ISSN No.-2321-8711

International Journal of Software Engineering and Systems

Volume No. 12

Issue No. 3

September - December 2024

ENRICHED PUBLICATIONS PVT.LTD

JE - 18,Gupta Colony, Khirki Extn,
Malviya Nagar, New Delhi - 110017.
E- Mail: info@enrichedpublication.com
Phone :- +91-8877340707

ISSN No.-2321-8711

International Journal of Software Engineering and Systems

Aims and Scope

Software Engineering has become very important with the ever-increasing demands of the software development to serve the millions of applications across various disciplines. For large software projects, innovative software development approaches are vital importance. In order to gain higher software standards and efficiency, software process adaptation must be derived from social behavior, planning, strategy, intelligent computing, etc., based on various factors. International journals of software engineering address the state of the art of all aspects of software engineering, highlighting the all tools and techniques for the software development process. The journals aims to facilitate and support research related to software engineering technology and the applications. International journals of software engineering welcomes the original research paper, review papers, experimental investigation , surveys and notes in all areas relating to software engineering and its applications. The following list of sample-topics its by no mean to be understood as restricting contributions to the topics mentioned:

Ø Aspect-oriented software development for secure software

Ø Dependable systems

Ø Experience related to secure software system

Ø Global security system

Ø Maintenance and evolution of security properties

Ø Metrics and measurement of security properties

Ø Process of building secure software

ISSN No.-2321-8711

ISSN No.-2321-8711

International Journal of Software Engineering and Systems

(Volume No. 12, Issue No. 3, September - December 2024)

Contents

ISSN No.-2321-8711

Data Format Conversions using Single SOA at 10 Gb/s

Jatinder Singh
Department of Electronics and Communication Engineering,
Sant Longowal Institute of Engineering and Technology, Longowal, Sangrur, Punjab.

ABSTRACT

This paper described the simulation demonstration of an all optical data format conversion scheme between nonreturn to zero (NRZ) to return to zero (RZ) that employs a Mach-Zehnder (MZ) Interferometer For the first time, I have proposed MZ-Interferometer using a single SOA for data format conversion from NRZ to RZ.

Keywords: Data Formats, Mach-Zehnder Interferometer (MZI), NRZ, RZ.

I. INTRODUCTION

Future all optical networks are likely to employ both wavelength-division multiplexing (WDM) and optical-time-division multiplexing (OTDM) techniques. Optical TDM (OTDM), a scheme that can increase the bit rate of a single optical carrier to values above 1Tb/s. A key feature of WDM is that the discrete wavelengths from an orthogonal set of carriers that can be separated, routed, and switched without interfering with each other. Transmultiplexing or interconversion [1] between wavelength-division multiplexing (WDM) and optical-time- division multiplexing (OTDM) may become an important operation for future optical networks [2]. All wavelength-division multiplexed (WDM) and optical-time-division multiplexed (OTDM) networks are required to support a variety of data formats. The extensively used two data formats in these networks are nonreturn-to-zero (NRZ) and return-to-zero (RZ). NRZ is preferred in WDM networks for its ease of implementation, relatively high spectral efficiency and timing-jitter tolerance. Although RZ format requires twice the NRZ transmission bandwidth, it is quite useful in applications including passive time-division multiplexing and demultiplexing due to its tolerance to fiber nonlinearities in spite of dispersion-induced effects. There will be a need for all-optical data format conversion between WDM and OTDM signals [3]. It has been stated that fully functional WDM networks should have the capability of all-optical format conversion between RZ and NRZ format [4]. Therefore data format conversions are likely to be used for all future al optical networks in order to add the flexibility to the optical networks. Especially, all the optical data format converters between NRZ and RZ is an essential function in interfacing metro/access and optical core networks [5]. Figure 1 shows an optical network that includes both metro network and core network. Access/Metro networks transport data to (and from) individual users. An access network is that part of a communications system which connects subscribers to their immediate service provider. In the

access/metro networks the transmission distance is only from few meters to few hundred kilometers. Therefore the preferred data format for the access/metro networks is low cost NRZ. Whereas a core network (or network core) is the central part of a telecom network that provides various services to customers who are connected by the access network as shown in figure 1. Therefore in the core network the data rate is very much high so we can transmit a very high speed OTDM signals such as 40Gb/s to 120Gb/s. So the preferred data format is RZ. From the access nodes to the all-optical transport layers, the OTDM system with the RZ data format has been suggested to increase the total transmission capacity over 40 Gb/s by using the bit-interleaving technique. For this high-speed OTDM transmission, the RZ format is preferred due to its robustness to the nonlinear effect in spite of the dispersion-induced effect.

Therefore, format conversion between NRZ and RZ data formats is an essential function in linking and interfacing the ultrafast OTDM networks and the low-speed access networks. Therefore we required a data format conversion that converts the NRZ signals in access/metro networks to RZ signals in core networks and vice-versa. Up to now, no data format converter between NRZ and RZ has been demonstrated using single SOA. In this paper I experimentally demonstrated 10Gb/s data format conversion between NRZ and RZ using physical Mach-Zehnder Interferometer with various delays in

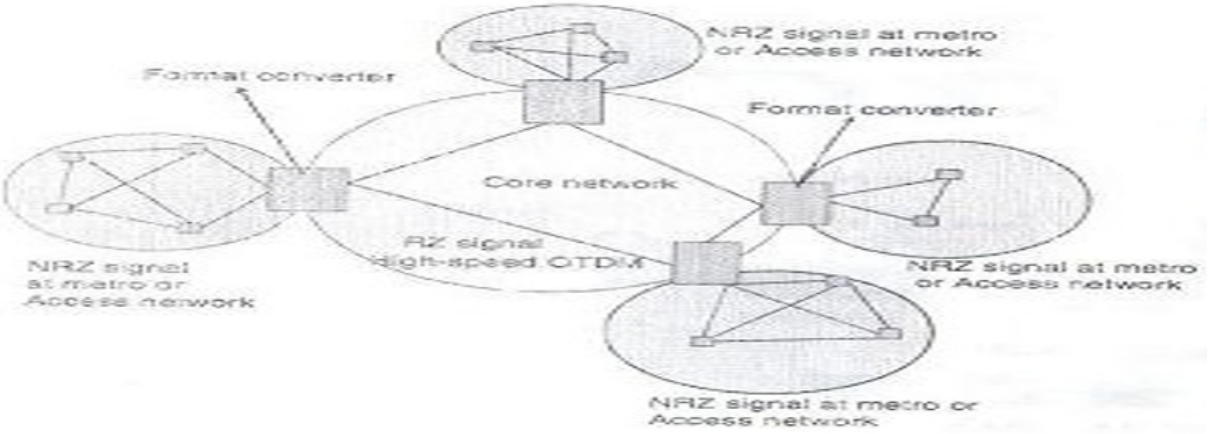


Figure 1: Data Format Converters at the interface between access/metro network and core network

II. DATA FORMAT CONVERSION USING MZ-INTERFEROMETER

Figure 2 shows the principle of data format conversion from NRZ-RZ based on Mach- Zehnder (MZ) Interferometer. We can see that consecutive ones in the symbol sequence alternate between two phase levels to provide the converted RZ signal. The input signal directly goes to the output coupler without any delay whereas the signal in the lower arm is delayed and combined with the un-delayed signal to provide the RZ signal. In principle, any duty cycle of RZ signal can be generated by choosing the proper delay between two arms using Mach-Zehnder (MZ) Interferometer, however the pulse width or duty cycle is limited by the rise and fall time of the data [6].

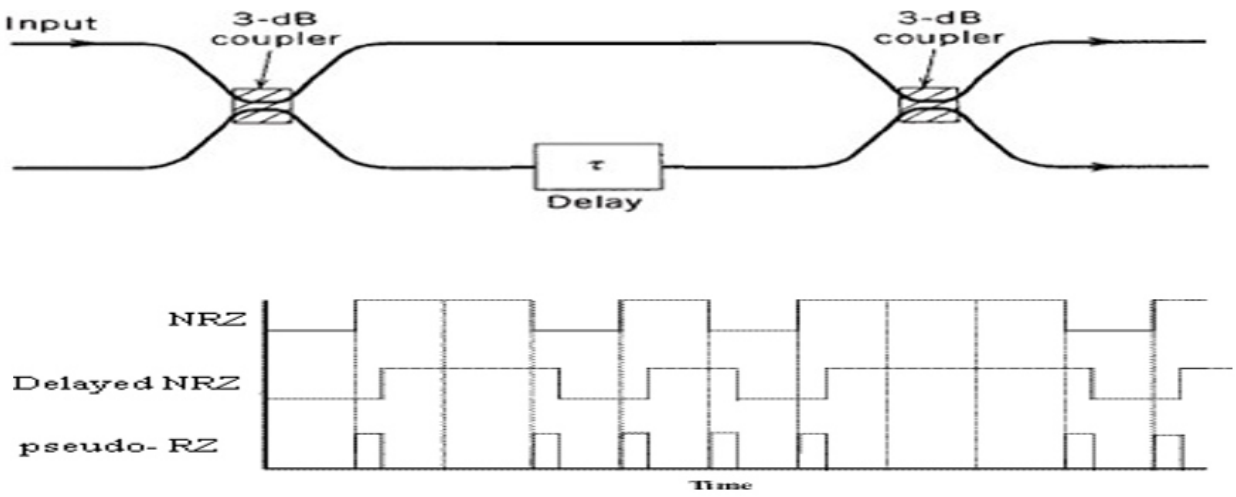


Figure 2: The Principle of Data Format Conversion Using MZ-Interferometer

Figure 3 shows the experimental set up for the data format conversion from NRZ-RZ. A continuous wave (CW) laser wavelength is 1558.2nm. A binary 10Gb/s NRZ electrical source is placed at the input. The generated 10Gb/s NRZ is modulated using MZ modulator before it is applied to the Mach-Zehnder (MZ) Interferometer. The input power set in the CW laser is -3dBm. Then I use a MZ interferometer to realize the data format conversion. The time delays set between the two arms in the MZ Interferometer are 3,5,10,15,20,25,30,35,40,45,50,55 and 60psec.

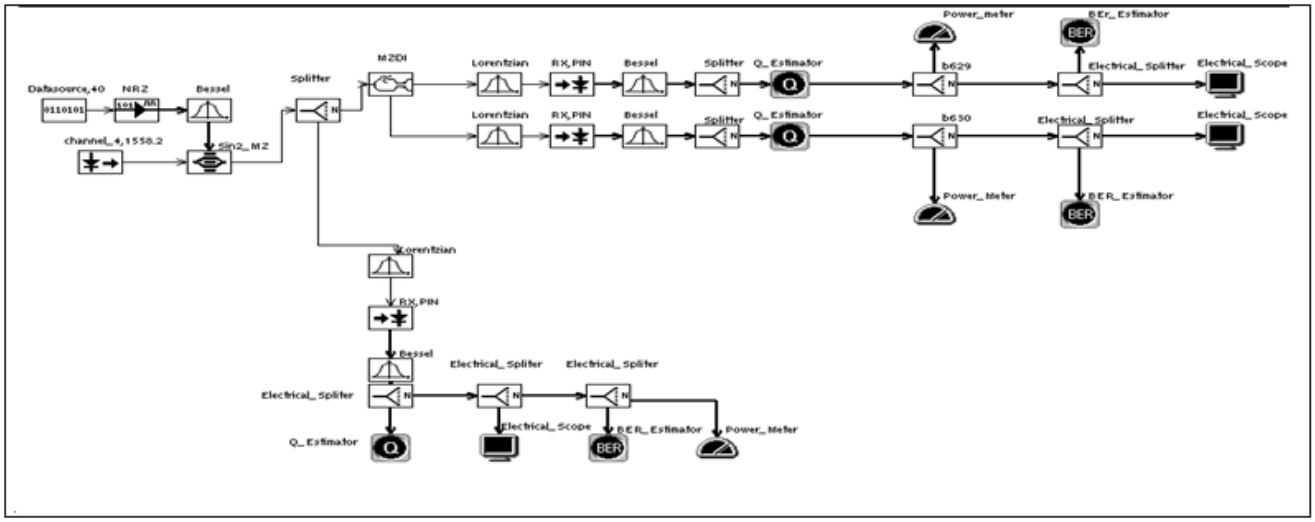


Figure 3: Experimental Setup

Figure 4 the eye diagrams of input and output signals at various time delays between two arms of the MZ-Interferometer are shown in figure 4. The eye diagrams are observed with 10 Gb/s oscilloscope. As shown in figure 4, the smaller the time delay between the two arms of the MZ-Interferometer, the smaller the duty cycle and the wider spectrum. Because for the 10 Gb/s the rise time and the fall time of the signal are 50 psec, the shortest pulse width of the optical signal is approximately 50 psec. Therefore, as the time delay between the two arms is greater than 50 psec the duty cycle of the converted RZ signal is improved [6].

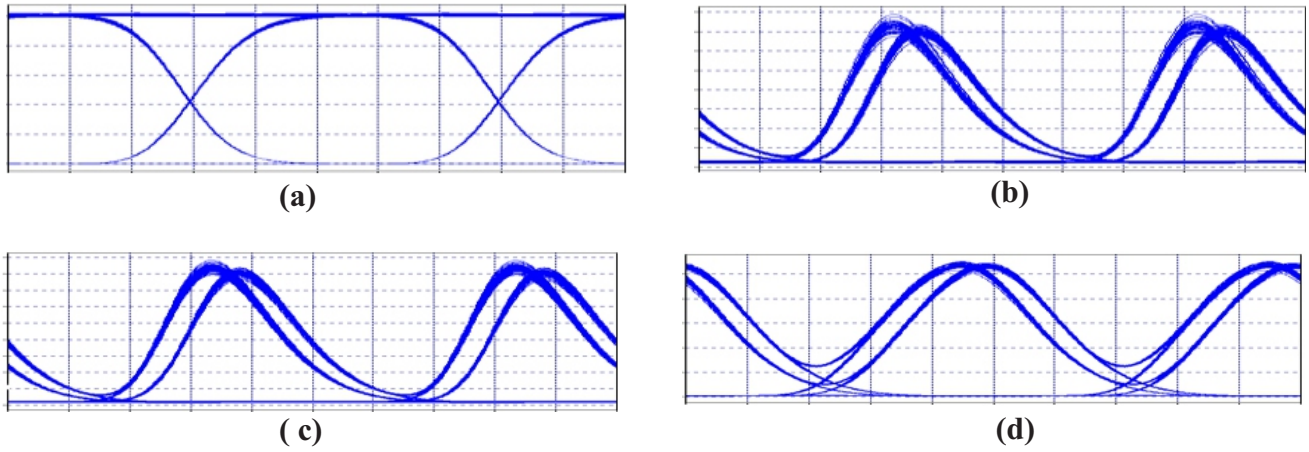


Figure 4: Measured Eye Diagram at Input and Output Signals at 10Gb/s (a) Input Signal (b) 20psec (c) 25psec (d) 60psec

III. DATA FORMAT CONVERSION USING MZ-INTERFEROMETER

In this section, ideal MZ-Interferometer is designed by using a single SOA at 10 Gb/s for NRZ to RZ data format conversion. The MZ-Interferometer is an ideal component which provides the delayed in the output signals according to the time delay in the lower arm of the interferometer as described in previous sections. In this section, the MZ-Interferometer isphysically implemented using a single SOA to provide the same outputs as obtained with MZ-Interferometer.

