ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Normalization For Removing Dependancies In XML Data

Ms. Jagruti S Wankhade *, Mr. Omkar V.Chandure
IT, JDIET Yavatmal, Maharashtra, India

Abstract

Designing a well-structured XML document is important for the sake of readability and maintainability. More importantly, this will avoid data redundancies and update anomalies when maintaining a large quantity of XML based documents. In this paper, we propose a method to improve XML structural design by adopting graphical notations for Document Type Definitions (GN-DTD), which is used to describe the structure of an XML document at the schema level. Multiples levels of normal forms for GN-DTD are proposed on the basis of conceptual model approaches and theories of normalization. The normalization rules are applied to transform a poorly designed XML document into a well- designed based on normalized GN-DTD, which is illustrated through examples.

Keywords: XML Document,Normalization,Diagramatical Representation of XML data,Normalization.

Introduction

What is XML:

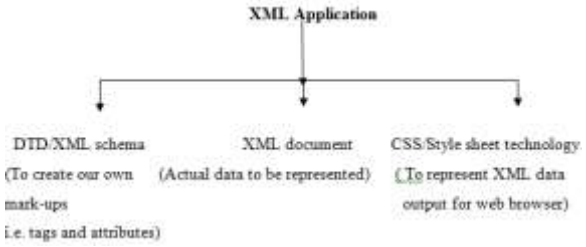
XML is nothing but the Extensible Mark-up Language. **Extensible Mark-up Language** (**XML**) is a mark-up language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications. It is a data representation and data communication standard for distributed application. it is used to store the data like data.

Extensible means:

- User Defined
- Customized
- Can be easily modified
- Dynamic

The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services. We can change its scope and maintenance according to our need called as customization.XML represents the new era for the web by transmitting structured data.XML is a specification for designing the mark-up languages. In other words XML is metalanguage.XML opens the door for carefully describing data and relationships between pieces of data. One of the main goals of XML is to separate content (data) from presentation in web documents.

The basic building block of an XML document is the entity, which content either parsed or unparsed data. Parsed data consists of character that are considered character, data or mark-up and that are processed by XML processor. Unparsed character data is handled as raw text and isn't processed. Mark-up is used to provide a description of a document's storage structure and logical structure. An XML processor is a software module that reads an XML document and provides access to its content and structure.XML parser is the component of XML processor that analyses XML mark-up and determine the structure of the document data.To create one application in XML document we have to create three programs.



DTD (Document Type Declaration):

Structure of an XML document is determined by DTD.IT is the actual grammar located within DTD.DTD is used to specify information about a document, including the document's root element and declaration. It is very important in establishing whether document is valid or just well-formed. Following are three main tasks carried out by a DTD.

- Specifying the documents Root element.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

- Defining element, attributes and entities specific to the document.
- Identifying an External DTD for the document.

DTD are the means of establishing a schema for XML document.DTD relay on a specialized syntax for describing the structure of XML vocabularies.DTD syntax is compact, the manner in which it describes document structure is rather cryptic and unintuitive.DTD can performed one or both of the following functions:

- Directly contain mark-up declaration in the prolog of the document that are collectively known as internal DTD subset describes structure specific to a given document.
- Reference external mark-up declarations that are collectively known as External DTD subset which typically describe general document structure for a class of document.

XML Schema:

- XML Schema is a new concept to defining the schema for XML documents that uses a XML vocabulary called XML data.
- It is very similar to DTD, in that it is used to establish the schema of a class of XML documents.
- XML schema describes elements and their content models so that documents can be validated.
- This allows XML processors to perform data content validation, which is an extremely significant benefit of using XML schema instead of DTD.
- An XML schema is that they are expressed in XML syntax.
- XML schemas present an open-ended data model, which allows extending vocabularies and establishing inheritance relationship between elements without invalidating documents.
- XML schemas support namespace integration, which allows you to associate individual nodes of a document with type declarations in a schema.
- XML can be parsed and manipulated just like any other XML document.

