} \quad (3)$$

$$\text{Accuracy (ACC)} = \frac{\text{Total Correct Decisions}}{\text{Total Number of Services}} = \frac{A+C}{A+B+C+D} \quad (4)$$

$$\text{Recall (R)}=\frac{\text{Correct Classes Found}}{\text{Total Correct Found}}=\frac{A}{A+D} \qquad (5)$$

$$\text{Error (E)}=\frac{\text{Total Incorrect Decisions}}{\text{Total Number of Services}}=\frac{B+D}{A+B+C+D} \qquad (6)$$

Table 1: All Possible Outcomes of a Binary Classifier.

Classifier outcome		Description (Number of)
A	True positive	services that are assigned correctly
B	false positive	services that are assigned incorrectly
C	True negative	services that are rejected correctly
D	False negative	services that are rejected incorrectly

In order to evaluate the total classifier performance, the previously mentioned values, which are calculated for each binary classifier should be averaged. For example, if three different classes are available, the macro-average of the precision can be represented mathematically as:

$$\text{Average Precision}=\frac{\frac{A_1}{A_1+B_1}+\frac{A_2}{A_2+B_2}+\frac{A_3}{A_3+B_3}}{3} \qquad (7)$$

11.3. Experiments Data and Results

For the testing purpose, we use the OWLS-TC v3.0 collection[35]. The collection targets to support the evaluation of the performance of OWL-S service matchmaking algorithms. It provides 1000 semantic web services written in OWL-S 1.1 from seven different domains (groups) (education, medical care, food, travel, communication, economy and weapons) as shown in Table 2.

Table 2: distribution of OWLS-TC services

Domain (Group)	Services
education	279
Medical care	73
food	33
travel	165
communication	58
economy	352
weapons	40
Total	1000

Three different experiments are designed using this collection. In each of them, we choose a percent of the available SWSs to be a training set (preclassified) and the rest is used as a test set (unclassified). The selection process of these services is done in a random manner taking into consideration selecting an equal percent from each group. Table 3 shows some information about each experiment.

Table 3: Experiments data

Experiment No	Percent of training set	Domain (Group)		actual number of Train set
1	60%	communication	34	597
		economy	211	
		education	167	
		food	19	
		medical	43	
		travel	99	
		weapon	24	
2	80%	communication	46	798
		economy	281	
		education	223	
		food	26	
		medical	58	
		travel	132	
		weapon	32	
3	90%	communication	52	897
		economy	316	
		education	251	
		food	29	
		medical	65	
		travel	148	
		weapon	36	

For each experiment, we test the performance of the "proposed classification and ranking algorithms (ASWSC and CRA)" using the previously mentioned performance criteria. We repeat this process more than 1000 times, in each time we change the values of WS, WP and WC randomly where their summation must equals one.

Table 4 shows a sample of the results of experiment number 3. It is worth to note that ASWSC fails to make any classification decision for two services (Regardless the values of WS, WP and WC) because the concepts used by these services does not belong to any GCL.

Table 4: sample of the results of experiment number 3

Trial id	Weights			Classification results			
	Ws	Wp	Wc	Right	Wrong	un classified	Total
1	0.38	0.48	0.14	91	12	2	103
6	0.12	0.16	0.72	89	14	2	103
12	0.18	0.5	0.32	88	15	2	103
78	0.64	0.12	0.24	86	17	2	103
577	0.37	0.09	0.54	87	16	2	103

ISSN 2229-5518

589	0.26	0.07	0.67	86	17	2	103
678	0.01	0.85	0.14	81	22	2	103
688	0.86	0.1	0.04	85	18	2	103
779	0.26	0.2	0.54	88	15	2	103
875	0.33	0.39	0.28	92	11	2	103

Figure 12 represents precision versus recall curve for experiment 3 trials. This curve is drawn using curve fitting and smoothing techniques provided by Eureqa2. The curve shows a proportional relation between precision and recall, which can be considered a good privilege for our classifier. This relation means that increasing the recall of the classifier does not affect the precision negatively.