The proposed model of this converter is shown in figure 5. For the conversion from NRZ-RZ, the local RZ pulse train is applied to the input port whereas the NRZ data signal which is to be converted is applied to the control port as shown in figure 5 [7]. The parameter of the SOA is so adjusted so that it would be able to generate some delay (phase shift) in the lower arm of the interferometer.

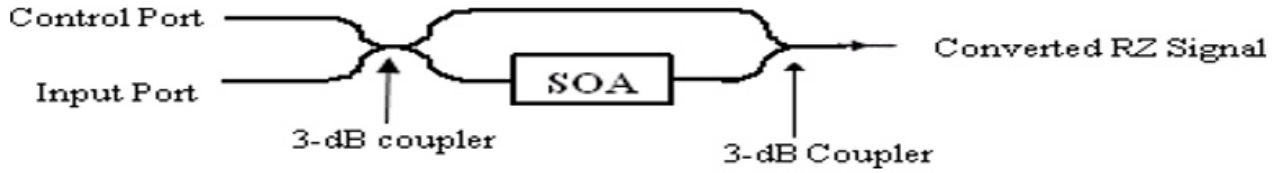


Figure 5: Single SOA Based MZ-Interferometer for NRZ to RZ Conversion

The simulation setup for single SOA based NRZ –RZ data format converter at 10 Gb/s is shown in the figure 6. A continuous wave (CW) Lorentzian Laser wavelength is 1555 nm. Which has a FWHM (Full Width Half Maximum) of 1 MHz and power of -3 dBm. A binary 10 Gb/s NRZ (Non-Return-to Zero) electrical signal is generated by data source at the input with pseudo-random sequence degree of 10. The generated electrical signal is then applied to the Bessel low pass filter (Number of poles are 2 and -3dB bandwidth of 20 GHZ).The filtered electrical pulses from the data source are modulated with the laser using Mach-Zehnder (MZ) Modulator which has Sin2 electrical shaped Input-Output P-V characteristic (-3dB bandwidth of 20 GHZ). The clock RZ pulses are generated by the pulse generator. The wavelength of the RZ clock is 1558 nm. The RZ clock is raised cosine with round off of 0.5. The RZ clock and NRZ input signal are applied to the 3-dB coupler where at the output of the coupler, the signal in the upper arm is directly applied to the output coupler. Whereas signal in the lower arm is pass through

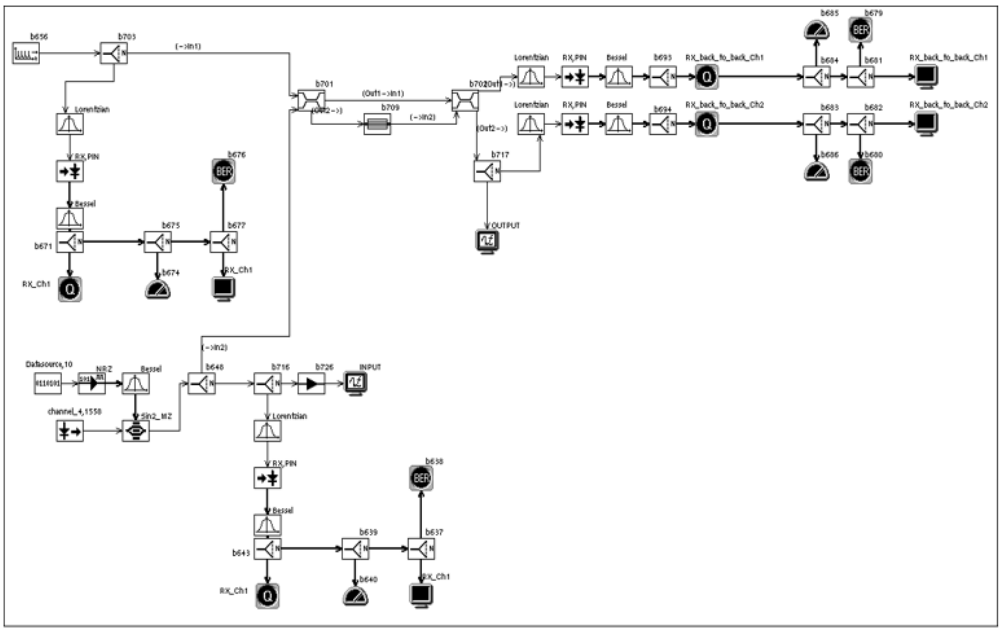
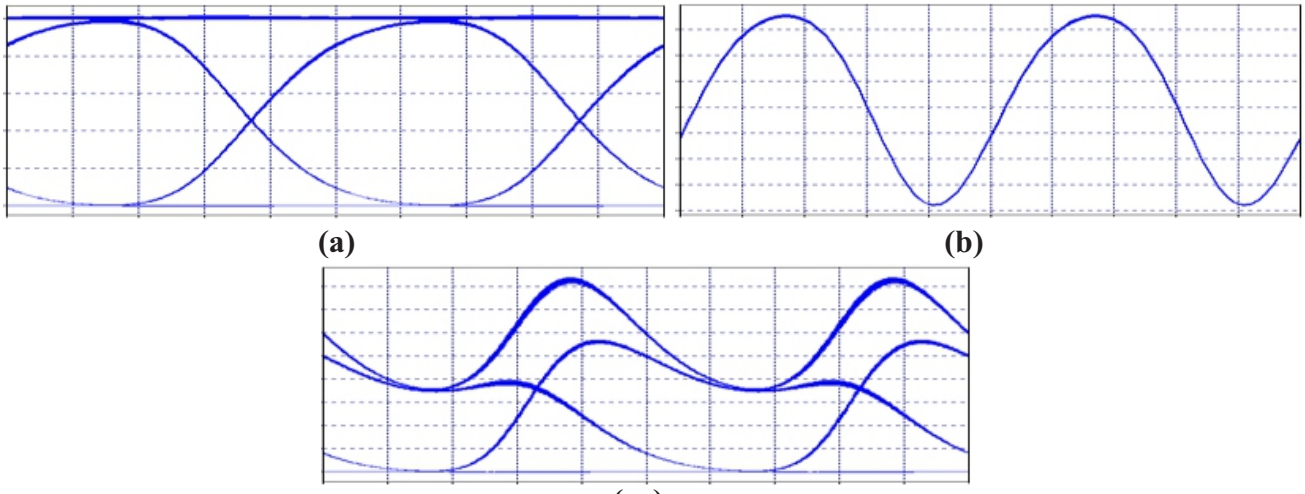


Figure 6: Simulation Setup for Single SOA based Data Format Converter at 10 Gb/s

The eye diagrams of input signal NRZ, clock signal RZ and output signal are shown in figure 7, when the wavelength of the input NRZ signal is 1555 nm. As shown in figure 7, the NRZ signal is converted in the RZ signal.



(a) **(b)**

(c)

Figure 7: Eye diagrams at 10 Gb/s (a) input signal NRZ and (b) clock signal (c) converted output RZ signal

V. CONCLUSION

I have proposed a 10Gb/s data format conversion between NRZ and RZ signals and successfully demonstrated for the first time using single SOA as MZ-Interferometer. The converted RZ signal using physical MZ-Interferometer with single SOA has improved quality of the converted signal as compared to the ideal MZ-Interferometer. Also, due to the SOA as delay element the signal after passing through the SOA is amplified. Hence the received power is more than the ideal MZ-Interferometer. Therefore, the designed physical interferometer using single SOA based NRZ-RZ data format converter at 10 Gb/s has provided the best results as compared to the MZ-Interferometer at 10 Gb/s.

V. REFERENCES

[1] *J. P. R. Lacey, M. V. Chan, R. S. Tucker, A. J. Lowery, and M. A. Summerfield, "All- optical WDM to TDM transmultiplexer," Electron. Lett., vol. 30, pp. 1612–1613, 1994.*

[2] *Sang-Gyu Park, L. H. Spiekman, M. Eiselt, andJ. M. Wiesenfeld," Chirp Consequences of All-Optical RZ to NRZ Conversion Using Cross-Phase Modulation in an Active Semiconductor Photonic Integrated Circuit," IEEE Photonics Lett., vol. 12, No. 3,pp. 233-235, 2000.*

[3] *D. Norte, E. Park, and A. E. Willner, "All-optical TDM-to-WDM data format conversion in a dynamically reconfigurable WDM network," IEEE Photon. Technol. Lett., vol. 7, no. 8, pp. 920–922, Aug. 1995.*

[4] *D. Norte and A. E. Willner, "Experimental demonstrations of all-optical conversion between RZ and NRZ data formats incorporating noninverting wavelength shifting leading to format transparency," IEEE Photon. Technol. Lett., vol. 8, no. 5, pp. 712– 714, May 1996.*

[5] *Jianjun Yu, Gee Kung Chang, John Barry, Yikai Su, " 40 Gb/s signal format from NRZ to RZ using a Mach-Zehnder delay interferometer," optics communication science direct, vol. 248 (2005), pp. 419-422.*

[6] *Jianjun Yu," Generation of Modified Duobinary RZ Signals by Using One Single Dual- Arm LiNbO3 Modulator," IEEE Photon Technology letters, vol. 15,no. 10, Oct. 2003.*

[7] *L. Xu, V. Baby, I. Glesk, and P. R. Prucnal, "All-optical data format conversion between RZ and NRZ based on aMach–Zehnder interferometric wavelength converter," IEEE Photon. Technol. Lett., vol. 15, no. 2, pp. 308–310, Feb. 2003.*

ISSN No.-2321-8711

Introduction To Semantic Web and its Future

Hedayat Bahadori, Sara Bahadori, Amin Samimi Behbahan, Milad Samimi Behbahan, Shabnam Azari

Department of Computer Engineering,
Omidiyeh Branch, Islamic Azad University, Omidiyeh, Iran

ABSTRACT

Semantic web is considered as the most important research subject in web realm which allocates many researches in recent years. One of the main implementation for accomplishing web is to promote the knowledge of present web and add semantic metadata to web pages or on the other hand, is semantic annotation. Regarding to the high value of web pages and remarkable subject variability of its present contents, it can be realized that annotation of web page based on present ontology is one of the most important challenging problems of confronting semantic web. Whereas annotating semantic web and developing ontology necessitate information extraction and discovering knowledge, it can be realized that various methods of data mining such as assortment, clustering, mining association rules, and text mining play effective roles in removing foresaid challenge.

1. INTRODUCTION

Digital world grew astonishingly during 50 years after naming AI in Dortmund Conference and produce information substrate that it was less than predicted. Today, World Wide Web links more than tens billion pages. Search engine can deduce subjects which include these links.

Inventor of World Wide Web presented a kind of semantic web as an article with James Hendler and Ora Lassila [1]. Semantic web creates structures for web pages that people can do complicated responsibilities by computer. The goal of research about semantic web is to pass the documents and to design for realizing and to process by machine.

Semantic web tries to develop documents of web as meaningful interchangeable.

Semantic web has broad applications in various fields such as management of knowledge, web-based training, bioinformatics, E-government, digital libraries and etc. to ensure identical realizing of machines of metadata labels and communication and collaboration between them, a subscription repository is necessary for description of labels. Ontology is used as such repository of meanings in semantic web. The term of ontology attributes to show the knowledge from special amplitude in realm of computer science. The set of objects and relationships between them are defined by dictionary in ontology. Ontology is a tool for describing added meanings to web documents and provides the possibility of using these meanings for intelligent software agents and web users [2].

2. SEMANTIC WEB

Future web is realizable and processed by computer in addition to human from point of view of Brenner-lee known as father of web. The main goal of semantic web is a web that is able to converse to machine [1]. Semantic web is substrate for many various applications which can be developed. Semantic web contains of operational information, the Information which should be obtained by semantic theory for interpretation of symbols. Semantic theory develops logic connections of terms with capability of collaborations between systems. Necessary key for semantic web is capability of collaboration. Among definitions of semantic web, we can mention following options [3]:

- Semantic web is a project with the goal of creating world media for exchanging information to be realizable for computer
- Semantic web is network of information on a global scale how it is easily to process by machines
- Web semantic includes intelligent data of web that is processed by machine

The future of web can be divided two sections from point of Breners' view. The first section is to manufacture web with environment with collaboration and communication of users. Second section is to manufacture web that is realized and processed by the machine. Figure 1 shows the point of Breners' view about presenting web [4]. The language HTML that is used in present web is not able to state objects and their relationships. Another language has been created called RDF to use in semantic web.

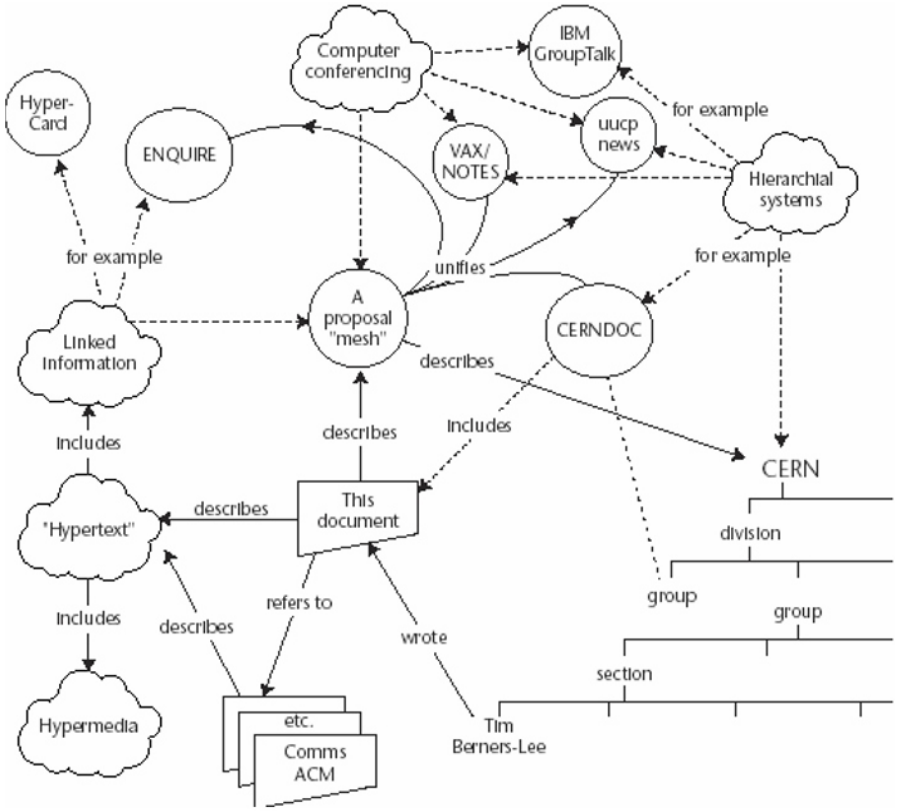


Figure 1: suggestion of Mr. Tim Berners Lee about web

The first step is to change the thought about data. Data should be more intelligent in semantic web for being realized by machines. More concepts should be saved with data how the machines can realize them [3]. Various levels of intelligent data have been showed in figure 2.