Cascaded Style Sheet (CSS):

- CSS is the style sheet language designed to provide a means of styling HTML documents, thereby allowing web developers to separate content from presentation.
- CSS is much simpler to learn and use than these approaches ,which makes it an ideal technology for styling HTML documents.
- When an CSS style sheet is applied to XML document it uses the structure of the XML tree as the basis for applying style rules.

Extensible style sheet language (XSL):

XSL is the style sheet technology that addresses two aspects of XML document rendering:

- Transforming documents from one type to another (XSLT).
- Styling documents according to formatting rules(XSL formatting Object).
- XSLT and XSL formatting objects are implemented as an XML vocabulary consisting of relevant elements and attributes for carrying out each portion of document rendering.
- When an XML document is transformed using XSL ,the transformation applies to the logical tree structure of XML document i.e. XML data not an XML document.

With wide exploitation of web and accessibility of huge amount of electronic data XML has been used as a standard means of information representation and exchange over the web.XML is currently used for many different types of appli cation which can be classified into two main categories, they are

Document centric XML:

It is used as a mark-up language for semi structured text documents with mixed content elements and comments; the order of sibling is significant.

Ex: User's manual, webpage etc.

Data centric XML:

It is consists of regular structured data for automated processing and there are little or no more element with mixed content, comment and processing instruction.

Ex:Geography,e-commerce

In this project we are focusing on Data centric application. In data centric applications, data redundancy and update anomalies are the most significant problem when data centric document applications are presented in XML, it is unavoidable that data redundancy and update anomalies will appear. Data redundancy and update anomalies occur in XML documents if their type structures, such as DTDs, are not well-formed. These problems are similar to those in relational databases [1, 6, 23].

In this project we are going to use **XML** file as a basic element for fulfilling the project requirement. As we better know because of the tree structure of XML document it obviously contains the redundant data, update anomalies due to DTD are not well formed. It surely creates the negative approach about XML document and also we can say that this is the one of the major issue that XML is not more popular than other languages, but it is one of most useful language for electronic data transfer and which is one of the mandatory and should be having highest level of security during data transfer in the world wide web.

To avoid this and to create more attractive approach we are going to create diagrammatical structure to XML document ,so that we can fast find out the redundant data, then applying different levels of normal form according to the dependency available in the XML document. Though we are creating the diagrammatical structure to represent the

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

XML document in the first level i.e. in the DTD formation, we can easily find out whatever the redundancy is available in document so that we can remove it with normalization so it automatically avoids the negative impact on the mail XML file and it improves the performance of the overall document.

To better understand; let us consider a university database which describes departments, courses, students, lecturers and their relationships as shown in the following figures (Figure 1 and Figure 2).

The DTD shown in Figure 1 has the following information:

- Each department offers many courses indicated by the notation *.
- Every course is described by the attribute course no (cno), title and numbers of students taking the course.
- Each student has a student number (sno), first name or last name as optional and an assigned lecturer.
- Each lecturer has his/her number (tno), and name (tname).

```
<! DOCTYPE department [
<! ELEMENT department (course*)>
<! ELEMENT course (title, student*)>
<! ATTLIST course cno ID #REQUIRED>
<! ELEMENT title (#PCDATA)>
<! ELEMENT student (fname|lname?,lecturer)>
<! ATTLIST student Sno ID#REQUIRED
<! ELEMENT fname(#PCDATA) >
<! ELEMENT lname(#PCDATA) >
<! ELEMENT lecturer (tname)>
<! ATTLIST lecturer tno ID #REQUIRED>
<! ELEMENT tname (#PCDATA)>
]>
```

Figure 1:DTD

```
<! DOCTYPE Department [
<Course>
<course cno = "csc101">
< title > XML database </title>
< student >
<student sno = "112344">
<fname> David</fname>
<lname> Grey </lname>
<lecturer>
<lecturer tno = "123">
<tname>Bing </tname>
</lecturer>
</student>
< student >
<student sno = "112345">
<fname>Helen</ftname>
<lecturer>
<lecturer tno = "123">
<tname> Bing </tname>
</lecturer>
</student>
</course>
<course>
<course cno = "csc102">
< title > Z Formal Database </title>
< student >
<student sno = "112344">
<fname> David</fname>
<lname> Grey </lname>
<lecturer>
<lecturer tno = "123">
<tname>Bottaci</tname>
</lecturer>
</student>
< student >
<student sno = "112345">
<fname>Helen</ftname>
<lecturer>
<lecturer tno = "123">
<tname> Bottaci</tname>
</lecturer>
</student>
</course>
</Department>]
```

