Semantic Web-Based E-Counseling System

Toshika Khandelwal, Garima Joshi, Anish Singhania, Animesh Dutta

Abstract**—** Presently on Internet, it is difficult to find a resource which serves as one-stop destination for students regarding their career-related queries. Students who use Internet to search for help end up with large amount of information, mostly irrelevant. This happens because of current syntax-based web search which just matches words of the query and returns all string-matched results. Thus, this research work proposes a Semantic web-based e-counseling system. It captures the query from the student in Natural Language, understands the logic of the query and returns a reply just as a human counselor would, which helps the student to find a proper career option with less efforts and time. The Natural Language Query is parsed by a NLP Parser to form a data structure which is compared to the Ontology, a hierarchical set of concepts and relations of a domain, and only relevant options are presented to the student.

.

Keywords**—**e-counseling, semantic web, ontology, natural language processing

I. Introduction

A. Counseling System

A counseling system helps a student to select a career depending on the student's choice of subject and field of interest. A counselor assists the student personally and guides the student to the path that it should take.

In a general e-counseling system, the virtual guide of the student follows the syntactic web approach. The approach results in the student having a lot of unwanted and unorganized data. The final outcome is that the student is left to sort a substantial portion of the data by itself, which is not desired. So, the data needs to be organized so that such discrepancies can be avoided.

B. Semantic Web

With all the data that is available in the world today, there is an increasing need for the data to be organized so that it becomes machine processable. Here comes the concept of semantic web, which organizes the data so that it is machine processable and not just human interpretable. Semantic web is the idea of having data on the Web defined and linked in a way such that it is machine processable and not just available for display. For this to become true, underlying support systems and technologies should be designed to make the Web more meaningful and human-friendly.

C. Ontology

Ontology is a formal explicit specification of shared conceptualization. Ontologies form the backbone of the Semantic web; it facilitates machine understanding of the linked information resources. The various annotations on the semantic web form links between resources on the Web and connects them to formal terminologies and these connective structures are called Ontology. Formally, Ontology consists of concepts (classes) and relationships between these concepts (relations) in a hierarchical (or taxonomical) format; thus forming a vocabulary for the domain it is defined in, along with the computerized meaning of the terms in the vocabulary.

II. Related Works

A lot of work has been done earlier and is still in pursue in the field of Semantic Web technologies and Ontology. Work has been done on many layers of the Semantic Web, including content generation, web services and e-connections. [5-6] A Web-based geospatial application using Ontology and Semantic Web Services is designed to provide demonstration of using recent advancement in GIScience in other fields to achieve Semantic based GIS applications over cyber infrastructure which can further contribute to "development of community-guided cyber infrastructure".[10] Semantic Web technologies has been used for an improved exploration and rating of hotels for business customers in order to reduce the search time and costs, which, in turn, results in a huge benefit for the end-users.[9] Ontology has immense applications in the field of medical science and many new systems are being developed which uses semantic technologies as their basic architecture. An ontology based Holonic Diagnostic System(OHDS) has been developed which combines the advantages of holonic paradigm with multi agent system technology, for the research and control of unknown diseases.[8] There are many other applications of Ontology including e-learning systems which provides virtual classrooms, remote courses and distance learning.[7]

In our previous work, the idea of Ontology driven model has been proposed to make e-counseling a better experience for the students and to help them out in selecting an appropriate career option.[11] We expand the previous related

Toshika Khandelwal is with the National Institute of Technology, Durgapur, India, phone: (+91)7679986039; fax: (+91)3432547375; e-mail: toshika28@gmail.com.

Garima Joshi is with the National Institute of Technology, Durgapur, India;e-mail: gjoshi0311@gmail.com.

Anish Singhania is with the National Institute of Technology, Durgapur, India;e-mail: anishsinghania92@gmail.com.

Animesh Dutta is with the National Institute of Technology, Durgapur, India;e-mail: animeshrec@gmail.com

work by implementing the idea and bringing a change to the algorithm used by the Semantic Web search system. The new algorithm uses a different approach to the same problem idea which has not been implemented yet.

III. Problem Overview

A vast expanse of unorganized data is available to a student getting counseled through a general e-counseling system. The system returns the answers to a query in such a manner, and with such unnecessary details, that the student in-turn becomes a sort of self-counselor and is left with everything to search and sort again. A student is not able to express exactly what it wants to know as everything cannot be written down in a form-based format and the machine is not built to crack the logic of the question that is intended to be asked. Hence a lot of useful data, available from various resources, could have been used more efficiently if it had been organized properly but instead, is wasted.