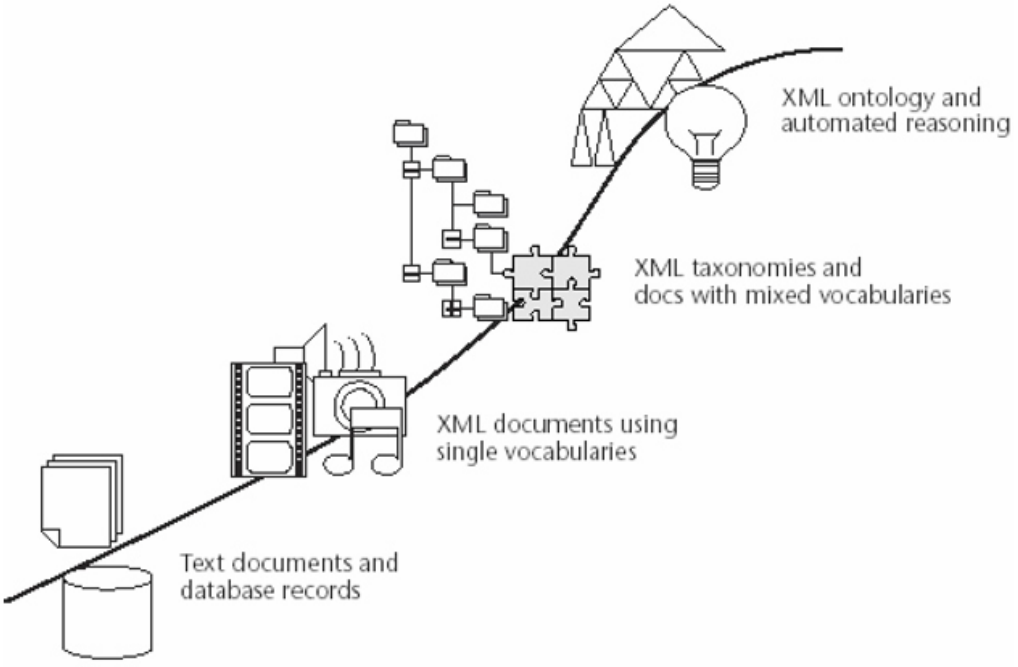


Figure 2: Various Levels of Intelligent Data

Different architectural layers of semantic web are shown in figure 3 RDF, basic language used in semantic web is established based on XML [1]. XML has been established based on Unicode and URL and backs up different languages. URL is used for determining concepts in semantic web. For instance, URL is used as a kind of URL for determining reservoirs in web. The main part of semantic web is ontology. Ontology communicates between labels of documents of semantic web and real objects. By using rules we can obtain new knowledge from present knowledge. Manufacturing standard framework for present rules, we can achieve proof. Obtained proofs are communally used in different applications. So, trust web is created as final level of semantic web.

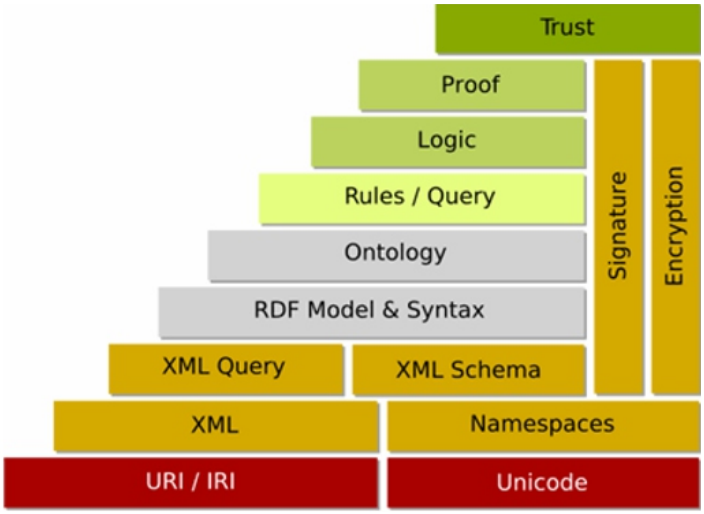


Figure 3: Architecture of Semantic Web

3. SEMANTIC RELATIONS

Ontology and semantic webs create semantic relations between demonstrated existences. Semantic relations are meaningful dependencies between two or more concepts, existences, or set of existences. Display systems of type information are divided into three general sections of terms list, classification and categories and relations between these lists [Zhang, 2007. Terms list is list of terms are usually shown by their definitions. Classifications and categories create sets of objects. Relationship lists specify the connections between terms and concepts. This list is summarized systems in following classification:

Term Lists
Authority Files Glossaries Dictionaries Gazetteers
Classifications and Categories Subject Headings Classification Schemes Taxonomies Categorization Schemes
Relationship Lists
Thesauri
Semantic Networks
Ontologies

Terms are listed with specific meaning (term list) like in alphabetical order in the first category. Terms are easily processed in necessary time. Dependency of these terms is separated from their alphabetical order. The meaning of the term is not related to previous term. Second section (classification & categories) terms and concepts are arranged as hierarchical manner. Hierarchy has been determined as specific type of relation between terms and concepts. Present terms are placed in higher level, higher classification or wider categories. These terms usually have wider meaning than present terms in lower levels.

Hierarchy lists show the relation of class-subclass as dependent to meaning between terms and concepts. In third section (relation lists), relations are shown between terms and concepts. Semantic relationship is more created and stated in relation lists. Terms and concepts are linked together as meaningful, for example, hierarchy, horizontal order (synonym), reverse order (antonym) or causal order [5]. Showed different Realization of semantic relation creates relation lists in term lists. Relation lists provide useful framework for understanding difference and similarity between ontology and the other forms of knowledge representation. Ontology depends to one or more traditional library display like reservoir, classification, category pattern, controlled dictionary and even a dictionary. Dependency of traditional library display models and ontology are shown by arranging them based on following category [5]:

Term Lists

Controlled vocabulary

Dictionary

Hierarchical lists

Classification Scheme

Taxonomy

Relationship Lists

Thesaurus

Ontology

4. TYPES OF MEANINGS

Summary of three meanings have been presented in this section [5]:

- **Implication:** this type of meaning is implicit of data patterns and it is not displayed explicitly in manner of being processed by machine. Tools and applications were developed for implications and it is easily available. Learning techniques of machine like clustering, hidden markov model, artificial neural network and so on apply implication.
- **Formal Meaning:** human beings often communicate by language. Natural language is based on syntax rules and it is ambiguous in concept. Computers are lack of removing ambiguities and perception of complicated natural language. Natural language is not used as a tool for communicating among machines. Concepts or facts must be stated by being processed by computers. Formal meanings are displayed the good way form.
- **Description Logic:** description logic is highlighted form of knowledge presentation. Formal meanings in DLs are based on theory of sets. Relations between concepts or roles are stated as multiplied vector from amplitude.

5. THE ABILITY OF PROCESS OF MACHINE

To develop a web with meanings, reservoirs on web must be displayed with realizable descriptions for machine. Descriptions are as standard concepts and relations among concepts. Annotating web contents must be done using dictionaries and structures which have been defined with a domain of ontology explicitly and officially. The ability of process of machine severely depends to accessibility and increasing ontology. Creating ontology is manually tedious task, time-consuming and prone to error. Manufacturing automatic tool for creating ontology of available information extraction is primary step of implementation. Semi-automatic ontology extraction method is considered as applied solution. Some steps are necessary for re-application of ontology online in creating new ontology. These steps include ontology search, ranking, segmentation, mapping and merging, annotating and assessing [5].

6. CHALLENGES OF INFORMATION SYSTEMS ABOUT SEMANTIC WEB

Superior semantic web is an approach of collecting simple words in search engines and is related to meaning of text. Subscription information, information extraction, maintenance and coincidence are easily possible in search based on key word [6].

Following cases have been stated as challenges related to semantic web in information systems in [6]:

- Challenge of developing ontologies domain
- Challenge of mapping, installing and merging ontology
- Challenge of interpretation management
- Challenge of retrieving information based on ontology

6.1. Challenge of Developing Ontologies Domain

Ontologies play important roles in semantic web. To develop ontologies, experts are appointed in the same domain. Each of experts may have different opinions (a social challenge). A development environment of collective ontology is necessary (a technical challenge).

A collective development environment must be able to control model, proposal, releasing and supporting coordination and collaboration. Developing such environment is a challenge. Today, most of the tools of ontology development like individual ontology editors are lack of this usefulness [6]

6.2. Challenge of Mapping, Installing and Merging Ontology

Using a unique ontology is so difficult for any data reservoirs. Users should apply ontologies related to themselves for interpretation of data reservoirs. Then if they intend, they create added mapping for designing them as standard domain ontology (central). Standard domain ontology is useful to back up the ability of collaborating data and inquiring. This mapping is a challenge among ontologies due to the possibility of dissimilarity in content, schema and meaning among ontologies. The process of mapping may not only include adjustment of ontology for create coherent ontologies. Ontologies are merged with central ontology for adding new words [6].

6.3. Challenge of Interpretation Management (annotation)

First challenge is to allocate interpretation tool for every domain based on presented ontology and requirement of users. Second challenge of world development is interpretation tool. Allocated interpretation tool for several domains needs an open architecture and backup operational interpretation for different domains. The third challenge is the capability of shared interpretation. Shared interpretation needs backup concurrency control. When the users interpret separated data sections, shared data interpretation is effective. Suppose that a person interprets the announcer's movements and the other specifications of discourse linguistic. Two ontologies can be used for this purpose. Each person may be familiar with one of them. One of projects of guiding shared interpretation is Annotea [6].

6.4. Challenge of Retrieving Information Based on Ontology

Interpreted data that can be searched by its interpretation is useful. One of affirmations of semantic web is high precision. Search engines should apply semantic reasoning and available ontologies in order to obtain accurate results and determine semantic relations between them. New opportunities need new approaches for refining search and strategies of connector of users. Main challenge of search is through data collection. These data collections were interpreted by various ontologies. Several ontologies can be existed. Ever domain can be schemed by several ontologies. A domain may need application of several ontologies. Therefore, not only mapping of ontology, but also mapping of user search is necessary [6].

7. ONTOLOGY

Ontology was created by two words onto means existence and Logia means study. Briefly, ontology defines as following: ontology is specification and definition of conceptualization [3].

Nowadays, ontologies include many similarities of structures regardless the language in that is schemed. Most of ontologies describe individuals (instances), classes (concepts), properties and relations. Ontologies are basic element in creating semantic web. If the meaning of data is displayed explicitly on web, interaction of meaning level is possible among users of web. Data of web should be realizable regarding domain and content for machine [7].

7.1. Languages Of Ontology

There are two types of languages for creating ontology. Languages perform based on graph. Like Topic maps، UML،RDF Semantic networks. Languages which perform based on proposition logic. Like (RuleML, LP/Prolog) Rules، First Order Logic. Some of languages stating ontologies include [3]:

- RDF(S) (Resource Description Framework (Schema))
- OIL (Ontology Interchange Language)
- DAML+OIL (DARPA Agent Markup Language + OIL)
- OWL (Ontology Web Language)
- XOL (XML-based Ontology Exchange Language)
- SHOE (Simple HTML Ontology Extension)
- OML (Ontology Markup Language)

7.2. The Manner of Creating Ontology

Two manual and automatic methods are used for creating ontology. Automatic methods use different approaches for extracting ontology from available texts. Most of these methods use process techniques of natural language. Automatic method that can produce plausible ontology, has not introduced yet.

Therefore, manual or semi-automatic methods are used for creating ontology. There is not standard method for creating ontology. Different methods are used based on application of ontology. One of these methods is shown in figure 4 [3].

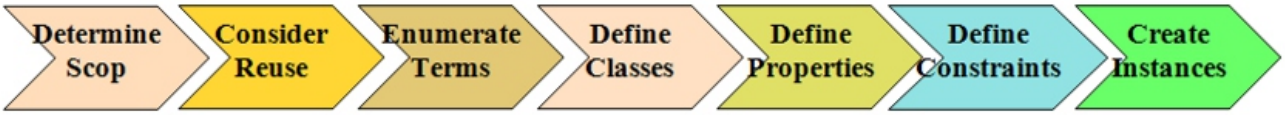


Figure 4: Method of Creating Ontology

The process of creating ontology is recursive. It may need to create amendment and recursive to previous step in each step. Steps of this method include [3]:

1- Realizing range: first study is done for creating ontology. Range that should be covered, type and manner of using ontology are determined.
2- Re-use: comprehensive study is done on available ontologies. It is investigated in this step whether available ontologies are useful for creating mentioned ontology.
3- Words diagnosis: mentioned words scope and their specifications are determined.
4- Diagnosis of classes and their structures: using obtained methods, their class and subclass are determined. For this reason, we can use bottom-up method or up- bottom or both of them.
5- Defining attributes: internal structure of classes is described. Usually, most of obtained words in step 3 that were not used in step 4 create mentioned attributes.
6- Defining ranges: ranges that every attributes can be formed are determined.
7- Creating objects: after creating ontology, mentioned instances can be created based on them.

7.3. Ontology Extraction

Ontology engineer has various responsibilities such as editing, assessing, writing, annotating, regulating, merging, re-application and extracting ontology. Responsibility of editing presents an editor for manually manufacturing ontology [8].

Extraction may includes techniques related to linguistic, statistical techniques, machine learning, and composite techniques affiliated to technology of retrieving information. Approach of machine learning refers to sets of techniques and algorithms for obtaining knowledge to an automatic method. Table 1 has briefly been stated different approaches of extracting ontology [8]:

Table 1: Different Approaches of Extracting Ontology

Name	Method	Language	Auxiliary source
Agirre et al.	Statistical	Unstructured	Ontology
Arasu and Garcia-Molina	Machine learning	Semi-structured	None
Buttler et al.	Machine learning	Semi-structured	None
Craven et al.	Joint method	Semi-structured	Both
Crescenzi et al.	Machine learning	Semi-structured	None
Davulcu et al.	Machine learning	Semi-structured	Samples
Faatz and Steinmetz	Statistical	Unstructured	Ontology
Faure and Poibeau	Linguistics	Unstructured	Both
Heyer et al.	Statistical	Unstructured	Samples
Jiang and Tan	Statistical	Unstructured	Both
Kietz et al.	Joint method	Unstructured	Both
Maddi et al.	Statistical	Unstructured	Samples
Maedche and Staab	Joint method	Unstructured	Both
Navigli and Velardi	Joint method	Unstructured	Both
Shamsfard and Abdollahzadeh	Linguistics	Unstructured	Both
Han and Elmasri	Machine learning	Semi-structured	Both
This study	Machine learning	Semi-structured	None

8. CONCLUSION

Semantic web and some of affiliated technologies have been described. The idea of semantic web is enrichment of web content with semantic metadata. Semantic web is developed copy of current web that cause to promote cooperation between human and computer by adding well-defined semantics.

REFERENCES

[1] Berners-Lee T., Hendler J., and Lassila O. "The Semantic Web", the Scientific American, 2001.

[2] Hassanzadeh H., Keyvanpour M.R., "Data mining methods for classification and evaluation methods for semantic annotation and ontology development", iran dataminig conference 4th, 2009.

[3] Al-ahmad A., "Introduction to the Semantic Web", TEHRAN University, faculty of electronic and computer Iran, 2006.