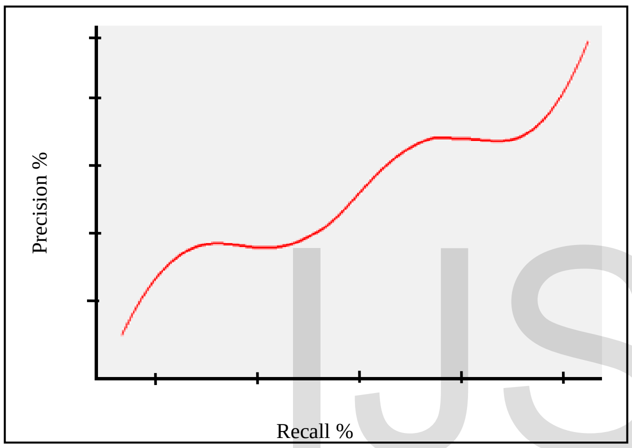

Figure 12: precision versus recall curve for experiment 3

Figure 13 represents a histogram that compares the accuracy between the case of using self component only (WS =1, WP = WC = 0) and the case of using all the ranking components with equal weights (WS =WP = WC= 0.33). It also compares these two cases with the case of using the best weights that we get from the empirical experiments. This histogram shows the positive effect of adding the parent and children components to the ranking algorithm. It is also noticed that there are small differences between the accuracy percentages in the case of using equal weights and in the case of using the best weights.

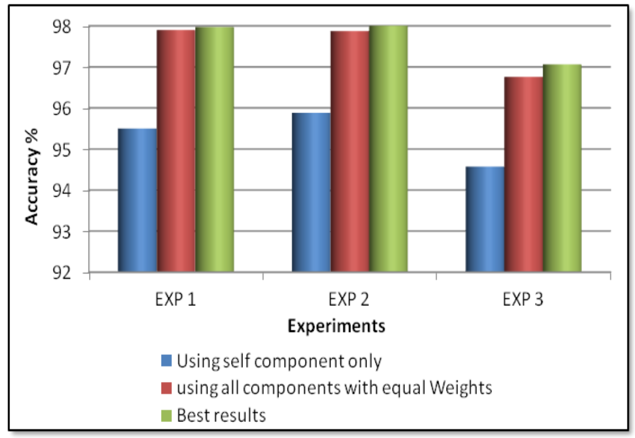

Figure 13: histogram of the accuracy in different experiments using different weights

The summary of the most significant experiments results is shown in Table 5. In all experiments, the percent of classification precision , accuracy and recall increases in the case of using equal weights when compared with the case of using the self-component only (WS =1). Also, the results show the high percentage of success of our proposed classification and ranking algorithms.

Table 5: summary of experiments results

Exp. No	**Cases**	**Classifier performance metrics**				**Weights**		
		P%	Acc%	R%	E%	Ws	Wp	Wc
1	self-component only	85.97	95.5	90.2	4.5	1	0	0
	equal Weights	93.14	97.9	94.2	2.1	0.33	0.33	0.33
	best	92.48	97.97	94.4	2.03	0.29	0.36	0.35
2	self-component only	86.88	95.88	93.22	4.12	1	0	0
	equal Weights	89.83	97.87	96.23	2.13	0.33	0.33	0.33
	best	90.6	98	96.43	2	0.41	0.34	0.25
3	self-component only	82.23	94.57	87.16	5.43	1	0	0
	equal Weights	85.05	96.76	90.83	3.24	0.33	0.33	0.33
	best	85.67	97.06	91.22	2.94	0.33	0.39	0.28
Average		87.98	96.83	92.65	3.17			

It is difficult to compare between the results of ASWSC and other classifiers due to the use of different datasets for empirical experiments which are designed and performed to evaluate the performance of these classifiers. Therefore, we just compare the accuracy of these classifiers, as stated by their authors, with the accuracy of ASWSC as shown in Figure 14. This comparison indicates that the proposed classification

[2] http://creativemachines.cornell.edu/eureqa

algorithm has the highest accuracy level when compared to the other web services classifiers.