Figure 2: XML document conforms to DTD in Figure 1

Any XML document that satisfies and conforms to this DTD is likely to contain data redundancies which may lead to update anomalies. For example, as shown in Figure 2, the lecturer named **Bing** who teaches the same **course number (cno) csc101** is stored twice, which will lead to the problems of redundant data which creates negative impact on XML document. To avoid such problems, a set of rules should be provided when designing a DTD for XML documents. Set of rules is nothing but the

- Creating the DTD,from that we create XML file and provide diagramatical form
- Finding redundant data which is responsible for anomalies
- Try to remove it with different levels of normalization rules

In this way we are going to create normalized XML document which will improves the overall performance of XML document also provide one approach to design attractive structure to the XML document which attract the web developer to use the it.

Related work

Two main approaches have been applied by XML database researchers to design non- redundant XML

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

documents which is a conceptual data modelling approach [7, 12, 15, 17,] and normalization theory [1, 11,16,]. Both of these approaches are based on relational database design theory [4, 5].

It provides some set of rules to represent DTD into the textual representation they are as follows:

DTD, D= (E,A,P,R,r) where

- ***E*** is a set of element sets
- ***A*** is a set of attributes using and illustrated as symbol @ and PCDATA as S
- ***P*** is a mapping from a set of elements to the children of the element set, indicating the semantic constraint of children (* ,?, +)
- ***R*** is a mapping from a set of elements to a set of attributes
- ***r*** is a root element set.

For example,consider the example in fig.1.

The set *E* contains all element sets such as department, courses, title, fname, lname, lecturer and tname. The set *A* contains all attribute sets such as @sno, @cno, @tno.

P maps from the element set to the children of the element set such as *P*(department)

R maps each element set to its attribute such as *R*(course) is sno, *R*(student) is cno and *R*(lecturer) is tno. The root element *r* is mapped to department.

With the help of this different redundancies are found and normalization algorithm is use.The input to this algorithm is a DTD and XFD and output is new normalized DTD.

This algorithm consists of two rules:

- Create new element types and
- Moving an attribute.

Using this algorithm, the following new DTD will be derived after eliminating XFD. The algorithm will be repeatedly applies until the DTD is in XNF.

```
<!DOCTYPE department[
<!ELEMENT department(course*, tinfo*)>
<!ELEMENT course(title, student*)>
<!ATTLIST course cno ID #REQUIRED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT student(fname|lname?,lecturer)>
<!ATTLIST student  Sno ID #REQUIRED>
<!ELEMENT fname(#PCDATA) >
<!ELEMENT lname(#PCDATA) >
<!ELEMENT lecturer(EMPTY)
<!ATTLIST lecturer  tno ID #REQUIRED>
<!ELEMENT tinfo (tname)>
<!ATTLIST lecturer  tno ID #REQUIRED>
<!ELEMENT tname (#PCDATA)>
  ]>
```

Figure 3: New DTD after eliminating XFD

Both approaches have a remarkable impact on XML document design. Each of them is complemented by the other in XML design research. In contrast, an ORA-SS data model is proposed to assist in XML document design at the conceptual level. The hierarchical structure of the object class is clearly shown in ORA-SS model. This approach followed from the ER normal form [15].This approach is guaranteed to produce a redundancy-free and compact database.

Analysis of problem

The objective of this work is to provide a methodology which simplifies the process of designing a non-redundancy XML document. To achieve this, a conceptual model called GN-DTD is proposed. GN-DTD is a graphical modelling approach for describing both DTD and XML documents. For GN-DTD itself, we define a complete set of syntax and structure which incorporates attributes, simple data type, complex data type, and types of relationship among them. Furthermore, semantic constraints are also precisely defined in order to capture semantic meaning among those defined objects.