The preceding works that are available try to solve this issue giving a few algorithms but these available algorithms also has some design issues that need to be addressed and hence nothing useful surfaces. In fact a new and better algorithm can be built for the same problem to increase the efficiency of the system to return a result and also to reduce the time complexity.

IV. Scope Of Work

In general, in an e-counseling system, a student is required to enter his choice of subject and field of interest in a form-based enquiry and a huge amount of data, both relevant and irrelevant, is returned to the student as a result. These results are not entirely according to the student's need as the form prevents the student from completely expressing itself. The vast expanse of data available on the web further adds to the woe.

The proposed system gives a real-life counselor-like reasoning and logical capability to the searching mechanism. The student now can enter a query according to its own needs and get a result that it wants. This allows the student to interact more with the system and hence a lot more queries are asked and results returned, in contrast to the form-based system wherein the extent of the queries is limited. This is the concept that is underlying the Semantic Web Technology.

The proposed idea uses forms a hierarchy of the available data, dividing them into classes and relating them through properties (relations) so that they can be semantically searched through Ontology. Now, the student queries to the system and the system forms a semantic hierarchy of the query after parsing the query and compares it with the ontology to return a result. The system helps the student by reasoning the question so that all the relevant results are returned. For example: If a student queries for 'Computer Science and Technology', all results elated to 'Computer' and 'Computer Science' will also be returned, helping the student by returning logical results.

V. System Architecture

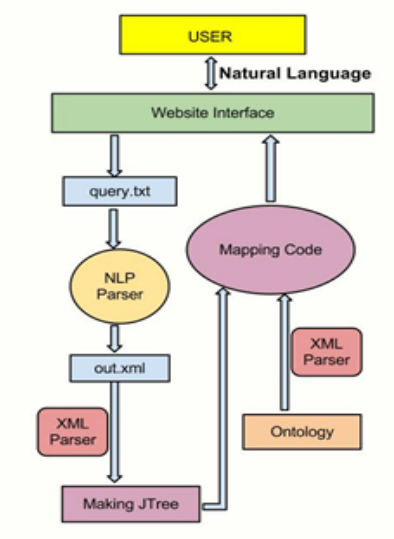

Fig. 1 System Architecture

A. Interface

A web-based medium between the user and the underlying semantic search system that accepts a query in Natural Language to be processed .The query is processed by the underlying system and finally the result is displayed. Therefore, the interface helps the user interact with the system.

B. NLP parser

Parser is a Java-based tool used to parse natural language query and to analyse the grammatical structure of the sentence. It groups together words to form phrases and resolves the different dependencies into various parts of speech. It parses the input in various output formats and saves the resultant output in a .txt or .xml file.

C. XML Parser

The XML Parser, SAX (Simple API for XML) is an event based sequential access parser API used to parse XML files. A parser that implements SAX functions as a stream parser, with an event-driven API. The user defines a number of call back methods that will be called when events occur during parsing. The SAX events include (among others) XML Text nodes, XML Element Starts and Ends, XML Processing Instructions, XML Comments. The minimum memory required for a SAX parser is proportional to maximum depth of XML file and the maximum data involved in a single XML event.

D. J-Tree

J-Tree is a class in the javax.swing package. It is a Swing component that displays a set of hierarchical data as a tree outline. Like most Swing components, it has a model, is highly configurable through the use of renderers, and fires events that can be handled. Each row displayed by the tree contains exactly one data item, which is called a node. Every tree has a root node from which all nodes descend. Branch nodes can have any number of children. Like any non-trivial

Swing component, the tree gets its data by querying its data model, where a specific node can be identified by a TreePath, an object that encapsulates a node and all of its ancestors.

E. Ontology

Ontology are logical systems that constrain the meaning of data. It provides logical linking of data along with its annotations. OWL makes it possible to describe concepts but it also provides new facilities. It has a richer set of operators-e.g. intersection, union and negation. It is based on a different logical model which makes it possible for concepts to be defined as well as described. Thus, Ontologies are considered as one of the pillars of the Semantic Web.