[4] Daconta M., Obrst L., Smith K., "The Semantic Web: A Guide to theFuture of XML, Web Services, and Knowledge Management.", (2006)

[5] Zhang J., "Ontology and the Semantic Web." Proceedings of the North American Symposium on Knowledge Organization., 2007, Vol. 1. Available: http://dlist.sir.arizona.edu/1897/

[6] Chebotko A., Lu S., Fotouhi F.,"Challenges for Information Systems towards the Semantic Web", Department of Computer Science Wayne State University, 2004

[7] Casevie D., Dguric D., devedzic V.," Model Driven Architecture and Ontology Development", chapter 3, Springer 2006

[8] Timon C., Du A. , Li B.F., King I.C., "Managing knowledge on the Web – Extracting ontology from HTML Web", Decision Support Systems 47 (2009) 319–331,Elsevier.

ISSN No.-2321-8711

Modeling of Nonfunctional Requirements for Agile Development Processes

Trupti Suryawanshi[1], Gauri Rao[2]
[1]MTECH, Computer Engineering, BVDUCOE, Pune, India.
[2]Associate Professor, Computer Engineering, BVDUCOE, Pune, India.

ABSTRACT

Agile method means deliver quickly, modify quickly and change frequently. Main purpose of designing agile software development process is to solve the difficulty of delivering high quality software on time and in under continuously changing requirements as per business need. Now days various agile software development process (Scrum, Extreme Programming (XP), Feature Driven Development (FDD) etc.), used to deliver quality functional requirements (FR). In software development process engineers are mainly focused only on functional requirements under the stress of deploying the software within a time. But Non Functional Requirement (NFR) is also important element of the development process. To develop quality software NFRs are equally responsible as FRs. NFR plays very sensitive role to deliver quality software products. Most requirement modeling methods are mainly concentrated on FRs where as NFRs received less attention. Discovering NFR is not easy job. There is no proper NFR elicitation method is available. In this paper we design a tool which is used to generate NFRs such as security, performance, scalability etc. This tool helps to the project manager and scrum team in risk evaluations, planning and visualization of the proposed plan.

Keywords: FR, NFR, Requirement elicitation, Scrum, Sprint

I. INTRODUCTION

Requirement Engineering is one of the most important areas of software engineering. Each software is successful if and only if it can fulfill all the requirements of customer. To develop any software first stage is requirement gathering and analyzing which plays important role. Requirements are classify into two types. First is the Functional Requirements and second one is Non Functional Requirements. Functional Requirements gives expected functioning of the system such as internal and external visible output of the system. On other side Non Functional Requirements gives detail constraints and control to the system [7]. NFRs are also known as quality attributes of the system.

Agile software development is becoming more and more popular in successful implementing and quickly delivering quality functional requirements. But academic and industrial researchers agree that NFRs have been ignored in software development process and mainly in agile methodologies [3]. FRs is easily collected from stakeholders but discovering NFRs is not an easy and straightforward job. There is no appropriate NFR generation method is available. Few methods have been proposed with some

advantages and its disadvantages. But still there is no any standard method defined by requirement engineering [13].

Now a day's most of the researchers are attracted towards the developing NFR generating tools. In this paper we propose a tool which easily focused on NFR elicitation. This tool helps agile development team modeling NFRs early on during requirement gathering as well as in analysis phases. This tool integrates FRs and NFRs modeling under one tool.

This paper presents a tool of modeling nonfunctional requirements for agile development process. Modeling of NFRs with FRs produces effective software. The paper organized as follows. Section II presents a background of our concept. Section III gives the overview and objectives of tool. Section IV describes brief analysis of the tool and how the system is designed with proper snapshots of our results. Section V discusses the conclusion and observation.

II. BACKGROUND

A. Agile Development Process:

Agile development software is allows to project team and to project developer for easily handle changes in requirements, at any stage of the schedule in development process [12].Agile software development methods are very popular due to its features. Quickly delivery, Simplicity, Face to face communication between stakeholders, Continuous awareness to technical superiority, constantly focus on excellent plan and his own selected team is some well-known features of agile development process [3].

Most common agile methods [6] are listed below:

- Extreme Programming (XP)
- Agile modelling (AM)
- Scrum
- Feature Driven Development (FDD)
- Dynamic Systems Development Method (DSDM)
- Adaptive Software Development (ASD)
- Crystal Methodologies (CM)

B. Non Functional Requirements:

NFR is defined as the software requirement which does not describe what actually software will do but how the software will do it [13]. NFR is the non behavioral requirement of the system. It defines quality and constraints of the system.

Examples of NFRs:

- Performance requirement is related with resource required, response time, throughput and anything else connected with performance.
- Security is related with protection of the system or software such as authentication, uthorization etc.
- Portability is nothing but the ability to run software on different platforms.
- Reliability is related with the ability of system to execute its proper functions and operations in its
- defined environment without experiencing failure or system crash.
- Usability means ease of use or user friendly software.
- Scalability requirement is the capability of software to change with the business environment.

III. PROPOSED WORK

Requirement Elicitation [14] is nothing but requirement gathering or discovering. Requirement elicitation is most important knowledge intensive activity in software development. This is one important job of requirement engineering. FR elicitation is easily handles by considering the requirements of stakeholders. But elicitation of NFR is not that much easy task. NFRs are prioritized in stockholders point of view so the elicitation process is closely related with stakeholder's consideration. NFR elicitation is crucial issue in software development process [7].There is no standard method is available for NFR elicitation. Recently several elicitation techniques are introduced with a few strengths and drawbacks. But yet not any method is agreed by requirement engineering team. There are standard definition defined for FR but no proper definition is available for NFR. To properly define NFR is complex because NFRs show inconsistency in different conditions. Sometimes integrating NFRs with FRs creates conflicts among stakeholders.

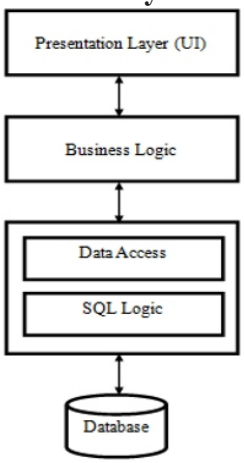
The proposed system has many features which help agile team in modeling NFRs. Rapid identification and linking with FRs will turn out effective software.

IV. ANALYSIS

A. Design

Our model is based on three tier architecture. Three tier architecture means separating project in three layers User Interface, Business Logic and Data Layer where we separate logic, data and user interface in three partition. It means if we replace SQL server database with oracle database then we need to change only data layer Or if we change user screen from windows application to website then we need to change only our user interface layer rest the whole things remain the same.

Fig.1 Three Tier System Architecture



Presentation Layer/ UI Layer: This layer of the architecture gives user interface where the user performs their activity. For example, login form, sign up or any other form fill by the users. Objective of this layer is the user's input validation and the rules for processing.

Business Logic: Most of the business operations are done here. For example, after collecting data, use our business laws and do the validation. Main purpose of this layer is defining classes and business components.

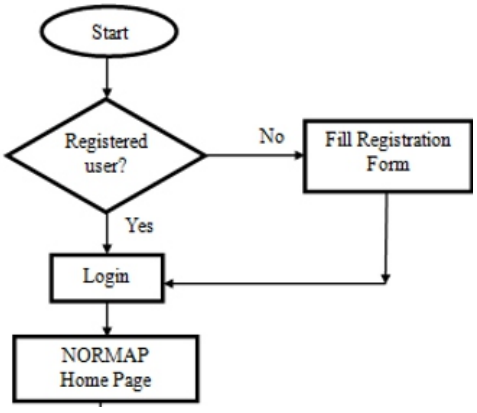
Data Access Layer: Each and every code which is connected with database and the things related with database done at the Data Access Layer. At this layer we define methods which gives help to the Business Layer for connection of the data and execute required actions, whether to return data or to control data (insert, update, delete and so on).

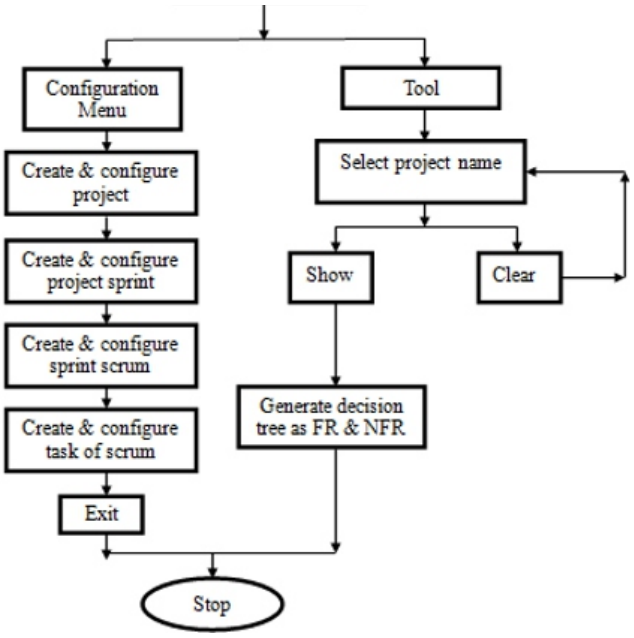
B. Features

1. Combine FRs with NFRs under single tool
2. Helps in risk evaluation process
3. Scheduling, designing and visualization of the future plan.

C. System Workflow

Fig.2 Flowchart of proposed system





D. System Description

The proposed approach gives two phases.

1. Configuration phase
2. Generate Decision Tree

1. Configuration phase

In this phase we provide different configurations. This configuration process is going through four different configuration phases.

a) Project configuration: At this stage we can add, delete or modify any information of project. This information includes project code, project name, description, time required to complete project, start date, end date, release date etc. This information is required for decision tree generation.

b) Task configuration: Here we can configure different task of project. It includes project name, type, task name, priority, code, duration, start and end date etc. This information is used in scrum and sprint configuration.

c) Sprint configuration: At this stage we can configure sprint details. Sprint details include same fields as in task configuration. Each sprint has duration like ½ month. The sprint is used for scrum creation.

d) Scrum configuration: Now here we can configure the scrum under sprint. Each scrum has duration like ½ weeks. The scrum is used for task tracking.

2. Generate Decision Tree:

For decision tree generation user needs to select the project. All sprint, scrum, task are categorized in two sections as FRs and NFRs.

Each node of decision tree holds below listed information:

- Project/sprint/scrum/task code
- Number of days complete
- Number of hours complete

Generated result helps to project manager for risk driven evaluations. It also useful for scrum and project manager to visualize and analyze proposed plan of project.

E. Algorithm

Input: Configuration details

Output: Decision tree of FR and NFR Algorithm:

Begin

Registered user will login.

Click on Configuration and select Project Configuration. Set project configuration details

Click on Configuration and select Task Configuration Set task configuration details

Click on Configuration and select Sprint Configuration Set sprint configuration details

Click on Configuration and select Scrum Configuration Set scrum configuration details

Click on tools and select generate tree Select Project Name

Click on Show button

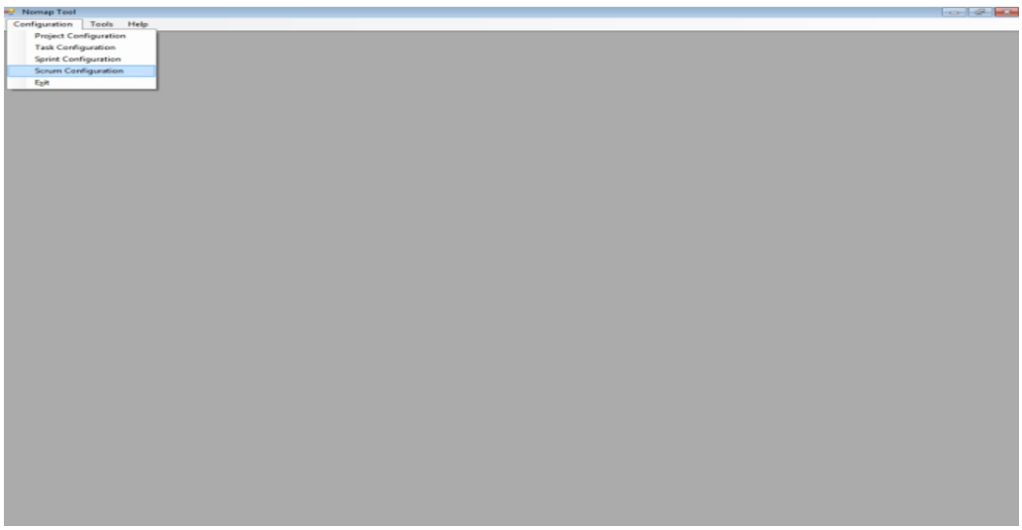
Display project, sprint, scrum and task categorized into FR and NFR Select exit

End

F. Results

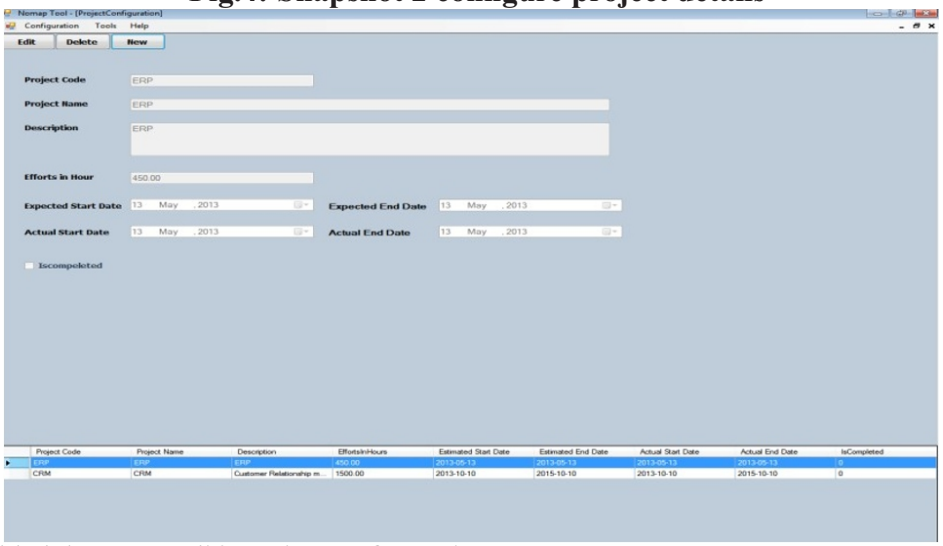
1. User will login with its username and password.
2. After successfully login user will get home page of application.