In this work, we present normal forms for GN-DTD based on both Arenas and Libkin's rules [1] in a simpler form to allow users/designers to find an 'optimal' structure of XML elements/attributes. This will produce a correct, complete and consistent representation of the real world XML data which may benefit the users. We ensure that XML file mapped from GN-DTD are similar to XNF [1]. Finally, we propose normalization and mapping rules to transform from normalised GN-DTD back to its new DTD.

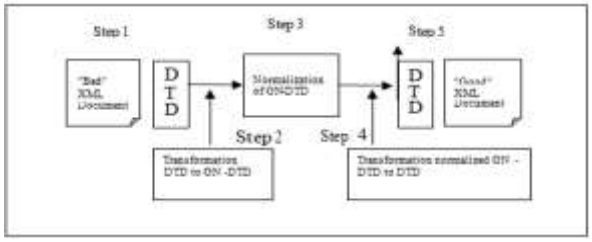


Figure 4: XML document design process

From the above diagram we can easily identify the main objective of this project is to create redundant free XML document .To do this four different steps are given as follows

Step1: Use XML document which contains the data dependency which will lead to the redundancy in the document.

Step2: Transform available XML into conceptual model i.e. into diagrammatical form so that we can easily find out the data dependency in the document.

Step3: Different levels of normalization algorithms are provided.

Step4: On the basis of document requirements appropriate normalization algorithm is used to remove the data

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

redundancy. i.e. transform un-normalized XMLfile into Normalized XML file which is our main output.

Data Dependency in GN-DTD:

This section gives the details about the dependencies available in the graphical structure of the XML. Data dependencies can be categorized into

- Key Attribute:
- Functional Dependency:
- Relationship Dependency:
- Global Functional Dependency:
- Functional Transitive Dependency:
- Functional Partial Dependency:

Normal Form for GN-DTD:

To avoid all the above dependency from XML document Different normal forms are given. With the help of this normalized XML document can be easily transformed into Normalized form. Following are the details of the different normal forms:

The First Normal form (1XNF GN-DTD):

GN-DTD is in first normal if and only if:

- Only one value for each simple element node or attribute node of GN-DTD can be stored. If there is more than one value, we must add some new element nodes or attribute nodes to store them.
- The root element of a GN-DTD model should be located at level 0 and the cardinality of the root element node must be one.
- Each set of complex element node in the GN-DTD has at least one key attribute node.

The Second Normal Form (2XNF GN-DTD):

The GN-DTD is in second normal form if and only if:

- GN - DTD is in 1XNF.
- There is no nested binary inheritance relationship or ternary inheritance relationship under many-to-many or one -to-much inheritance relationships with the following condition:

The Third Normal Form (3XNF GN-DTD):

GN- DTD is in third normal form if and only if:

- GN-DTD is in 2XNF.
- There exists no nested inheritance relationship type of *n*-ary many-to-one or many-to-many under a one-to-many inheritance relationship set in GN-DTD and the following conditions are satisfied:

Normal form GN-DTD (NF GN-DTD):

GN- DTD is in Normal Form if and only if:

- GN-DTD is in 3NF.
- There are no global dependencies between attribute and simple element of complex element Nodes under nested one-to-many or many-to-many inheritance relationships.

Proposed work/implementation

In this title we are going to see step by step execution of the actual project work, how we read the regular XML file ,how can we find out the redundancies in the XML document, How we use the Excel sheet to represent these dependencies and how we apply different levels of normalization algorithm to remove redundancies, all these we are going to see in this section as follows

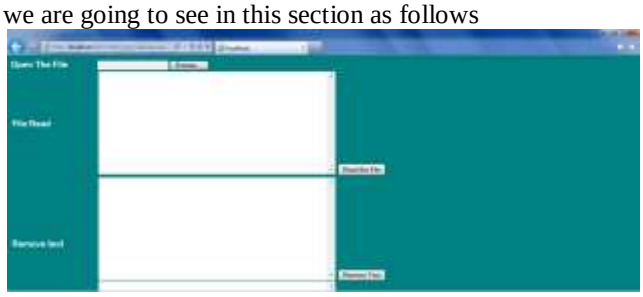


Figure 5:Default page

This page contains detailed information about project.As we can see from the page that first it contain