F. Mapping code

The Mapping code maps the tree, obtained from the parser output, with the Ontology. It matches the nodes of the output tree with the tags in the XML file of the OWL Ontology. The XML file is parsed using the SAX parser. The complete tree is traversed and the nouns are mapped with concepts (classes) and verbs with properties (relations) and then the result of the search is displayed with the help of the interface.

VI. IMPLEMENTATION

The Ontology is created using an open source tool called Protégé, the coding has been done in Java, XML parsing is done using SAX parser and parsing of natural language query has been done using Stanford parser.

The implementation of the system requires the following steps:

- Creation of Ontology in OWL/XML format using data from various resources available.
- Arranging individuals as classes and members, assigning hierarchy to various classes and defining relationships.
- Input query will be taken from the user via a Web-based interface and stored in a text file (file.txt).
- The Stanford Parser will take the file (file.txt) as input, parse the natural language query and store the output in an XML file (out.xml). The XML file will contain structural information about the sentence in format that gives the parts of speech of every word which helps to identify the type of each word.
- Output file from the Parser (out.xml) will be taken as input to the first algorithm and output in the form of a tree with words as nodes is formed.
- Tree formed in the first algorithm is taken as an input to the mapping algorithm, the code will map noun to the concepts and its annotations, and verb to the properties. When a match is found, the data associated with the individuals participating in the relation will be returned and stored in a data structure.
- Union or Intersection of the result from each branch will be done on the basis of type of query (and/or etc.) in the Mapping Code and the final result displayed to the user.

VII. ALGORITHM

A. Algorithm 1-Making a tree

```
Input: The output XML file returned by the Parser (out.xml)
Output: A J-Tree with the ROOT as its first node.
Data Structure Used: J-Tree
START:
Main()
   Integer nn,vb,sbn,cc <= 0
   for(s=0 to no. of words)
      Node firstword= listOfWords.item(s)
      Element firstword1=(Element) firstword.item(0)
      Nodelistfirstword2=firstword.getElementsByTagName
      ("word");
      String type=word.getAttribute("pos");
      If type="NNS"
         nn=1
         Tree_func(type,getNodeValue())
      Else if type="VBP" and nn=1
         Vb=1
         Tree_func(typr,getNodeValue())
      Else if type ="NNP" and nn=1 and vb=1
         Sbn=1
         Tree_func(type,getNodeValue())
      Else if type="CC" and  vb=1
         Cnj=1
         Node secondWord=listOfWords.item(s+1)
         String test = word.getAttribute("pos")
         If (test="VBP")
            Tree_func("TP0",getNodeValue())
         Else
            Tree_func("TP1",getNodeValue())
         End If
      End If
   End For
End Main()

Tree_func (String type, String word)
   Integer x=0
   DefaultMutableTreeNode parent = new
   DefaultMutableTreeNode("Root",true)
   If type="NNS"
      Add word as child of Root->Node 1
   Else if type="VBP"
      Add word as child of Node 1->Node 2
   Else if type="NNP" and x=0
      Add word as child of Node 2->Node 3
   Else if type ="TP0"
      Add word as child of Node 1 ->Node 2(a)
      X=1
   Else if type="NNP" and x=1
      Add word as child of Node 2(a)
   Else  type="TP1"
      Add word as child of Node 2
   End If
End Tree_func()
```

Explanation:

Step 1:A list of words is formed and all the elements with the tag name "word" are stored in a nodelist. The "pos" attribute of the word tag is referenced and the value of the tag stored in the string "type".

Step 2: First and foremost, the NOUN off the sentence is to be found. Therefore, the string "type" is equated against "NNS"(which symbolizes the noun in the subject of the sentence).Once the type matches, the variable nn (flag to know whether noun has been found or not) is set; and the function Tree_func() called so that the noun can be added to the tree.

Step 3:The function Tree_func() checks the type of the incoming argument and then add the node(NODE 1) as a child of the ROOT(NODE 0).

Step 4:Next, the main verb of the sentence is to be found. So,the type is equated against "VBP"(which symbolizes the main verb of the sentence). Once the type matches, the variable vb(flag to know whether verb has been found or not) is set; and the function Tree_func() called so that the verb can be added to the tree.

Step 4:In the function Tree_func(),the type is matched with "VBP"; and if found, the word is added as a child node(NODE 2) of the parent node(NODE 1).

