ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 9(48), DOI: 10.17485/ijst/2016/v9i48/106686, December 2016*

Ontology based Retrieval of Components

Iqbaldeep Kaur*, Dimple Nagpal and Amit Verma

Computer Science and Engineering, Chandigarh Engineering College, Landran, Mohali - 140307, Punjab, India;
iqbaldeepkaur.cu@gamil.com, dimplenagpal009@gmail.com, Dramitverma.cu@gmail.com

Abstract

Objectives: Component based development commences the field of reusing the component rather than redeveloping it.To reuse the component, firstly the retrieval of component through repository has been done. **Methods/Statistical analysis:** There are many ways to retrieve the component but every scheme has its own pros and cons. To retrieve the best component, the relationships between components must be there. This paper presents the overview of whole process of component retrieval using ontology. **Findings**: Ontology helps in effective retrieval by providing relationships between concepts (classes), interfaces etc. **Application/Improvements:** Additionally this paper reviews what the component is, innovation in ontology, what are the reusable assets in component and how the component are formally described using languages are presented. The goal of ontology based retrieval of component is that best component retrieves from the interconnected repositories..

Keywords: Algorithm, Ontology Languages, Retrieval

1. Introduction

TNow a day's software component reuse is an important aspect. The programmer's perspective shifts the emphasis from building software to reusing software. Basically to reuse the component, the component should be stored in the repository or database. Storage of a component can be done by classifying the components by using various classification schemes, namely enumerated, faceted, attribute value, formal specification, etc. To retrieve the component we first have to classify the component. Every classification scheme has its own pros and cons.

Based on classification scheme, retrieval of component can be done by two ways, either the component exactly matches the query or the component matches the user query approximately. To retrieve the component the essential elements required are described in Table 1.

As said earlier, to retrieve the component, component should be placed in well defined repository having component specification, component description etc. But the problem in repository is component extendibility. When components are added in the repository, the retrieval process slows down. So, to overcome this entire problem a well defined repository having formal specification, shared conceptualization and having relationships between the concepts (classes) and interface (objects) are used, named as ontology[1].

Basically ontology helps in effective retrieval of component by providing concepts (classes), interface (objects) and the relationship between concepts and interfaces, axiom, rules etc. In this paper, we present ontology based retrieval of components process. Section 2 explains the basic concepts involved in retrieval of component using ontology. Section 3 explains various types of formal languages used in ontology. Section 4 explains the process of ontology based retrieval and at the end we wind up with the conclusion.

*Author for correspondence

Table 1. Essential element to retrieve component

Elements	Features
Database	Stores all the information.
Query Processing	User can query the component from repository.
Automated library system	Software with GUI, for browsing and searching database for components.
Friendly GUI	Client can query and retrieve the relevant Components.
Documentation	Provides all the details about particular software component.

2. Basic Concepts

2.1 Description of Component

In[2] Component is described as the nearly independent, practical and replaceable part of a system that fulfils a function in the context of well defined architecture.

Basically Component are not specifically made, they are made that much general so that they can be reused in future.

To reuse the component in future the component may be either composed, assembled and update so that user can benefit from these components and simply plug the components according to the needs and play the application with different components.

As described in[3] reusable components are self contained, clearly identifiable artefacts that describe specific functions and interfaces, appropriate documentation and define reuse status.

Table 2. Reusable asset in component

Reusable asset	Features	Represented By	Indexed
Executable code	Function that computes	Machine readable form	Functional properties
Source code	Embodies a function	Programming languages	Structural and functional properties
Design	They capture structural information	Patterns instantiated in different ways	Feature of the family problem they solve

Test Data	Perfectly legitimate reusable assets	Straightforward	Indication of function in the system
Documentation	Natural Language documentation is itself a reusable asset	Natural language	Asset that it document
Requirement Specification	Product of user requirement and recording them in notion	Natural Language and formal notation	Functional properties
Architecture	Defines the structure of software system	Specialized notions	Architectural features

Basically the reusable asset[4] in component are described in Table 2.

2.2 Component based Retrieval Process

TComponent Retrieval can be done by inputting a user query and according to the various classification scheme in which component is stored in the repository, the retrieval of components can be done. The basic retrieval process is shown in Figure 1.