- open file option:
  With the help of this we can open any XML file
- Read File:
  In we extract XML file in the field area which is open by open file option
- Remove text:
  In this we are separating XML tags from Data values
- Normalization:
  In this we are transforming the extracted XML file into the graphical format and applying normalization

The main important feature is that each text represents the Execution Time

When we click on the Browse we can easily browse any XML file means we find XML path information and with the help of Read file option we read XML file from the specified path

Figure 6: Open file

Next option is about removing the data values from the XML tags which we are using for the graphical representation of the XML document

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Figure 7: Remove Text

After removing data values from the tags ,next operation is about the normalizing the XML document and representing it with help of Excel sheet .when we click on the **Normalization** button one message box is open which will ask about the replacement of the XML document ,

Figure 8: Normalization

When we click on the Yes option a message is generated from the web page which shows Execution time of the Normalization button

Figure 9: Execution Time

It also ask about whether you want to Open, Save, SaveAs. You can choose an appropriate Option

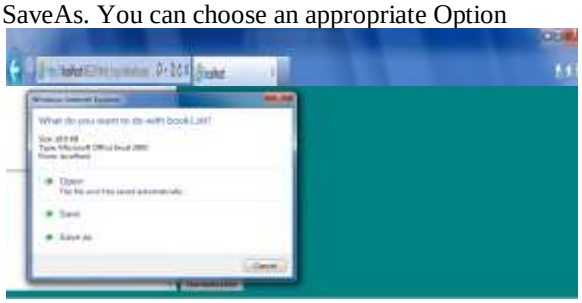


Figure 10: Message Box

When we click on Open option It will automatically open the Excel sheet which shows the graphical representation of XML Document .Here we are using Independent and dependant approach. when the values of XML Document are not present in the database it is considered as an Independent Data and it is stored in database, in the Excel we are representing independent data on the left side and dependant data on the right side when XML values are present in the database it is considered as Dependant Data and it shows on Right side Below diagram shows independent data

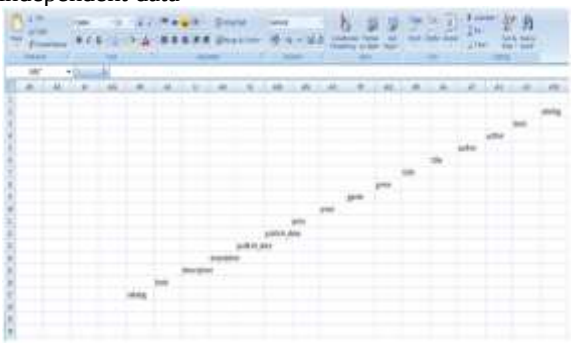


Figure11 : Independent data

This Normalization button also contains problem definition about transitive dependency which is already represented in analysis of Problem we can easily remove it with Normalization below Diagram shows perfect representation of XML document in Graphical Form.

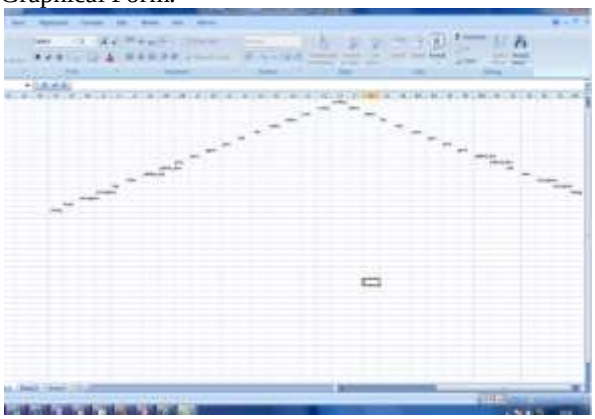


Figure 12: Graphical form

In this way we have tried to give some different representation to XML document which makes it more impressive so that everyone wants to use it according to their need and requirement of user application. After this we have found redundancies which cause data dependencies in XML document, then we have applied Different levels of normalization to create redundant free XML document. Here I have used transitive dependency to represent it into the XML document and apply third normal form to remove it, here I have used Excel sheet to represent XML document in the form of diagram.