Fig. 3: Snapshot 1 of Home Page



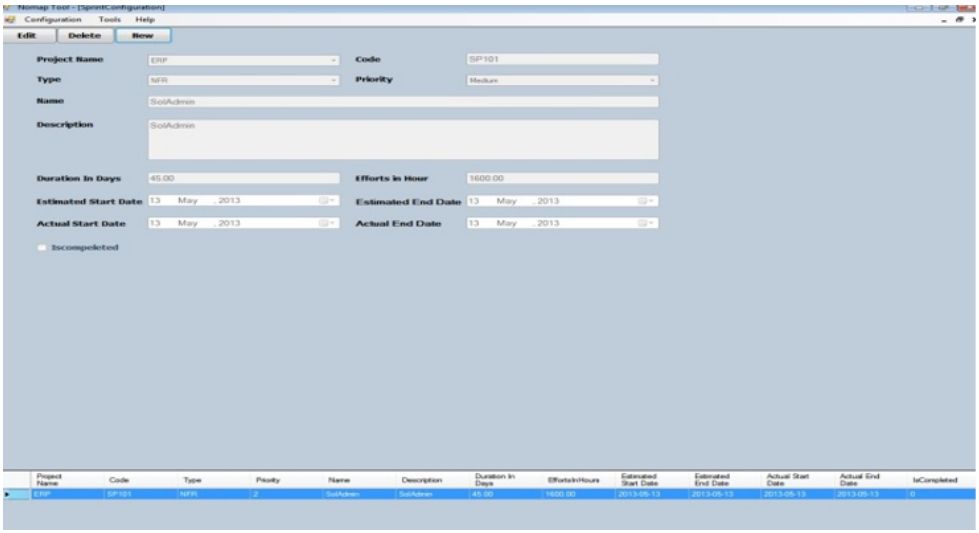
3. User can add , delete or modify the Project configuration

Fig.4: Snapshot 2 configure project details



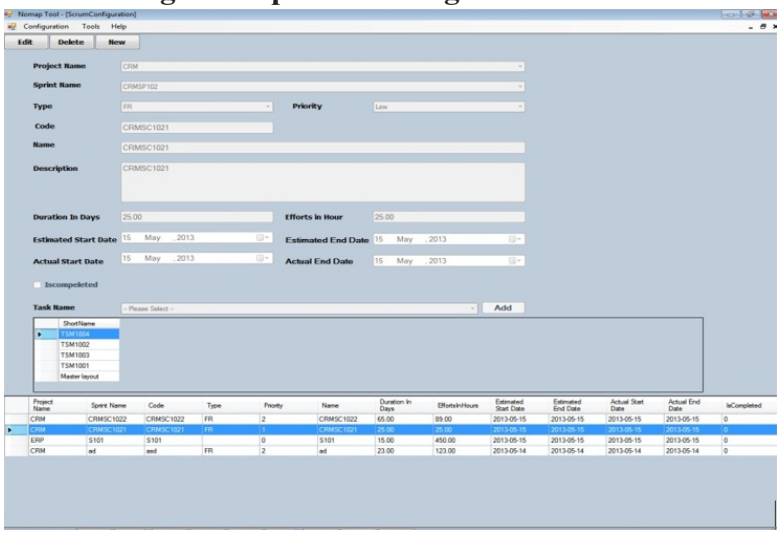
4. User can add, delete or modify sprint configuration

Fig.5: Snapshot 3 configure sprint details



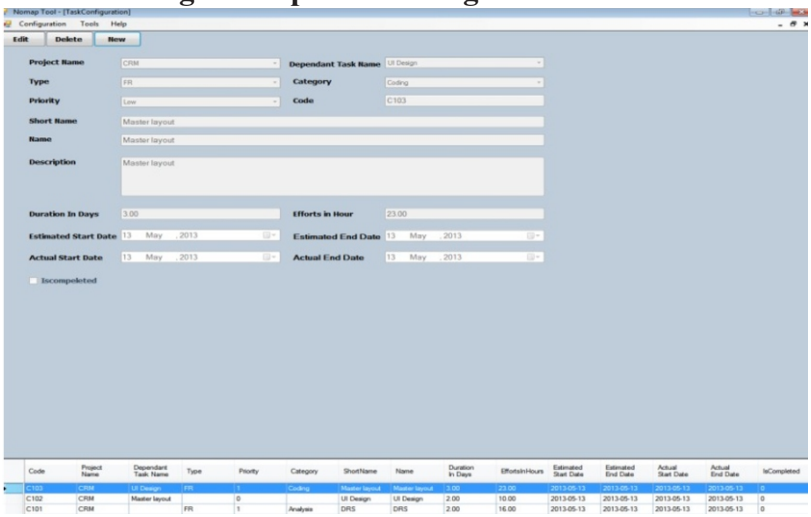
5.User can add, delete or modify scrum configuration

Fig.6: Snapshot 4 configure scrum details



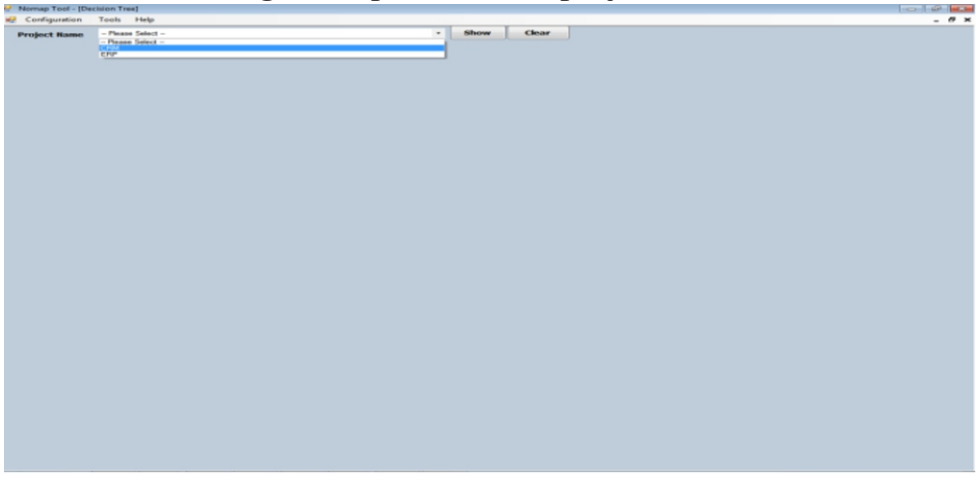
6. User can add, delete or modify task configuration

Fig.7: Snapshot 5 configure task details



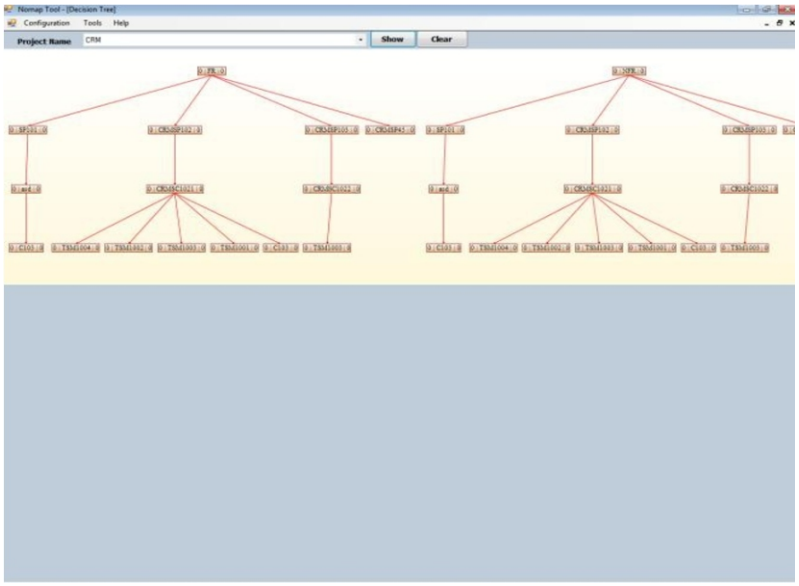
7. Select project name.

Fig 8: Snapshot 6 select project name



8. Click on show button to generate decision tree of FRs and NFRs.

Fig.8: Snapshot 7 decision tree of FRs and NFRs



V. CONCLUSION

At the end, we conclude that appropriate managing of NFR has very significant role in design stage of software development. It can provide help to get better the features and happiness of customers. It also helps the developer to improve the development time as well as cost which is used by the developer to revisit those requirements in further development stages. Failure of Projects, prolonged delay or significant increase in cost are some drawbacks of ignoring NFRs. Success of any software depends on the customer satisfaction which is basically depends on maximizing NFR elicitation and integration in the software product. Lack of support to NFR is responsible for the failure of system.

In this paper we have proposed NFRs elicitation technique. This paper proposes a design of tool to develop NFRs by considering that NFR plays sensitive role in the success of the software system.

REFERENCES

[1] Andrea De Lucia and AbdallahQusef "Requirements Engineering in Agile Software Development", Journal of Emerging Technologies in Web Intelligence, Vol. 2, No 3, August 2010.

[2] JayakanthSeinivas and Kristina Lundqvist, "Using agile methods in software product development: A case study", Information Technology: New Generations, Third International Conference, vol.0,1415-1420,2009.

[3] Weam M. Farid / Frank J. Mitropoulos "NORMATIC: A Visual Tool for Modeling Non-functional Requirements in Agile Processes" , 2012.

[4] A.Sidky and J.Arthur, "Determining the Applicability of Agile Practices to Mission and Life-Critical Systems",Proceedings of the 31st IEEE Software Engineering Workshop, IEEE Computer Society,2007,pp.

[5] Kaushal Pathak an AnjuSaha "Review of Agile Software Development Methodologies", International Journal of Advannced Research in Computer Science an Software Engineering,Vol.3,Issue 2,February 2013.

[6] Malik Qasaimeh, HosseinMehrfard, AbdelwahabHamou-Lhadj, "Comparing Agile Software Processes Based on the Software Development Project Requirements", CIMCA 2008, IAWTIC 2008, and ISE 2008.

[7] Md. Mijanur Rahman and Shamim Ripon, "Elicitation and Modeling Non-Functional Requirements – A POS Case Study", December 20, 2012.

[8] FraukePaetsch, Dr. Armin Eberlein, Dr. Frank Maurer, "Requirements Engineering and Agile Software Development".

[9] PratimaSingh and Anil Kumar Tripathi, "Issues In Testing of software with NFR", International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.4, July 2012.

[10] David Ameller, OriolCollell, Xavier Franch, "ArchiTech: Tool Support for NFR- Guided Architectural Decision-Making", June 25, 2012.

[11] NupurChugh and Aditya Dev Mishra,"Assimilation of Four Layered Approach to NFR in Agile Requirement Engineering", International Journal of Computer Applications (0975 – 8887) Volume 78 – No.5, September 2013.

[12] Gayane Azizyan, Miganoush Katrin Magarian, Mira Kajko-Mattson, "Survey of Agile Tool Usage and Needs", Agile Conference,,IEEE,2011.

[13]A. Matoussi, R. Laleau. "A Survey of Non-Functional Requirements in Software Development Process"October 2008.

[14]Chian Wen Too, Sa'adah Hassan, Jamilah Din, and Abdul AzimAbd. Ghani, "Towards Improving NFR Elicitation in Software Development", International Journal of Information Technology & Computer Science(IJITCS) (ISSN No : 2091-1610) Volume 7 : No : 1 : Issue on January / February, 2013.

[15] G. Goth, "Agile Tool Market Growing with the Philosophy", 2009, IEEE Software, 26(2), pp. 88-91

[16] G.Sousa, G.Silva, and J.Castro, "Adapting the NFR Framework to aspect Oriented Requirements

Process", Information and Software Technology Journal, vol.49, no. 11-12, pp. 1162-1171, 2007.

[18] S. Ambler, "Agile Requirements Modeling", 2012 available at : http://www.agilemodeling.com /essays/agile Requirements.htm

[19] WaleedHelmy, AmrKamel and Osman Hegazy, "Requirements Engineering Methodology in Agile Environment", International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012

[20] A. Ananda Rao and M.Gopichand, "Four Layered Approach to Non-Functional Requirements Analysis", International Journal of Computer Science Issues, Vol. 8, Issue 6, No 2, November 2011

[21] Saravana. K.M, G. N. Basavaraj, Rajkumar, Dr. A. Kovalan, " Case Study On Agile User Stories Prioritization Using Imaginative Standard", International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 5, September- October 2012

ISSN No.-2321-8711

Enhancement of Y-Model for Service-Oriented Model

Varun Kumar[1], Rajpoot Pragya Siddhi[1], Dr. Anurag Singh Baghel[2]
[1]M.Tech (Software Engineering), Gautam Buddha University, Greater Noida.
[2]Phd, Assistant Professor, Gautam Buddha University, Greater Noida.

ABSTRACT

Service-Oriented Architecture (SOA) has received significant attention. In SOA, services are discovered and composed according to the goal specification. SOA can rightly be called an improvement over Component Based Software Engineering (CBSE), however it still uses CBS technologies for development of services. We aim to present a novel process for development, publishing and use of services. In addition, this paper presents a hypothetical process for composition of these services with more complex services, to cater to the end user.

Keywords: Component-Based Software Development (CBSD), Service-Based Development.

I. INTRODUCTION

Component-Based Development (CBD) is older better-defined, and better –known in the software engineering environment. In the past a few years, rapid progress has been occurred in the field of Service-Oriented Development (SOD), which represents a paradigm shift from Component-Based Development (CBD) to the Service-Based Development. “Figure 1” shows Component-Based Development.

The Service-Oriented Computing (SOC) paradigm refers to the set of concepts, principles, and methods that represent computing in Service-Oriented Architecture (SOA) in which software applications are constructed based on independent component services with standard interfaces (W.T. Tsai, et al. 2005).

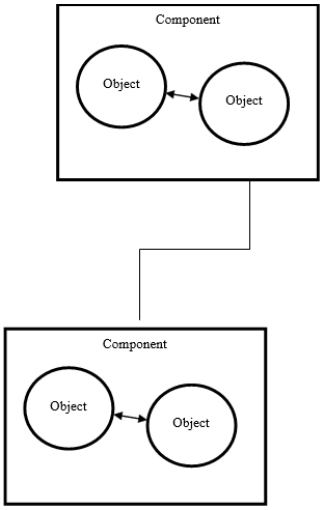


Fig 1: Component-Based Development

Service-Oriented Development is evolution of Component Based Development, Interface based Design (Object-Oriented) and Distributed Computing. “Figure 2” shows Service-Based Development.

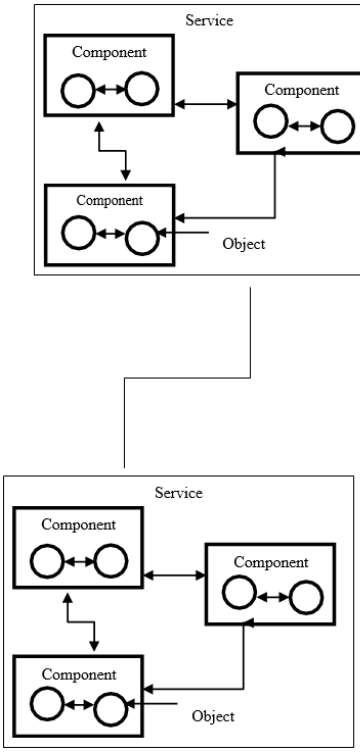


Fig 2: Service-Based Development

1.1 SOA Entities: Service-Oriented Architecture is an architectural style that defines an interaction model between three main functional units: the application builders (also called service requestors), the service brokers (or publishers), and the service developers (or providers) as shown in “Figure 3” in which the consumer of the services interacts with the service provider to find out a service that matches its requirement through searching registry (Y.H Wang, et al. 2011).

1.2 Service Provider: Service provider is an entity network addressable. It can accept and perform the request from service consumer, and service provider can make request by using mainframe, service or other software. It provides the definite service description and the implementation of the services. The service provider can be a component, or other type of software system that fulfils the service consumer's requirements.

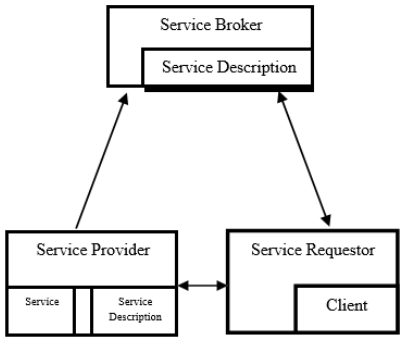


Fig 3: SOA Entities

1.3 Service Broker: Service broker is similar to service dictionary, which has Universal Description, Discovery and Integration (UDDI) function, responsible for receiving and storing the contract from service provider, then provides contract to service consumer which need it.