As described in[5] there are three layers to retrieve the component. These are:

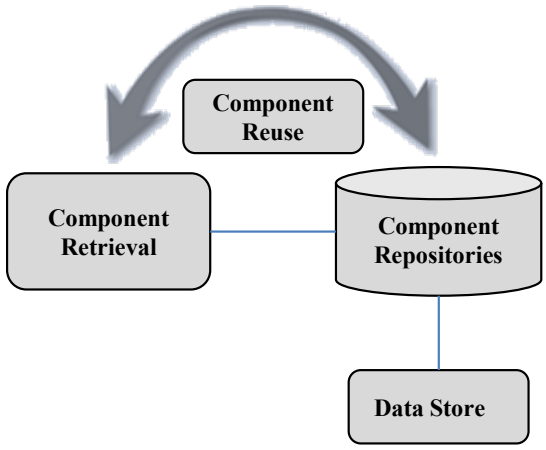

Figure 1. Basic retrieval process.

- User Layer.
- Interface Layer.
- Repository Layer.

User Layer: Basically it is a kind of web form where user can query and retrieve the results.

Interface Layer: It is responsible for processing of query in Natural language, semantic reasoning and helps in construction of retrieval expressions.

Repository Layer: All component information are stored in repository layer.There are basically 4 types of repositories namely:

- Describing Repository: It can provide some information such as environment, versions, brand, applied domain, functions to search software component.
- Component Repository: It stores components and retrieve the component can be done by services like download etc.
- Metadata Repository: It's a database created to store metadata. A metadata is far beyond from only the description of data.
- Ontology Repository: It store various repositories information (as per figure1) about component which helps in effective retrieval of components.

2.3 Innovation in the Ontology

The word ontology is defined in psychology which means the branch of metaphysics that is concerned with the nature of being. In computer science, ontology is defined as the systematic arrangement of components that is based on concept, interfaces and the relationship between concepts and interfaces. It is based on formal specification in such a way so that repositories of component can

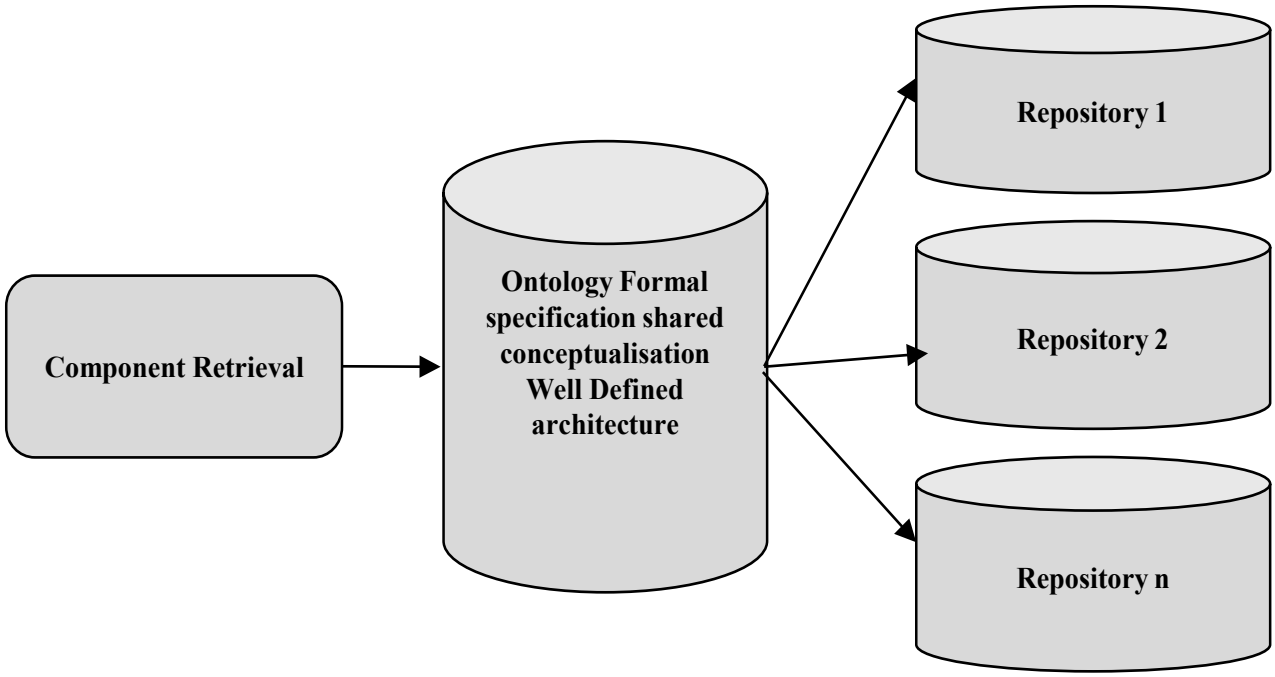

Figure 2. Basic process of ontology based retrieval.

be extended easily. An ontology is a collection of classes, objects and relationship between the two, which gives us precise output based on user query. Construction of ontologies can be done in two ways i.e., either domain dependent or generic ontologies[6].