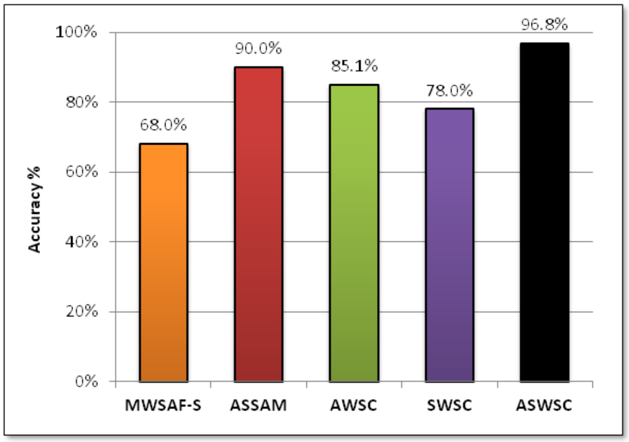

Figure 14: histogram of the accuracy of different web services classifiers

12. CONCLUSION

In this paper, we introduce a new framework to describe Semantic Grid Resources. As one of the most important semantic grid resource, SWSs is used to test the efficiency of this new framework. So, we test a new intelligent approach to classify the SWSs. The approach is based on a novel method for concepts ranking that measures the importance of each concept for each group of services. These ranking values are used to decide the most appropriate group for any unclassified SWSs. The empirical experiments show very encouraging results about our proposals.

ACKNOWLEDGMENT

The authors gratefully acknowledge the Taif University for its support to carry out this work. It funded this project with a fund number 2859-434-1.

REFERENCES

[1] I.Foster and C. Kesselman, “The Grid: Blueprint for a New Computing Infrastructure: Morgan Kaufmann”, 1999.
[2] Pieter Simoens, Filip De Turck, Bart Dhoedt, and Piet Demeester ,” Remote Display Solutions for Mobile Cloud Computing” , IEEE computer society, vol 44 Issue 8 , pp.46-53,2011
[3] Moreno-Vozmediano , "Key Challenges in Cloud Computing to Enable the Future Internet of Services", accepted for publication in IEEE Internet Computing ,2012
[4] Nathaniel Borenstein and James Blake ,”Cloud Computing Standards: Where's the Beef?” , IEEE Internet Computing , vol 15 issue 3 , pp74-78, 2011.
[5] http://imsresearch.com/press-release/Internet_Connected_Devices_Approaching_10_Billion_to_exceed_28_Billion_by_2020&cat_id=190&type=LatestResearch
[6] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web", Scientific American Magazine, vol. 284, no. 5, pp. 34–43, 2001.
[7] Michael C. Daconta , Leo J. Obrst and Kevin T. Smith, "The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management" , Book, Wiley Publishing 2003.
[8] G. Antoniou and F. van Harmelen, "A Semantic Web Primer", Book (2nd edition), The MIT Press 2008.
[9] H. Stuckenschmidt and F. van Harmelen, "Information Sharing on the Semantic Web", Book, Springer 2004.
[10] L. Li and I. Horrock. "A software framework for matchmaking based on semantic web technology". In Proceeding of 12th International World Wide Web Conference on E-Services and the Semantic Web (ESSW 2003), 2003.
[11] Jorge Cardoso, "Semantic Web Services: Theory, Tools, and Applications", Book, IGI Global 2007.
[12] M. Burstein, C. Bussler, M. Zaremba, T. Finin, M. N. Huhns, M. Paolucci, A. P. Sheth, and S. Williams, "A Semantic Web Services Architecture", IEEE Internet Computing , vol. 9, no. 5, pp. 72–81, 2005.
[13] Resource Description Framework (RDF), W3C Recommendation 2004, online :http://www.w3.org/RDF/
[14] RDF Vocabulary Description Language 1.0: RDF Schema (RDFS), W3C Recommendation 2004, online :http://www.w3.org/TR/rdf-schema/
[15] Web Ontology Language (OWL), W3C Recommendation 2004, online: http://www.w3.org/2004/OWL/, 2004 .
[16] D. De Roure, Y. Gil, and J. A. Hendler, "Guest editors' introduction: E-Science," IEEE Intelligent Systems, vol. 19, pp. 24 - 25 2004.
[17] C. Wroe, C. A. Goble, M. Greenwood, P. Lord, S. Miles, J. Papay, T. Payne, and L. Moreau, "Automating Experiments Using Semantic Data on a Bioinformatics Grid," IEEE Intelligent Systems, vol. 19, pp. 48-55, 2004.
[18] S. Cox, L. Chen, S. Campobasso, M. Duta, M. Eres, M. Giles, C. Goble, Z. Jiao, A. Keane, G. Pound, A. Roberts, N. Shadbolt, F. Tao, J. Wason, and F. Xu, "Grid Enabled Optimisation and Design Search (GEODISE). " UK e-Science All Hands Meeting 2002 Sheffield, UK, 2002.
[19] L. Pouchard, S. Bechhofer, B. Matthews, J. Myers, D. Snelling, and Y. Sure, "A Virtual Organisation Ontology", 2006, http://www.csm.ornl.gov/~7lp/onto-library/DagstuhlVirtualOrganization1.0.owl, (2006).
[20] Dieter Fensel, " Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce", Springer-Verlag, 2001.
[21] John Davies, Dieter Fensel and Frank van Harmelen ,"TOWARDS THE SEMANTIC WEB: Ontology-driven Knowledge Management" , Book, Wiley Publishing 2003.
[22] Ed Ort, "Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools", Sun Microsystems Technical Article, April 2005, online: http://java.sun.com/developer/technicalArticles/WebServices/soa2/.