1.4 Service Consumer/Application Builder: Service Consumer is the entity in SOA that looks for a service to execute a required function. Service consumer can be end-user, application or some type of software module that needs the services. It will be looking for suitable service in service registry, then confirm the location of service, bind service, and perform the function of service. Service performs services on the basis of request and response on the basis of contract.

2. BACKGROUND

The development of service-oriented system faces many challenges, as stated by SOC research studies. Many question need to be answered, such as:

- Which activities should be considered for the development of the service-oriented system?
- How to make modern service-oriented system from legacy system?
- How to attain suitable services from business process?

In this context, various methodologies have been proposed to demonstrate the lifecycle of the service-oriented development. None of the existing service-oriented methodologies cover all issues of the service-oriented development; they are only relevant to specific aspects of the service-oriented development (M.F.Gholami, et al. 2010).

Model proposed by (Y.Kim and H.Yun2006) is based on identifying services from the legacy system. These models have lack of full coverage of the service-oriented development cycle, lack of supportive documents on their practical use.

(S. Moosavi et al. 2009) proposed a method for service-oriented design which is based on identifying different types of variations then establishing variability model for establishing adaptable services.
Luiz Fernando's Y model is most widely used for the development of component based software. Objective of this thesis is to adapt the existing Y-model to Service-Oriented Development to achieve the following:

- To cover most of development activities of service provider, service consumer and service broker.
- Legacy system adaption and reusability to build individual service and complete application.
- Service Analysis

- Variation Analysis in the service.
- SWOT analysis.
- To consider communication among components.
- Service-Oriented testing to test services.
- Quality assurance after integration.

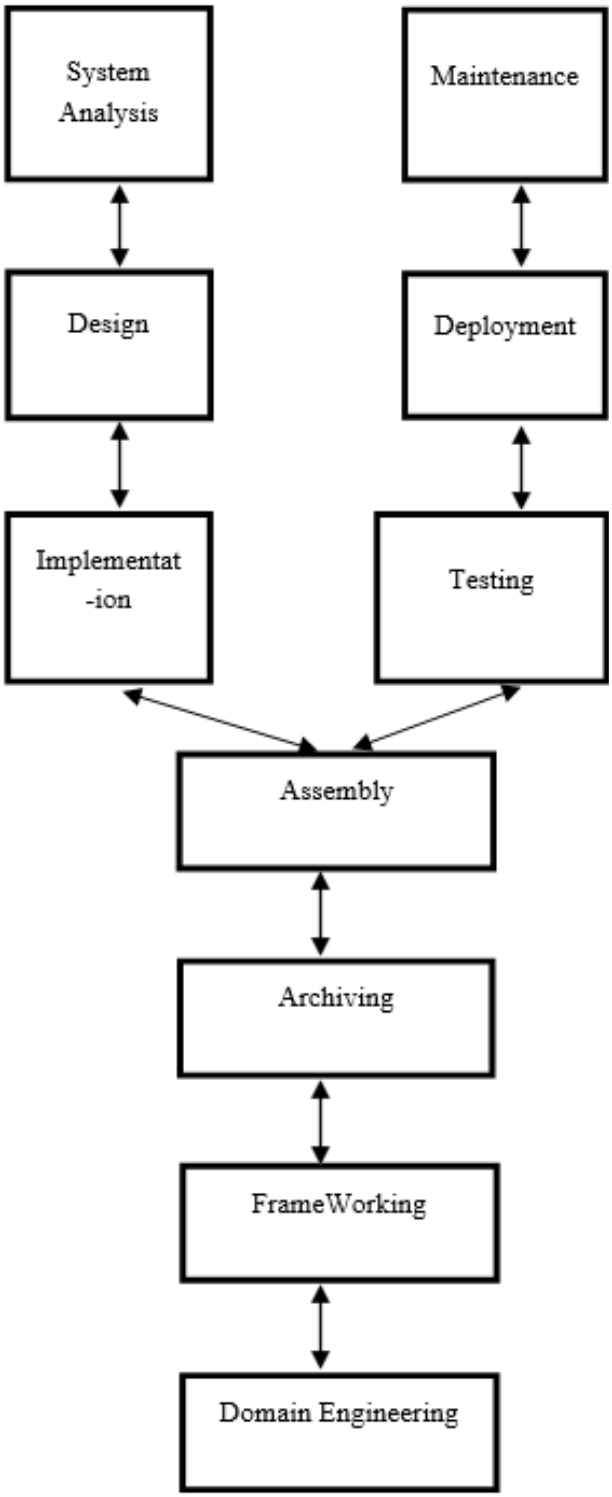


Fig 4: Y-Model for Component-Based Development

3. ENHANCED Y-MODEL

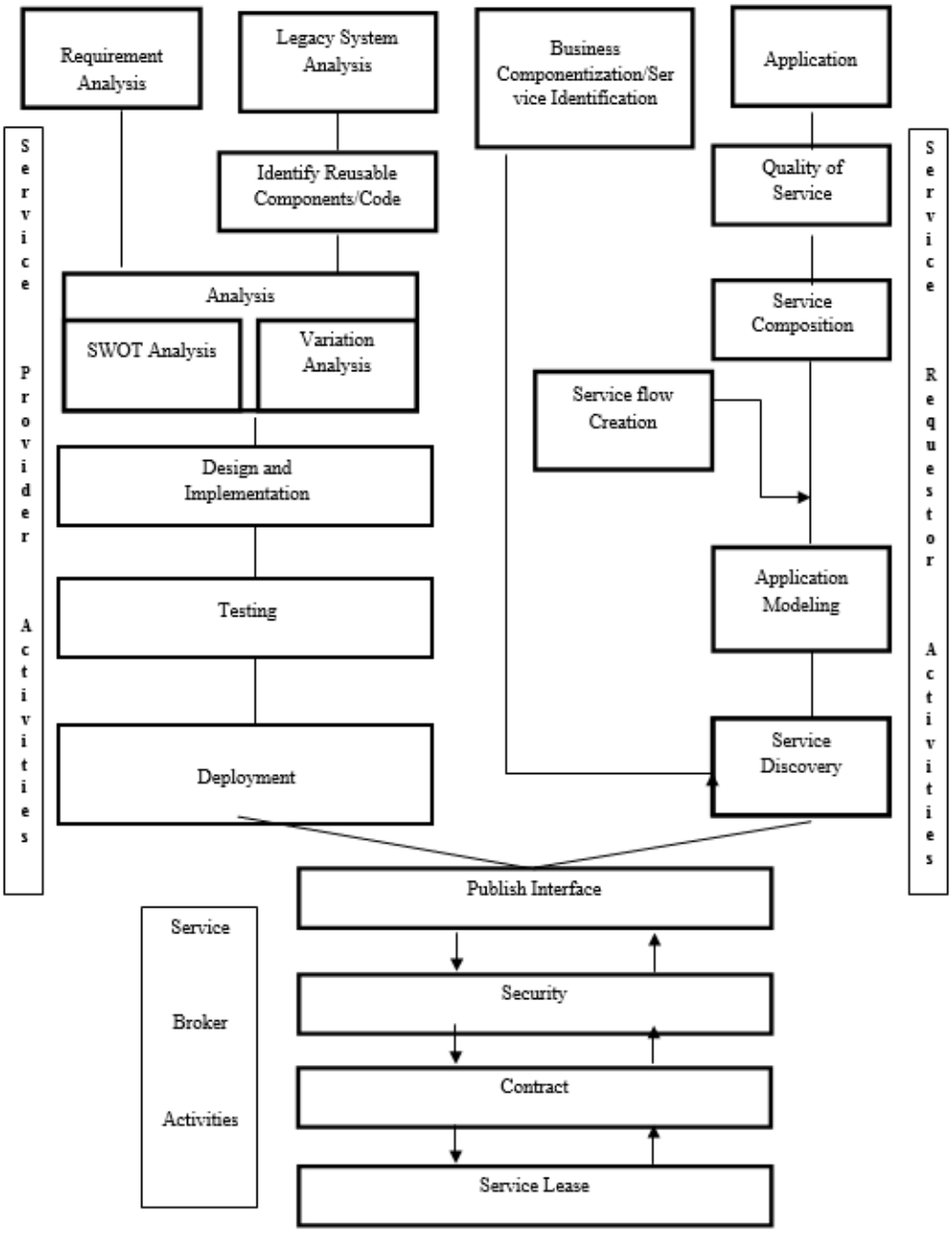


Fig 5: Enhanced Y-Model

3.1 Service-Provider Activities

Service Provider develops the loosely coupled services and these services are deployed on the provider sides.

3.1.1 Requirement Analysis: Service-Oriented Requirement Engineering (SORE) is different from traditional requirement. SORE shares some activities with the traditional requirement engineering but its focus is to identify reusable services. Before starting requirement analysis some design should be made. Some of the SORE features are:

3.1.1.1 Reusability-orientation: SOA emphasize on reusability. Once item is published, it can be

reused by other.

3.1.1.2 Domain Specific: Most requirement process/techniques are often domain independent, as they are often application to a wide range of applications and domains.

3.1.1.3 Model-Driven Development: SORE assumes a model-driven approach. In which series are modeled using modeling language. UML, BPMN are used as modeling language.

In the beginning informal and unstructured requirements exists. Unstructured and informal requirements are translated into structured and formal model.

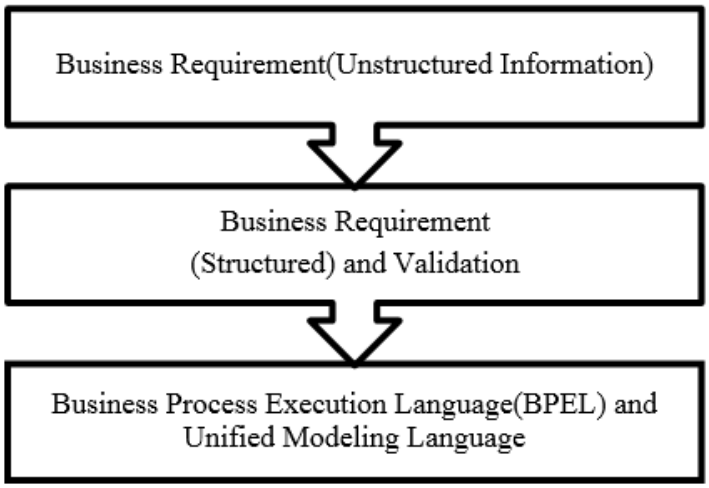


Fig6: Service Requirement Modeling

After requirement analysis model is transformed into Unified Modeling Language (UML) or Business Process Modeling Notation (BPMN) to start design and integration activities is shown "Figure 6".

In the detailed study of legacy system Use Case Model help to identify reusable services. From legacy system, reusable services or reusable module are identified.

3.1.2 Legacy System Analysis: A service is implemented by complex collaboration of many components or objects. Legacy systems are one of the most valuable software assets to reduce development time, cost, and effort and to increase reliability. In the legacy system analysis, Use Case Model helps to identify unit service or components for reuse and in analysing workflow of reusable units. In legacy system analysis, reusable components or code are identified.

3.1.2.1 Reusable Component Identification: After breaking up a service into components reusable components are found out and their dependency is analyzed.

3.1.2.2 Reusable Code Identification: Aspect oriented programming and program slicing helps to find out reusable code from a service or component. Identified code can be used in the implementation of a service.

3.1.3 Analysis Activity: In proposed model we introduced two type of analysis before starting design

activity, which are as follows:

3.1.3.1 SWOT Analysis: Same service can be developed by different service providers. One can have higher probability to compete with other. Here 'S' stands for Strength, 'W' stands for Weakness, 'O' stands for Opportunity and 'T' stands for Threat. Opportunity is the area which attracts the enterprise, in the area, enterprise owns competition to use the services. Threat means the challenges formed by unfavorable development trend in environment. Every enterprise should periodically check its strength and weakness. Strength means one service is better with other service.

3.1.3.2 Variation Analysis: Services are not created for predefined users but are created for many unknown users and to adapt new concepts. Service variation can exist in business process or in unit service; variation can exist in workflow, composition, interface and logic (Chang H. S. et al., 2007). On the basis of variation analysis variation model is identified for establishing adaptable services. Some basic variations are:

3.1.3.2.1 Workflow Variability: A business process is carried out in a sequence of unit services, called workflow. In a workflow, some unit service may not be requested to a particular service user and some part of the sequence may be differently performed to other particular service user. Thus, a service (or a part of service) can perform in different ways for different users.

3.1.3.2.2 Composition Variability: A business process composes several unit services to fulfill the end users' requirements. For one unit service in the workflow, theremay be more than one possible service interfaces which implement the service with different implementation logics or quality attributes. In this case, variation occurs on selecting the most appropriate service interface. That is, depending on user requirements, different service interfaces of the implementation can be composed and it is called Composition Variability.

3.1.3.2.3 Interface Variability: It occurs when the interfaces of unit services do not match to the interfaces of services published in UDDI service registry. A unit service is a logical unit of service that is composed into various business processes. The services registered in a service registry are published and available services, of which interfaces are typically derived from implemented service components and interface specifications of some standards.

3.1.3.2.4 Logic Variability: A service component includes operations for providing the functionality of unit services. The service component should provide different logics depending on particular requested services. It is called Logic Variability.

3.1.4 Design and Implementation: On the basis of requirement analysis, identified reusable services and variation model, Design activity is started. UML, BPMN, BPEL and Use Case Model helps in design activity. These services can be implemented in any language like C, C++, Java etc.

3.1.5 Testing of the Service: Traditionally in component-based software system three types of testing are performed: unit testing, integration and system testing. Service provider performs unit testing, service testing, integration testing and system testing for testing services. For service-oriented integration unit testing and group testing of interface is performed.

3.1.5.1 Unit Testing: Unit testing in SOA can be carried out just like in Component-Based Software Development. In unittesting individual unit of code is tested. In unit testing focus is on functional correctness and hence correct implementation of algorithm. Unit testing can be performed as white-box and black-box, manual and automated, code-based and model-based.

3.1.5.2 Service Testing: Integration testing of SOA is different from traditional testing. In SOA new layer of testing service testing is performed. In service testing, there is less focus on correct implementation of algorithms but on the integration of the functional units inside the component (or service).

3.1.5.3 Integration Testing: Services are more loosely coupled than components. In contrast to the CBS approach, integration testing cannot rely on homogeneous components with tightly connected interface, therefore adaptability and distribution of SOA demands additional consideration for integration testing. Especially the effect of message racing and its implication have to be considered during system development and should be tested thoroughly. Message racing in this context refers to the situation where messages are not received in the same order as they were sent.

3.1.5.4 System Testing: Service testing is based on high-level usage scenario and business requirements that have been defined by business analyst or customers. Use interface testing is most appropriate to carry out system testing as the system will be carried as a whole. Generally system testing is performed after the deployment of the service on the server.

3.1.6 Service Deployment: The goal of this activity is to make the system available for use. Services are deployed in an operational environment on the service provider side. In this environment service become available to service consumer. Services deployment location is transparent. Service consumer does not have any need to know the location of the service. Universal Description, Discovery and Integration (UDDI) protocol is added with the deployed service to make it discoverable for service consumer. System testing is performed after deployment to check service is working correctly or not.