In[7] ontology is described in formal language[7], having explicit knowledge and sharing the various concepts between components ·Basically it describes the whole component in a predefined manner, so that component can be searched easily. The basic process of ontology based retrieval is shown in Figure 2.

Basically ontology extends the various repositories by using semantics of component. Ontology adds basic information of component in such a level, so that there is no ambiguity of components in searching.

2.4 Ontology Components

The components of ontology are:

- **Individuals:** It is the basic component of ontology that contains concrete objects.
- **Classes:** These are the concepts that can be defined as either extensional or intensional. According to extensional definition, classes are abstract collection of objects. According to intensional, abstract objects are defined by value of aspects for having membership in the class.
- **Attributes:** Attributes can be a class or individual.
- **Relation:** Describes the connection of one object with the other.
- **Function terms:** Certain relation that helps to form a complex structure which can further be used in a statement in place of an individual.
- **Restriction:** Description are stated formally of the truthiness in order for some affirmation to be accepted as inputs.
- **Rules:** A logical inference that comes to a conclusion by using antecedent-consequent statements.
- **Axioms:** Include statement asserted as a priori knowledge.
- **Events:** The changing of attribute or relation.

3. Formal Description of Component

3.1 Specification/Description Language

As described in[8] specification language in computer science is a language used to describe a component during analysis and design at much higher level than a programming level· Specification languages are not directly executed.

3.2 Catalogue Language

The catalogue language of a component consists of the basic information of component that includes component name, version etc. It also includes how the components are presented, and the kind and hierarchy of components.

3.3 Ontology Development Language

Ontology language is formal language used to construct ontologies that are based on description logic and first order logic. The description of various development languages are shown in Table 3.

Table 3. Description of various languages in ontology

Language	Description	Based on	created for
KIF	Knowledge interchange format	First Order Logic	diverse knowledge related system

Table 3 Continued

LOOM	Knowledge representation language	DLs and production rules and general Knowledge bases	Automatic classification of concepts
OCML	Options configuration modeling language	Ontolingua	developing executable ontologies and models
FLogic	Frame Logic	-	Merging frames for representing various concepts, axioms etc
SHOE	simple html ontology extension	Ontology -Extension	To address unique problem semantic on the web
OML	Ontology Markup language	SHOE(partially)	Rich knowledge capabilities
XML	Extensible markup language	HTML	To display data
X.O.L	Exchange Ontology Language	XML	Development of ontologies by using any tool.
R.D.F	Resource Description Framework	XML(homogeneous, interoperable manner)	Modeling of information that is implemented in web services that represents services, while information is not clear.
RDFS	RDF schema	RDF	Providing basic language for the description of ontologies.
O.I.L	Ontology Interference Layer	DL+RDFS	Providing Efficient Reasoning support, Formal semantics and exchange of syntactical notation .
OIL+DAML	Ontology inference layer+DARPA agent Markup Language	OIL	Allow representation of Concepts, Taxonomies, Function ,Binary relations and instances
OWL1	Ontology Web Language1	DAML+OIL	-
OWL2	Ontology Web Language2	OWL 1	Adding richer datatypes to the types of OWL
CycL	Formal Language	First order predicate logic	To represent the knowledge of Cyc Knowledge Base

4. Ontology Based Retrieval Process

The retrieval of component can be done by the following steps are shown in Figure 3:

- Firstly, with the interface of the system, user inputs the query.
- Then from that query after stop word removal, query terms are extracted.
- Query interface refine the terms so that they can match with the describing terms in repository and retrieves best component. Then the parsing of query is done.
- Then query matching is done by converting the parsed text into web ontology language to ensure the correct terms are used in the query.
- Then the retrieval of component including specification and description are done, by matching the user query with the terms in the repository.