[23] Andreas Heß, Eddie Johnston and Nicholas Kushmerick ,"ASSAM: A Tool for Semi-Automatically Annotating Semantic Web Services", in Proceedings of 12th International Conference on Web Technologies, pp. 470–475,2008

[24] Abhijit A. Patil, Swapna A. Oundhakar, Amit P. Shethand Kunal Verma, "METEOR-S Web Services annotation framework", In proceedings of the 13th international conference on WWW,ACM Press, 2004

[25] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web", Scientific American Magazine, vol. 284, no. 5, pp. 34–43, 2001.

[26] Derek Wright, "Cheap cycles from the desktop to the dedicated cluster: combining opportunistic and dedicated scheduling with Condor", Conference on Linux Clusters: The HPC Revolution, June, 2001,

[27] Klaus Krauter, Rajkumar Buyya and Muthucumaru Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing ", Software: Practice and Experience (SPE), ISSN: 0038-0644, Volume 32, Issue 2, Pages: 135-164, Wiley Press, USA, February 2002.

[28] Web Ontology Language (OWL), W3C Recommendation 2004, online: http://www.w3.org/2004/OWL/, 2004 .

[29] Protégé :open source ontology editor and knowledge-base framework, http://protege.stanford.edu.

[30] N. Nisan, S. London, O. Regev, and N. Camiel, Globally Distributed computation over the Internet: The POPCORN project, International Conference on Distributed Computing Systems (ICDCS'98), May 26 - 29, 1998, Amsterdam, The Netherlands, IEEE CS Press, USA, 1998.

[31] Y. Aridor, M. Factor, and A. Teperman, cJVM: a Single System Image of a JVM on a Cluster, Proceedings of the 29th International Conference on Parallel Processing (ICPP 99), September 1999, Fukushima, Japan, IEEE Computer Society Press, USA

[32] Alchemi Documentation : **Error! Hyperlink reference not valid.**

[33] Erin Allwein, Robert Shapire, and Yoram Singer. "Reducing multiclass to binary: A unifying approach for margin classifiers", Journal of Machine Learning Research, pp 113–141,2000.

[34] Chih-Wei Hsu and Chih-Jen Lin, "A comparison of methods for multiclass support vector machines", In IEEE TRANSACTIONS ON NEURAL NETWORKS, vol. 13, pp 415–425 , 2002.

[35] Database. MIT Press, 1998. OWLS-TC v3. 0, http://projects. semwebcentral. org/projects/owls-tc/, 2009.