After the deployment of the service broker activities start to make the market for the services.

3.2 Service Broker Activities:

Service broker performs third party role in SOA. Broker is intermediary between service provider and consumer. Service Broker uses a repository (or registry) where serviceproviders publish interfaces of

services. These interfaces are queried by consumer to meet there required. A service broker provides contract, security protocols and data format (incoming data and outgoing data).

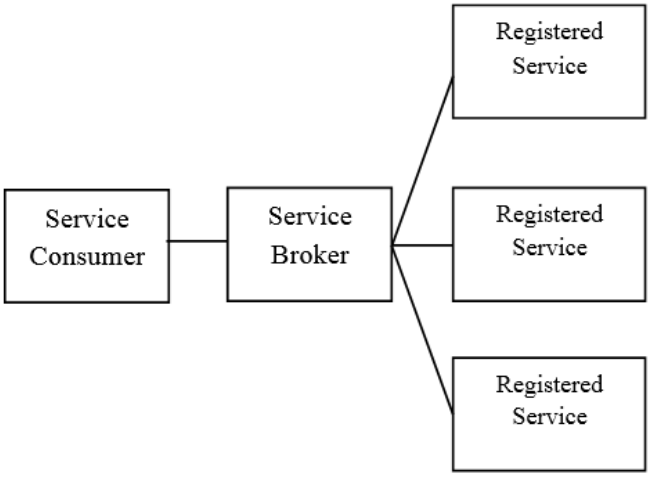


Fig 7: Service Broker Registered Services

A broker can publish services of different service providers and a consumer can invoke services of different brokers. The broker may invoke one or many services concurrently depending on how it is configured. If many services are invoked, it may wait for all to complete or just one to complete before notifying the client, if running synchronously.

3.2.1 Security: Service broker provides security to service consumer. Security allows consumers to reach their application securely, without having complex back end configuration.

There is a lot of personal information and potentially secure data that people store on their computers, and this information is now being transferred to the service. This makes it critical to understand the security measures that their service provider has in place, and it is equally important to take personal precautions to secure their data.

The first thing consume must look into is the security measures that their service provider already has in place. These vary from provider to provider and among the various types of services.

- What encryption methods do the providers have in place?
- What methods of protection do they have in place for the actual hardware that consumer's data will be stored on?
- Will they have backups of my data?
- Do they have firewalls set up?

Many service providers have standard terms and conditions that may answer these questions. A small business user may have slightly more room to discuss the terms of their contract with the broker and will be able to ask these questions during that time.

There are many questions that consumer can ask, but it is important to choose a service provider that considers the security of service consumer's data as a major concern.
Since the processing takes place on the server and there is no hard drive, there is less chance of the malware invading the device. Also, since thin client do not work without a server, there's less chance of them being physically stolen.

3.2.2 Service Contract: A service contract is an interface that defines the message type used by service provider and service consumers to exchange message (Skonnard, 2005). Service contracts specify one or more operations that represent individual message exchange or a request/reply message exchange. Service broker provides contract to service consumer provided by service provider.

The service contract attribute defines the service interface and the operation contract attribute specifies methods of the services. A service contract is converted into Web Service Definition Language (WSDL) definition.

A web definition file has the following two parts:
1. A technology-independent abstract description that describes the service interface (port Type), its methods (operations) and their input-output parameters.
2. A concrete description for binding the communication protocol with the port (end point) of the service.

3.2.3 Service Lease: Service lease specifies the amount of the time that a service contract is valid. Service broker grants it to consumer, only in this duration a consumer can request for a specified service. When the lease runs out, the consumer must request a new lease from the registry. The lease is necessary for services that need to maintain state information about the binding between the consumer and provider. The lease defines time for which the state may be maintained. It also further reduces the coupling between the service consumer and the service provider, by limiting the amount of time consumers and providers may be bound.

After publishing of the services they are ready to be used by service consumer.

3.3 Service Consumer Activities

Service consumer invokes the published services on the network on the basis of contract and lease. First of all service consumer identify the service to adapt into the application. Activities performed by the service consumer are as follows.

3.4 Service Discovery: After componentization service consumer discovers the service on the network after querying service repository with required characteristics. Discovery can be done in various domains. Service discovery and service composition is done by syntactical matching of Web Service Description Language (WSDL) or by semantic matching of Web Service Description Language (WSDL). Two basic domains are:

3.4.1 Structured Discovery Approach: Structure discovery is a syntactical approach to find out services which is based on interface description and the definition of the data (Input/Output) message exchanged between the communications partners. In structured discovery WSDL are matched to check that input/output message of service description are semantically linked via inheritance relationships or not.

3.4.2 Lexical Discovery Approach: Lexical discovery approach use natural language description. Web service operation names usually contain some terms describing their functionality. WSDL and other standards allow embedding natural language description. Lexical algorithms remove stop words from those descriptions; find synonyms using lexical databases like WordNet.

3.5 Application Modeling: After business componentization services are identified and after finding required services process is modelled. Business processes should be modelled and optimized in order to map appropriately to the services that satisfy business goal. During the modeling phase, service dependencies are found out with the help of service flow creation. In service flow creation message flow among services is found out.

After finding message flow among services communication among services is considered. In CBS implementation In a CBS implementation, the communication would most likely be handled by synchronous calls between the components. Synchronous communication means that the initiator is blocked from further computation until the requested component is providing the desired answer. The SOA approach however demands a loose coupling that allows more flexibility and better distribution of the components. Therefore asynchronous channels are used in addition to the synchronous ones. Asynchronous channels provide different reliability degrees or instance they can guarantee that every message is received either exactly once (EO) or even exactly once in order (EOIO). Both types of channels have mechanisms to ensure that each sent message is received exactly once, thus preventing message loss and multiple receiving of the same message. While messages on an EOIO channel are received in the same order as they are sent, messages on the EO channel may overtake each other. On EOIO channels, message racing is usually prevented by re-sorting the messages at the receiver side.

3.6 Service Composition: Service Consumer composes the services in order to provide a new or

complex service to final user. In 2001 IBM introduced WSFL (Web Service Flow Language) to represent workflow of composite services. Same time Microsoft presented XLANG an extension of WSDL. In 2003, IBM and Microsoft combined them into a new Language BPEL4W (Business Process Execution Language). In 2004 BPEL4W has changed for WS-BPEL.

3.7 Quality of Service: There are two types of requirements; functional and non-functional requirements. Functional requirements express what software is aimed at. Non-functional requirements describe its quality parameters. Developers mostly focus on functional requirements during the service development (service identification, service design, service implementation, service usage) and mostly ignore non-functional requirements.

Galster and Bucherer (2008) have proposed taxonomy for identifying and specifying non- functional requirements in service oriented development. Quality model in service-oriented development process has four stages: service design, service discovery, service composition, runtime. Non-functional requirements vary stage to stage. Non-functional requirements can be categorized as: process requirement, non-functional service requirements (Sadiq J. et al. 2011).

3.7.1 Process Requirement: Process requirements are placed on the service-oriented development process. This process consists of service design, service discovery, service composition, and runtime. Process requirements become important when the customer wishes to influence the development process (e.g., in governmental or federal contracts). Some process requirements are:

- Implementation requirements
- Composition Requirements
- Standard requirements
- Cost constraints
- Time constraints
- Documentation requirements

3.7.2 Non-Functional Service Requirement: Non Functional Service Requirements are placed directly on the service-oriented system or an individual service under development. It is important to mention that these non-functional requirements can be derived directly from user needs, compared to the previous two categories of non-functional requirements which require more domain and environmental analysis.

Non Functional Service Requirements are difficult to handle, due to the highly distributed nature of

service-oriented applications:

3.7.2.1 Usability: Usability in service-oriented systems is impacted by the data granularity used by services and by normal usability operations.

3.7.2.2 Reliability: Reliability can be separated into message reliability and service reliability. Message reliability addresses reliability of communication channels of the network over which services are made available. It therefore impacts the reliability of the whole system. Service reliability refers to the correct operation of an individual service. The use of standards can ensure a reasonable level of reliability of a service.

3.7.2.3 Performance: One individual service can have acceptable performance but due to network delays, high modularity and low cohesion within the system, the overall performance might decrease.

3.7.2.4 Safety : Often, safety is a quality which does not relate to the service alone but to the whole system. Standards can support safety, but for service-oriented systems not many mature standards exist . The risks involved in safety usually result from problems with the functionality of a system.

3.7.2.5 Security: As with safety, not many mature standards exist that could support this property. Failure in security can cause risks which impact safety.

3.7.2.6 Interoperability: Interoperability is the most prominent benefit of service-oriented systems. Through the use of standards (e.g., WSDL) interoperability is supported from a technical point of view. To allow full interoperability, a service must also provide semantic interoperability.

3.7.2.7 Availability: Service level agreements (SLA) cover availability agreements. To improve availability within a system, dynamic service discovery and composition in case of a service fall-out would increase availability.

3.7.2.8 Adaptability: No standards exist to support adaptability of a service so it is up to theservice user and service provider to manage adaptability.

3.7.2.9 Modifiability: An individual service can be difficult to modify, but a service-oriented system directly supports modifiability. A service interface must be designed carefully because changes that impact service users might be difficult to identify if the service is provided form external sources.

After the composition of the services application is ready for use.

4. CONCLUSION

In this thesis we have focused on WHAT should be done for service-oriented development instead of HOW it should be done. Proposed model presents service-oriented development process in a hierarchically structure and well defined way so that they can be used as reusable services. Service based development is much faster than component based software development; it reduces cost and effort of

the system development using legacy services. Enhanced model offers flexible, loosely coupled and reusable service logics. Proposed model utilizes external (third party) services to supply business process. Model manages changes (variations) in the business process.

5. REFERENCES

1. *Crnkovic I. and M. Larsson (2001). "Component-Based Software Engineering – New Paradigm of Software Development". Proceedings of the MRTC Publication.*
2. *"Security in a Web Services World: A Proposed Architecture and Roadmap", IBM/Microsoft, http://msdn2.microsoft.com/en-us/library/ms977312.aspx, v1.0, 7, 2002.*
3. *Wang, Y., Stroulia (2003), E.: "Flexible interface matching for web-service discovery. In: Web Information Systems Engineering (WISE)". Proceedings of the Fourth Inter- national Conference on, 147–156.*
4. *Tsai T.W. (2005). "Service-Oriented System Engineering: A New Paradigm". Proceedings of the IEEE International Workshop on Service-Oriented System Engineering, 0-7695-2438-9/05, pp. 3-6.*
5. *Michael N.Huhns and Munindar P. Singh (2005). "Service-Oriented Computing: Key Concepts and Principles". Proceedings of the IEEE Journal Internet Computing, 1089- 7801/05, pp. 75-81.*
6. *Alan W. Brown, Simon K. Johnston, Grant Larsen, Jim Palistrant (2005). SOA Development Using the IBM Rational Software Development Platform: A Practical Guide.*
7. *Kim Y. and H. Yun (2006). "An Approach to Modeling Service-Oriented Development Process". Proceedings of the IEEE International Conference on Services Computing, 0- 7695-2670-5, pp. 273-276.*
8. *Ramollari E., D. Dranidis, and A. J. H. Simons (2006). "A survey of service oriented development methodologies". Proceedings of the second Young Researchers' Workshop on Service Oriented Computing, Leicester, U.K., pp. 1-6.*
9. *Kokash, N., van den Heuvel, W.J., D'Andrea, V. (2006). "Leveraging web services discovery with customizable hybrid matching". Proceedings of the Fourth Springer Interna-tional Conference on Service Oriented Computing (ICSOC), pp. 522–528.*
10. *A.W. Brown and S.K. Johnston (2006), "A Model-driven Development Approach to Creating Service-oriented Solutions",Proceedings of ICSOC06, International Conference on Service Oriented Computing, pp. 624-636.*
11. *Zhang Y., t. Yo, Raman K. and K.J.Lin (2006). "Strategies for Efficient Syntactical and Semantic Web Services Discovery and Composition". Proceedings of the eight IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC/EEE'06), 0-7695-2511-3.*
12. *Brown W. A., S. Iyengar, S.K. Johnson (2006), "A Rational Approach to Model-Driven Development", IBM Systems Journal, pp.463-480, Vol. 44,http://www.research.ibm.com/journal/sj/453/brown.pdf*
13. *Papazoglou M. P., P.Traverso, S. Dustdar and Leymann F. (2007). "Service-Oriented Computing: State of the Art and Research Challenges". Proceedings of the IEEE Journal, 10.1109/MC.2007.400, pp. 38-45.*
14. *Bai X., D. Xu, Dai G. (2007). "Dynamic Reconfiguration Testing of Service-Oriented Architecture", Proceedings of the 31st IEEE Annual International Computer Software and Application Conference (COMPSAC), 0-7695-2870-8/07.*
15. *Tsai T.W., Jin Z., P.Wang and Wu B. (2007). "Requirement Engineering in Service- Oriented System Engineering". Proceedings of the IEEE Conference on e-Business Engineering, 0-7695-3003-6/07.*
16. *Xu.X, Mo.T, Wang.Z (2007). "SMDA: A Service Model Driven Architecture". Processing in the 3rd International Conference on Interoperability for Enterprise Software and Applications, pp. 28-30.*
17. *S. Stein, Katja B., and M. E. Kharbili. "Enabling Business Experts to Discover Web Services for Business Process Automation". http://www.ids-scheer.com/soa/*
18. *Chang H. S. and S. D. Kim (2008). "A Variability Modeling Method for Adaptable Services in Service-Oriented Computing". Proceedings of the eleventh IEEE Conference on Software Product Line, 0-7695-2888-0, pp. 261-268.*
19. *Lopez S. M., C. J. Acuna, E. C. Cuesta and Marcos E. (2008). "Defining Service- Oriented Software Architecture Models for a MDA-based Development Process at the PIM level". Proceedings Seventh Working IEEE/IFIP Conference on Software Architecture, 0-7695-3092-3, pp. 309-312.*
20. *Arsanjani et al. (2008). "A Method for Developing Service-Oriented Solutions (SOMA)". Proceedings of the IBM System Journal, 0018-8670, vol. 43.*
21. *Engels G. and M. Assmann (2008). "Service-Oriented Enterprise Architectures: Evolution of Concepts and Methods".Proceedings of the twelfth IEEE Enterprise Distributed Object Computing Conference, 978-0-7695-3373-5, pp. xxxiv – xliii.*