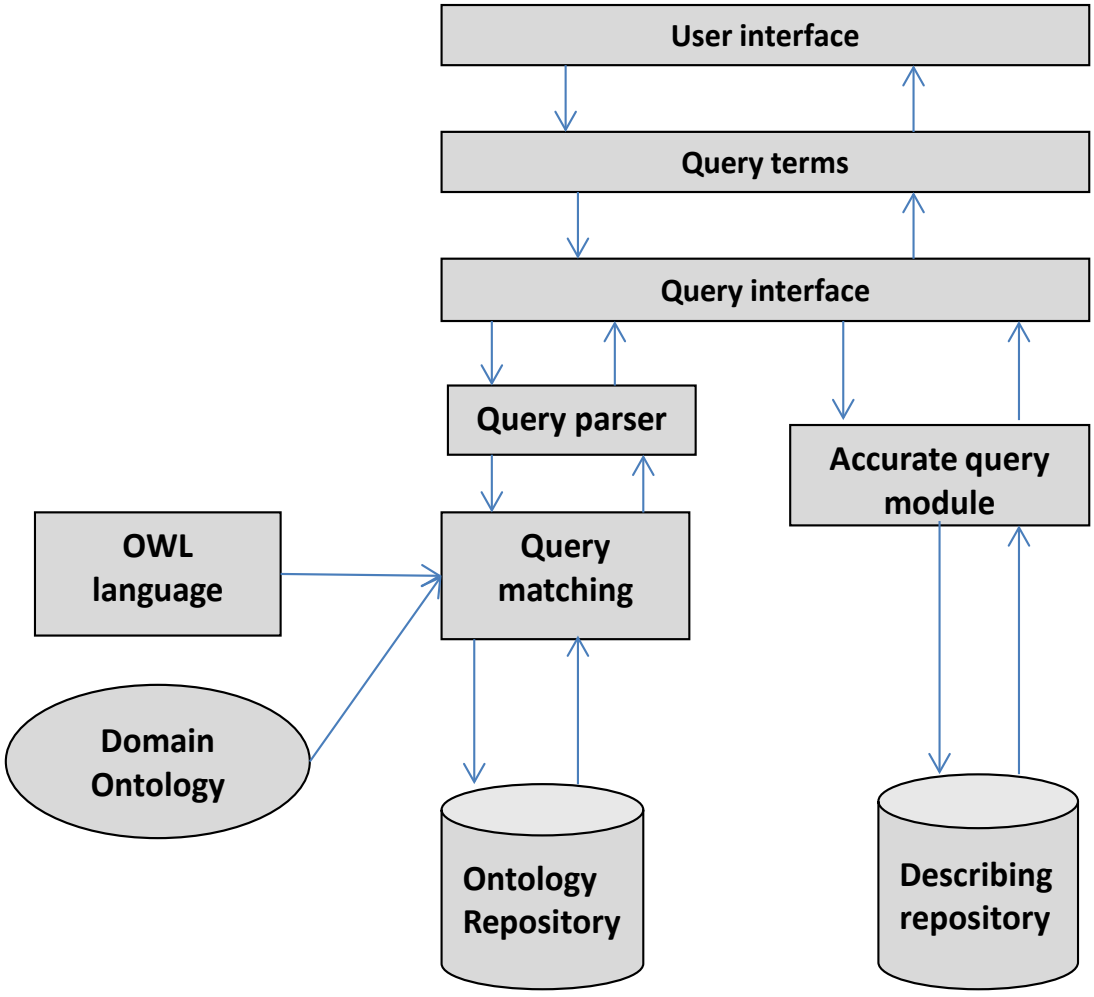

Figure 3. Ontology based retrieval process.

Ontology Based Retrieval Process

Step 1: Putting the component in repository and classifying them.

In order to retrieve the component, it should firstly be stored in the repository. The algorithm for storing the component is shown below:

In this algorithm as per Figure 4 building of component instances and relations among various components instances are described.

As described in[9] storing of component can be done by classifying the component. The classification of component can be done on the basis of, that whether the required component is class, relation, object, axioms, instance etc. The algorithm for classification of component is shown in Figure 5.

Inserting Algorithm

Let the component is com which is to be inserted in the repository

Firstly produce instance of the component.Let the instance be compInstance

Now filing the compInstances with description values(DV)

For description values,choose a class DO for every part in domain ontology

If DO does not contains instances

Create compInstance for every class DO

Endif

Add com to the instance list

End For

Figure 4. Algorithm for inserting the component in repository.