Step 5: Next, in the main function, the object of the sentence is to be found. So, the type is equated against "NNP"(which symbolizes the subnoun or the object of the sentence). Once the type matches, the variable sbn(flag to know whether the object has been found or not) is set; and the function Tree_func() called so that the object can be added to the tree.

Step 6: In the function Tree_func(),the type is matched with "NNP";and if found, the word is added as a child node(NODE 3) of the parent node(NODE 2).

Step 7: Now, in the main function if a conjunction is found (matched with the type "CC"), then the type of the next immediate word is checked against a "VBP" or a "NNP". If the type is a "VBP", then the function Tree_func is called with the arguments "TP0" and the node value; or else with the arguments "TP1" and the node value.

Step 8:Now,in the function Tree_func(),if the type is matched with "TP0",the word is added as a child node (NODE 2(a)) of the parent node(NODE 1) ;and the variable x is set symbolizing that a new path of the tree is beginning. If the type is matched with "TP1";then the word is added as a child node (NODE 3(a)) of the parent node (NODE 2).

Step 9:The steps 2-8 are the repeated until the complete sentence is parsed to form a tree and the resultant tree is complete.

B. Algorithm 2:Mapping The Tree And Ontology

Input: Tree output from Algorithm 1.

Output: R1 containing keywords to the results required.

Data Structures used: Stack and String(character array).

START:

1. Identify all paths to leaf nodes.
2. For each path starting from the child of the Root node, repeat till leaf node is reached.
3. Find a concept(or its annotation) in the ontology which matches the child of the Root node, if a match is found store it in R, else continue to the next child of Root.
4. For the next immediate node, find a object property(or its annotation) in the ontology which matches the node, if a match is found store it in R, else continue to next immediate node.
5. For the next immediate node, find a subproperty of matched property in the ontology which(or its annotation) matches with the node, if a match is found get the range of individuals from the property.
6. Data to be return as result is defined here as individuals of the related class which will be stored in R(i).
7. After getting and storing the results in R(i) for each path, we display the result as union of all, if the conjunction is 'or' in the query and similarly as intersection if 'and' conjunction is there in the given query.

VIII. CASE STUDY

A. User Query

Which are the colleges that offer CSE and IT?

B. Parser Output

```
<corpus>
<s id="1">
  <words pos="true">
    <word ind="1" pos="WP">Which</word>
    <word ind="2" pos="VBP">are</word>
    <word ind="3" pos="DT">the</word>
    <word ind="4" pos="NNS">colleges</word>
    <word ind="5" pos="WDT">that</word>
    <word ind="6" pos="VBP">offer</word>
    <word ind="7" pos="NNP">CSE</word>
    <word ind="8" pos="CC">and</word>
    <word ind="9" pos="NNP">IT</word>
    <word ind="10" pos=".">?</word>
  </words>
</s>
</corpus>
```

C. Working of Algorithm

The user's query is a natural language sentence which will be sent as the input to the NLP parser and the parser returns the output in an XML file in words and tags format as shown in the code above. Now this file is the input of the tree making algorithm. The algorithm makes a list of all the "word" tags in the Nodelist. Now, tags are retrieved sequentially and the "pos" attribute of the tags is checked. Firstly, the SAX parser retrieves the string "WP"; which is discarded as it is of no use to the algorithm. Now the next tag, although being a verb "VBP", is also discarded because the subject noun of the sentence is not yet found. Next the tag "DT" is ignored as it is

not required by the algorithm. Next the tag "NNS" is received, and since it is the noun of the sentence and useful to the algorithm, it is sent to the function Tree_func() and is added as a child node of the Root node. Next the tag "WDT" is ignored as it is not useful. Next the tag "VBP" is received as is passed on to the function Tree_func(). It is useful as the noun has been received and the verb is required next by the algorithm. So it is added as a child node of the second node. Next the tag "NNP" is received and passed on to the function Tree_func(). It is useful as the noun and the verb have been received and hence the object of the noun is required by the algorithm. So this node is added as a child node of the third node. Next the tag "CC" is received and since it is a conjunction, the next immediate node is checked for a noun pr a verb. Since here the next immediate node is a noun, the noun is passed on to the function Tree_func() with the tag "TP1". In the function Tree_func(), since the variable x is 0, which symbolizes that no new verb was found after the conjunction, the noun with the tag "TP1" is added to the tree as a child of the third node. The tree is complete thereafter.