Conclusion

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

We have presented GN-DTD to describe XML documents based on graphical form of document. Using GN-DTD, the syntax and semantic constraints of XML documents can be described clearly and precisely, with help of this we can easily find out the redundancy which causes the data dependency. To show the flexibility of the GN- DTD, we will define the different levels of normalization according to the type of dependency, so that we can easily remove it also improves the execution time of XML document which automatically improves the overall performance of XML document

References

1. Arenas, M. and Libkin, L. A Normal Form for XML Documents, *ACM Transaction on Database System*, vol 29(1), 2004, pp. 195-232.
2. Bex, G. J., Neven, F., and Bussche, J.V., DTD versus XML Schema.A Practical Study. *In Proceeding of the Seventh International Workshop on the Web and Databases*, 2004,pp. 79-84
3. Dobbie, G., Xiaoying,W., Ling. T.W., and Lee, M.L. ORA-SS: An Object-Relationship-Attribute Model for Semi-Structured Data. *Technical Report, Department of Computer Science, National University of Singapore*, 2000.
4. Embley, D. and Mok, W.Y., Developing XML Documents with guaranteed "good" properties, *In Preceding of the 20th International Conference on Conceptual Modeling*, 2001, pp. 426-441.
5. Feng, L., Chang, E., and Dillon, T., A Semantic Network-Based Design Methodology for XML Documents, *ACM Transactions on Information Systems*, Vol 20, Number 4, 2002, pp. 390-421.
6. Goldman, R., and Widom, J., Data guides: Enabling query formulation and optimization in semi structured database, *In Proceeding of the 23rd International Conference on Very Large Databases(Athens)*, 1997, 436-445.
7. Hegaret, P.L, The W3C Document Object Model (DOM), available At *Http://www.w3.org/2002/07/26/dom-article*,2002.[accessed January 10, 2009]
8. Kolahi, S., Dependency-preserving normalization of relational and XML data, *Journal of Computer and system Sciences*,2007, pp. 636-647.
9. Lee, S.Y., Lee, M.L., Ling, T.W., and Kalinichenko, L.A., Designing Good Semi-structured Databases, 1999, pp 131-135.
10. Lv, T., Gu, N.,Yan, P., Normal forms for XML documents, *Information and Software Technology*, 2004, pp. 839-846.
11. Mani, M., Lee, D., and Muntaz, R.R., Semantic Data Modeling Using XML Schemas, *In Proceeding of 20th International Conference on Conceptual Modeling*. 2001
12. Papakonstantinou, Y., Garcia-Molina, H. and Widom, J. Object exchange across heterogeneous information sources. *Proceedings of the Eleventh International Conference on Data Engineering*: 1995, pp. 251–260.
13. Powell, G., *Beginning XML Database*,. USA: Wiley, 2007
14. *W3C XML Specification DTD, available at Http://www.w3.org/XML/1998/06/xmlspec-report-19919010.htm*[accessed December 29, 2009]
15. Wang, J. and Topor, R., Removing XML Data Redundancies Using Functional and Equality-Generating Dependencies, *16th Australasian Database Conference*, 2005, pp. 65-74.
16. Yu, C. and Jagadish, J.H., XML schema refinement through redundancy detection and normalization, *The VLDB Journa*l, 2008, pp. 203-223.
17. Zainol, Z. and Wang, B., GN-DTD: Graphical Notation for Describing XML Documents, *In Preceding of 2nd International Conference on Advances in Databases, Knowledge, and Data Applications, DBKDA*, IEEE Computer Society, 2010, pp. 214-221.

Author biblography

Ms.Jagruti S. Wankhade
M.E.(I.T.)
Jawaharlal Darda Institute of Engg and Technology

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Mr. Onkar V. Chandure
M.E.(I.T.)
Jawaharlal Darda Institute of Engg and Technology