Classification Algorithm

```
Extraction Algorithm
{
Input the Program name called pfname
While s = GetLine(pfname)<>null
[Repeat steps 5 to 18]
If Contains(s,"class")
{
Entry=Entry +1
mname=Extract Entry(s)
m= ToEntry(mname)
Methods =GetMethods(m)
TotalMethods =TotalMethods+length(Methods)
Attributes=GetAttributes(m)
TotalAttributes=TotalAttributes +length(Attributes)
Axioms=GetAxioms(m)
TotalAxioms=TotalAxioms +length(Axioms)
Relations=GetRelations(m)
TotalRelations=TotalRelations+length(Relations)
}
Return(Module, TotalMethods, TotalAttributes,TotalAxioms, TotalRelations)
}
```

Figure 5. Algorithm for classifying the component.

After classifying the component we have to search and retrieve the component from repository

Step 2: Searching and retrieving the component from repository:

Searching Algorithm

One dimensional array user[] is declared for storing the words of user quer query

For(i=0;i<n;i++) wheren is the total no. of components in repository

Set LOOP[i]=0

Set j=1

The user inputs the query

While(user query=/= describing terms)

{ Read user[i]

i=i+1

}

For(j=0;j<=n;j++){

Dt is the no. of accurate describing terms.Set dt=j-1

}

For(j=0;j<=m;j++)

{

Calculate the characters in user[j].Let it be N

Compare characters of user query with the describing terms.Let no. of matched terms be N

Now calculate Z=M/N.

LOOP[i]=LOOP[i]+Z

}

Sorting of LOOP column in descending order in the table.

Set j=1

while(LOOP[j]=/=0)

{

Print STORED_COMPONENT[j]

Calculate j=j+1

}

Exit

Figure 6. Algorithm for searching of component.

As described in[10] searching helps in effective retrieval of component**.** Algorithm for searching and retrieving the component is shown in Figure 6.

In this algorithm firstly the array USER for multidimensional input is declared. There can be n no. of array according to the given queries. These array user inputs are then matched with the terms in the repository (where describing terms are stored). When user array matches with the describing term, the value of corresponding loop are incremented by 1. When the word in user input suppose n partially matches with the describing terms in repository suppose m. so the fraction of m/n decided the ranking of component. If the fraction output is one then that component is most relevant and ranked the first in the list and vice versa.

5. Conclusion

In this paper we present retrieval of component by classifying the component. The concept ontology based retrieval of component stores the information of component in a formal, explicit way. Basically Ontology adds the interrelationships between components by using semantics. Ontology in combination with semantic provide formal, explicit, shared conceptualization of component. The paper firstly describes the basic concepts in retrieval of component. After that formal language is presented in which components are described and at the end ontology based retrieval of component with various algorithms are shown.

6. References

1. Gupta S, Kumar A. Reusable software component retrieval system. International Journal of Application or Innovation in Engineering and Management. 2013; 2(1):187–94.
2. Roger PS. Software engineering: A practitioners approach. Palgrave Macmillan; 2005. p. 54.
3. Pandove D. Designing an interface for effective retrieval of components [PhD diss]. Patiala: Thapar University; 2010.
4. Huang JP, Yong PC, Huang K. Presented on topic an ontology-driven paradigm for component representation and retrieval. 9th International IEEE Conference on Computer and Information Technology; 2009. p. 187–92.
5. Khan L, Hovy E, McLeod D. Retrieval effectiveness of an ontology-based model for information selection. Very Large Data Bases (VLDB) Conferences; 2004. p. 71–85.
6. Usha GSY, Duhan ND, Jain V, Murthy BK. Presented paper on topic, development and visualization of domain specific ontology using protégé. Indian Journal of Science and Technology. 2016; 9(16):1–7.
7. Slimani T. Ontology development: A comparing study on tools, languages and formalisms. Indian Journal of Science and Technology. 2015; 8(24):1–12.
8. Catherine R, Kang MA, Corcho O. An introduction to ontologies and ontology engineering. London: Springer; 2011. p. 9–38.
9. Sandhu K, Gaba T. A novel technique for components retrieval from repositories Compusoft. An International Journal of Advanced Computer Technology. 2014; 3(6):1–9.
10. Singh. An experiment in software component retrieval based on metadata and ontology repository. International Journal of Computer Applications. 2013; 61(14):1–8.