Once the tree is complete, it is passed as input to the matching algorithm. In the mapping algorithm, each node of the tree formed is checked against a concept or property of the ontology. The nouns (here- colleges, CSE and IT) are checked against concepts and the verbs (here- offer) are checked against the properties(relations) and the corresponding results(if applicable) are returned.

Now, after the results from the ontology is returned to the mapping algorithm, the conjunction of the sentence is checked. If it is a "and", then the intersection of the two results is displayed on the interface. If it is a "or", the union of the two results is displayed on the interface.

D.Final Output Tree

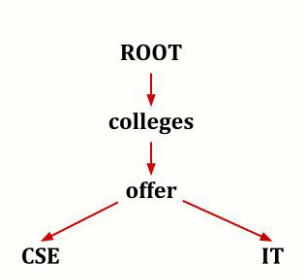

Fig. 2 Final Output Tree

CONCLUSION

In this work, we have introduced an e-counseling system for the benefit of the students. Thorough case studies have been done in this regard and the results have been quite satisfactory up to this stage. In the future, this work can be expanded to include larger domains of education and form a singular aide for all students regarding any of their career related queries.

REFERENCES

[1] Asunción Gómez-Pérez, Mariano Fernández-López, Oscar Corcho (2004). Ontological Engineering: With Examples from the Areas of Knowledge Management, E-commerce and the Semantic Web. Springer, 2004.

[2] Matthew Horridge,Simon Jupp, Georgina Moulton, Alan Rector, Robert Stevens, Chris Wroe :A Practical Guide To Building OWL Ontologies Using Prot´eg´e- 4 and CO-ODE Tools, Edition 1.1 , The University Of Manchester.

[3] K.K. Breitman, , M.A. Casanova and W. Truszkowski, Semantic Web: Concepts, Technologies and Applications, NASA Monographs in Systems and Software Engineering, Springer-Verlag London Limited 2007.

[4] Dieter Fensel, Holger Lausen, Axel Polleres Jos de Bruijn, Michael Stollberg, Dumitru Roman John Domingue, Enabling Semantic Web Services: The Web Service Modeling Ontology

[5] Jennifer Golbeck1, Bernardo Cuenca Grau, Christian Halaschek-Wiener, Aditya Kalyanpur, Yarden Katz, Bijan Parsia, Andrew Schain, Evren Sirin, and James Hendler, University of Maryland

[6] Natalya F. Noy, Semantic Integration: A Survey Of Ontology-Based Approaches, Stanford Medical Informatics, Stanford University

[7] Chakkrit Snae and Michael Brueckner, Ontology-Driven E-Learning System Based on Roles and Activities for Thai Learning Environment, Faculty of Science - Naresuan University, Phitsanulok, Thailand

[8] Maja Hadzic, Mihaela Ulieru, Elizabeth Chang, Ontology based Holonic Diagnostic System (OHDS) for the Research and Control of Unknown Diseases,Curtin University of Technology School of Information Systems

[9] Magnus Niemann, Malgorzata Mochol and Robert Tolksdorf, Enhancing Hotel Search with Semantic Web Technologies, Free University of Berlin

[10] Naijun Zhou, A Web-based Geospatial Application Using Ontology and Semantic Web Services, University of Maryland

[11] Sreeja Golui, Eshnna V.P.Ekka, Nidhi Jain, Animesh Dutta: An Ontology-Driven E-counseling System as an Implementation of Semantic Web Technology

[12] http://protege.stanford.edu/

[13] http://nlp.stanford.edu/software/lex-parser.shtml

[14] http://docs.oracle.com/cd/A97630_01/appdev.920/a96621/adx04paj.htm

[15] http://docs.oracle.com/javase/tutorial/

[16] http://www.w3schools.com/xml/

Toshika Khandelwal A 5th semester student pursuing B.Tech at National Institute of Technology, Durgapur, India in the field of Information Technology.

Garima Joshi A 5th semester student pursuing B.Tech at National Institute of Technology, Durgapur, India in the field of Information Technology.

Anish Singhania A 5th semester student pursuing B.Tech at National Institute of Technology, Durgapur, India in the field of Information Technology.

Animesh Dutta Received B.E. in Compute Computer Science and Engineering from Regional Engineering College, Durgapur, India and M.Tech from National Institute of Technology, Durgapur, India in the year 2002 and 2006 respectively. He is currently with the department of Information Technology of National Institute of Technology, Durgapur, India. He is pursuing his Phd work in the area of Software Engineering and Distributed Computing in the department of Computer Science and Engineering at Jadavpur University, Kolkata, India.

Development of Reliability Model for Complex Distributed Software System Considering Erasure Code, Physical and Logical Interlink

B.Ashwin [1] and Rajakannu Amuthakkannan[2]

***Abstract*—** Distributed System is the one where there are many computer systems that are connected via a network. The computers interact with each other in order to achieve a common goal. Software systems are constructed by integrating software subsystems by logical and physical interlinks. Software subsystem within a network suffers from the logical and physical interlink reliability where some papers have shown the reliability of the system by using it. But reconstruction of the erasure code in the distributed systems due to previous failure will also affect the reliability of the distributed software system. So, in this paper, a new model is proposed considering logical, physical interlinks reliability along with the erasure code reconstruction to find the reliability of the complex distributed subsystem.

***Keywords*—** Distributed systems, Erasure code, Logical interlink, Physical interlink

I Introduction

RELIABILITY is the ability of the computer program to perform its intended function and operation in a system environment without expecting a failure. Various models are available to assess the reliability of individual components. Shukla et al., [6] proposed a framework for assessing the reliability of individual components by test case execution and evaluation techniques. Wang et al [6] proposed a reliability model for complex distributed software system considering physical and logical interlink but without considering the sub system. The extension of Wang et al model is done by Amuthakkannan et al.,[9] by considering subsystem's physical and logical reliability. But Amuthakkannan et al did not consider the reliability change due to erasure code reconstruction. Actually, recent technology of networking allows designing distributed software system consisting of a number of software sub systems at various hierarchy levels which are physically interlinked to provide a solution for any software application. Normally erasure code is occurring due to transmitting a fixed set of data to multiple clients over unreliable links. So, erasure code correction is necessary to recover lost data in a complex distributed system because to rebuild the whole system is loss of money and time consuming. So, erasure code reconstruction concept is widely followed in the software industries to recover the failure data. But reconstruction of code may also affect the reliability of the system which was not considered previously by the researchers. In this paper the reliability changes due to erasure code reconstruction is the core content. By considering this, it is proposed to extent the Architecture based software reliability model to the distributed software system by incorporating interface and erasure code reliability. Enterprise resource management is taken as the case study and assessed using the newly proposed approach.

B.Ashwin is with Department of Computer Science and Engg, Kumaraguru College of Technology, Coimbatore, Tamil Nadu, India (corresponding author's phone: +917708467183; e-mail:b.ashwin1991@gmail.com).

Rajakannu Amuthakkannan is working as senior faculty at Department of Mechanical and Industrial Engineering, Caledonian College of Engineering (affiliated to Glascow Caledonian University,Scotland,UK), Muscat, Sultanate of Oman (e-mail:amuthakkannan@caledonian.edu.om phone:+968-98893272)

II Problem Background

The functioning of computers depends on the correct operation of their software. Unlike computer hardware, whose reliability has improved dramatically with the advances in integrated circuit technologies, the improvements in the reliability of software has been lagging. The engineering of reliable software is still a young and immature discipline. The main reason for the system failures are poor development practices, incorrect assumptions with regard to system requirements, Poor user interfaces, faulty hardware, inadequate user training /user error, Poor fit between systems and organisations. In distributed systems, there are many opportunities to fall the reliability because of complex software and too many hardware. So, in-depth research work is required to improve the reliability of system. The frequent software failure will lead to erasure code reconstruction in the concept of distributed system reliability which engineering is in infant stage at present.

III Literature Review

Few Models are available for estimating the reliability of component based software systems. Various approaches are available for assessing the reliability of individual components. Dolbec et al.,[1] provided a model which estimates the reliability of component based software system based upon component usage ratio. Krishnamurthy et al. [3] assessed the reliability of component based reliability estimation. This approach does not consider component interface faults. Wang et al.[6] proposed an approach which helps in estimating the reliability of a heterogeneous architecture using Markovian-chain properties. Hamlet, et al., [2] presented a theory which describes how a component developer can design and test their components to produce measurements that are later used by system designer to assess