22. *DijkmanR.M. and M. Dumas (2008). "Service-oriented Design: A Multi-viewpoint Approach", Proceedings of the CiteSeer International Journal of Cooperative Information Systems.*
23. *M. Galster, E. Bucherer (2008), "A Taxonomy for Identifying and Specifying Non- functional Requirements in Service-oriented Development", Congress on Services, IEEE Computer Society.*
24. *Moosavi S., M. A. Seyyedi and N. Moghadam (2009). "A Method for Service Oriented Design", Proceedings of the Sixth IEEE International Conference on Information Technology: New Generations, 978-0-7695-3596-8, pp. 290 – 295.*
25. *Gurupur V. and M. M. Tanik (2009). "Abstract Software Design Framework: A Semantic Service Composition Approach". Proceedings of the IEEE Conference on SouthEastCon,978-1-4244-3976-8, pp. 295-300.*
26. *Wieczorek S. and A. Stefanescu (2009). "Service Integration: A Soft Spot in the SOA Testing Stack". Proceedings of the fifth Central and Eastern European Software Engineering Conference in Russia (CEE-SECR), 978-1-4244-5664-2, pp. 211-216.*
27. *Balfagih Z., M. F. Hassan (2009), "Quality Model for Web Services from Multi- Stakeholders' Perspective", Proceedings of theInternational Conference on Information Management and Engineering.*
28. *Dan A. and N. Priya (2009). "Dependable Service-Oriented Computing", Proceedings of the IEEE Internet Coputing Journals, 1089-7801, pp. 11-15.*
29. *D. Becker (2009). "Service component architecture (SCA) lets you invokecomponents from different technologies",http://www.ibm.com/developerworks/opensource/library/os-apache-tuscany-*
30. *sca/index.html?ca=drs-.*
31. *Bai L., J. Wei (2009). "A Service-Oriented Business Process Modeling Methodology and Implementation". Proceedings of the IEEE International Conference on Interoperability for Enterprise Software and Applications, 978-0-7695-3652-1.*
32. *Web Services Reliable Messaging (WS-Reliable-Messaging), Vers. 1.1. OASIS Consortiom. Online at: http://docs.oasis-open.org/ws-rx /wsrm/v1.1/wsrm.pdf*
33. *Gholami F. M., J. Habibi, F. Shams, and S. Khoshnevis (2010). "Criteria-Based evaluation framework for service-oriented methodologies", Proceedings twelfth IEEE International Conference on Computer Modeling and Simulation, pp.122-130.*
34. *Yang. L, Xing. K, Lee. S (2010), "A new conceptual life cycle model for Result-Oriented Product-Service System development". Proceeding of the IEEE International Conference on Service Operations and Logistics and Informatics (SOLI).*
35. *Fahmideh M., M. Sharifi, Jamshidi P., F. Shams and Haghighi H. (2011). "Process Patterns for Service-Oriented Software". Proceedings of the fifth IEEE Conference on Research Challenges in Information Science (RCIS),978-1-4244-8670-0, pp. 1-9.*
36. *Leite S. D., C. Mary, F. Rubira, Castor F.(2011). "Exception Handling for Service Component Architectures". Proceedings 13th IEEE International Conference on Computer Modeling and Simulation.*
37. *Sadiq J., A Mohsin and F. Arif (2011). "Quantifying Non-functional Requirements in Service Oriented Development". Proceedings of theIEEE International conference on Frontiers of Information Technology (FIT), 978-0-7695-4625-4, pp. 224-229.*
38. *Wang H. Y., J.C.Liao and C.H.Tsai (2011). "Objective Concept for Evaluating Service- Oriented Architecture", Proceedings eighth international conference on Fuzzy System and Knowledge Discovery (FSKD), 978-1-61284-181-6.*

ISSN No.-2321-8711

Automata Theory for Software Development

Avdhesh Mann[1], Dr. Rajneesh Talwar[2], Dr. Bharat Bhushan[3], Rakesh Gupta[4]

[1]B.Tech (Computer Science), U.I.E.T , Kurukshetra University, Kurukshetra (Haryana), INDIA

[2]Principal, SWIFT Group of Institutions, Rajpura (Patiala),Punjab, (INDIA)

[3]Associate Professor & Head, G.N.K, Yamunanagar (Haryana), INDIA

[4]Programmer , D.B.S.C.R., Government Polytechnic Sampla (Rohtak), Haryana, INDIA

A B S T R A C T

In this paper we discussed about a software i.e. Compiler. In each language users deal with compiler.But how the compiler will work that depend totally on language features. So here we discuss if someone want to design a new compiler what type of theory he or she should study before making that compiler according to the new features which he or she should use in his program.

INTRODUCTION

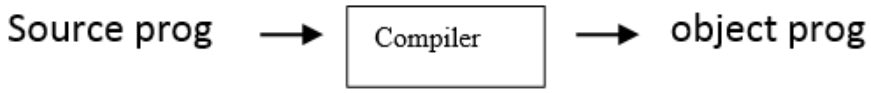
As we know, software development is directly related to customer need. Software is developed with systematic approach. Means with the different models of software engg. like spiral model, water fall model, RAD model, v-model, iterative model and so on .We take the water fall model now : it has different phases to develop the project and these phases are like this:

Specification
Design
Coding
Testing
Operation and maintenance

If we develop the software using this approach means before coding if we do the designing part and before designing if we clear our all specification than we can get a better quality product according to that customer need. However we can use other models by analyzing customer requirements.

HOW TO RELATE SOFTWARE DEVELOPMENT WITH AUTOMATA

It's a smart work .As every body knows that the software is a collection of programs that do need full task for the users. Now we talk about all system programs like translators e.g. compiler, interpreter, assembler, etc. Now here we deal only with one translator i.e. compiler. So it means compiler is also a software .A compiler is usually used to convert high level language in to machine level language. For different languages there can be different types of compilers like c compiler, java compiler etc.



Now the compiler software is also made up of different phases. Its different phases are

a) Lexical analysis

b) Syntax analysis

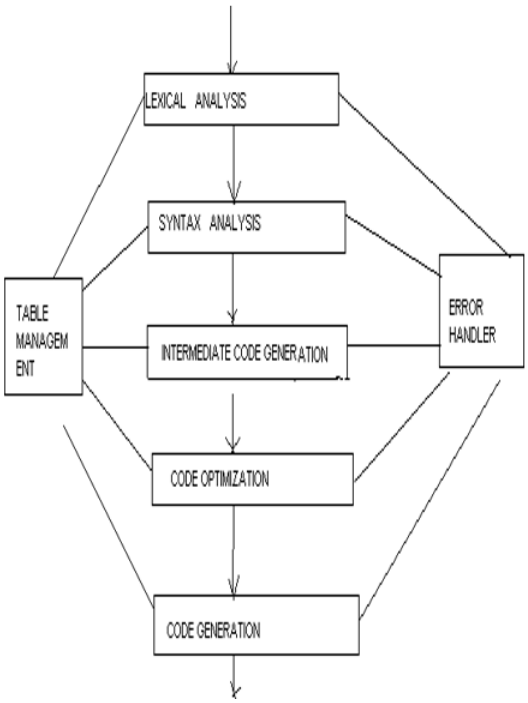
c) Intermediate code generation

d) Code optimization

e) Code Generation

f) Error Handling

g) Table Management



We can say all these phases behave like a software model. By studying all these phases we can design diifferet types of compiler according to language features. So these phases are analyzed by automata theory for making compilers.

I. WHAT IS AUTOMATA THORY

An automaton is generally called a Machine. It does all the work automatic. Like automatic packing m/c etc. We just need to give input, we get the output. We can say that it is a basic machine. Example:

Input	Output
(0, 0)	0
(0, 1)	1
(1, 0)	1
(1, 1)	1

Such type of basic m/c doesn't deal with states.

Computer is also a machine. We give the input, get the output. But in this m/c, we have to deal with states.

State: The configuration of machine at particular instant of time.

Like holding state, wait state, processing state etc. In this, record of state is kept. So we always have to use storage elements i.e. auxiliary memory. .

So such type of automaton (or m/c) has following characteristics:

1. **Input:** At each discrete instant of time t1, .t2…..,input values I1,I2,….. each of which can take a finite no. of fixed values from the input alphabet which are applied to the O/P side of mode.
2. **Output:** o1, o2 which can take finite no's of fixed values from an o/p alphabet.
3. **States:** At any instant of time the automaton can be in one of the following states q1, q2 qn.
4. **State Relation:** The next state of an automaton at any instant of time is determined by the present state and the present input.
5. **Output Relation:** Output is related to either state only or to both the input and state.

An automata represented by 5 tuple (Q,∑, δ, q0, F) where

Q is a finite nonempty set of states

∑is a finite non empty set of states called input alphabet.

δ Is a transition function such that:

$$\delta: (Q * \sum) \rightarrow Q$$

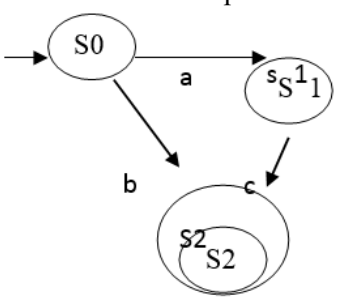
q0 ∈ Q as the initial state.

F is final state.

Where F ∈ Q is the set of final states.

→ Initial state is represented by a circle with arrow head.

→ Final state is represented by two concentric circles



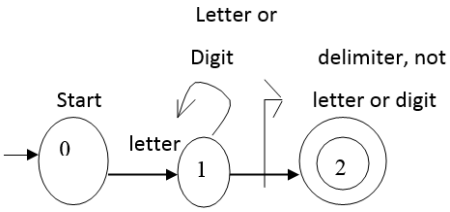
Q= {s0, s1, s2}
∑ = {a, b, c}
δ(s0, a)=s1
δ(s0, b)=s2
δ(s1, c)=s2
F= {s2} i.e. final states
Q0= {s0} i.e. initial state
We study here only 2 phases of compiler by automata theory:

First is **LEXICALANALYZER**
Now, I will take an example in C program. I will relate it with C compiler and then relate, different features of C program with automata.

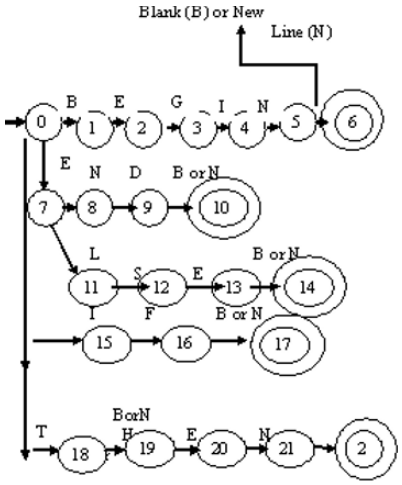
Suppose I have a program in C language .We use identifier, constants, operators, keywords in our program. Now the question arise how our compiler recognize that this particular string is identifier or keyword. The answer comes that it is up to the compiler.

But compiler solves the problem by automata concept in which we design a machine called automata. In this machine we take a string and try to evaluate that string. If we reach on the final state of M/c using that string, we can say that string is acceptable by automata else not acceptable by automata

Suppose a M/C for the identifier is:



If the given string match by this machine that the given string will be identifier else it can check for keywords. Similarly, we can draw machine for keywords, constant etc.
For Keyword is:



REGULAR EXPRESSION:

These are very useful notations. We can say that regular expressions are language for the set of inputs.

Example 01

100

1100

For all these input string we can write a common language i.e. (0+1)*

Similarly we can write language for 'identifier' is: Identifier=letter(letter| digit)*

For keywords:

Keyword=BEGIN|END|IF|THEN|EL

NOW (2ND PHASE) SYNTAX ANALYZER WITH AUTOMATA:

Here we study about context free grammar. CFG are useful for describing arithmetic expressions with arbitrary nesting of balanced parentheses and block structure inprogramming languages. A CFG is denoted by G= (V, T, P, S) where V and T are finite sets of variables and terminals. P is a finite set of productions is of form:

A-> x

Where A is a variable and x is a string of symbols from(V∪∑)*

And S is a special variable called the Start symbol.

Now a concept related to grammar is derivation tree. It imposes a structure on the words of a language that is useful in application such as the compilation of programming languages. The vertices of a derivation tree are labeled with terminal or variable symbols of the grammar. If an interior vertex n is labeled with A and the sons of n are labeled X1, X2…..Xk from the left then A->X1, X2, X3, Xk must be a production.

Example:

S->Aas|a

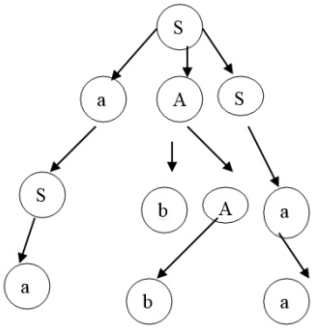
A-> SbA|ss|ba

We have to derive

S-> aabbaaa

Sol: S->aAS=>aSbAS=>aabAS=>aabbaS=>aabbaa

Its derivation tree is



So according to our language feature we can design leftmost and rightmost derivation tree. A derivation A=>w is called a leftmost derivation if we apply a production to the leftmost variable at every step.

A derivation A=>w is a rightmost derivation if we apply production to the rightmost variable at every step.

PROBLEM STATEMENT AND PROPOSED SOLUTION:

As we are seeing daily new applications are created and programmer are trying to find the ways to solve applications by software. Every software required a platform that fully satisfied the application. What the features we needed we map into the programs code. So for every new language we may required to develop a new compiler. So to map these new features we study automata theory which is a deep study of compiler making.

In this paper I took only two modules or phases of compiler and trying to relate it with program code. That is made up of different features like variables,identifier,keywords etc. and I tried to make it understand easily how our compiler understand all these features and give successfully compilation of program.

REFERENCES:

- *"Animation of the generation and computation of finite automat a for learning software" Beatrix Braune, Stephan Diehl, Andreas Keeren and Rein hard Wilhelm*
- *"The theory of definite automata" Perles, M.Robin, M.O. Sharmir*
- *"Tree Automata Techniques and Applications" Hubert Common, Max, Dauchet, Remi Gilleren*
- *"Tree Walking Pebble Automata" Joost, Engelfriet and Hendrik Jan Hoogeboom*
- *"The general and logical theory of automata" John Von Neumann*
- *"Learning compiler design as a research activity" Francisco Moreno Seco, Mikel L.Forcada*

Instructions for Authors

Essentials for Publishing in this Journal

1 Submitted articles should not have been previously published or be currently under consideration for publication elsewhere.

2 Conference papers may only be submitted if the paper has been completely re-written (taken to mean more than 50%) and the author has cleared any necessary permission with the copyright owner if it has been previously copyrighted.

3 All our articles are refereed through a double-blind process.

4 All authors must declare they have read and agreed to the content of the submitted article and must sign a declaration correspond to the originality of the article.

Submission Process

All articles for this journal must be submitted using our online submissions system. http://enrichedpub.com/ . Please use the Submit Your Article link in the Author Service area.

Manuscript Guidelines

The instructions to authors about the article preparation for publication in the Manuscripts are submitted online, through the e-Ur (Electronic editing) system, developed by **Enriched Publications Pvt. Ltd**. The article should contain the abstract with keywords, introduction, body, conclusion, references and the summary in English language (without heading and subheading enumeration). The article length should not exceed 16 pages of A4 paper format.

Title

The title should be informative. It is in both Journal's and author's best interest to use terms suitable. For indexing and word search. If there are no such terms in the title, the author is strongly advised to add a subtitle. The title should be given in English as well. The titles precede the abstract and the summary in an appropriate language.

Letterhead Title

The letterhead title is given at a top of each page for easier identification of article copies in an Electronic form in particular. It contains the author's surname and first name initial .article title, journal title and collation (year, volume, and issue, first and last page). The journal and article titles can be given in a shortened form.

Author's Name

Full name(s) of author(s) should be used. It is advisable to give the middle initial. Names are given in their original form.

Contact Details

The postal address or the e-mail address of the author (usually of the first one if there are more Authors) is given in the footnote at the bottom of the first page.

Type of Articles

Classification of articles is a duty of the editorial staff and is of special importance. Referees and the members of the editorial staff, or section editors, can propose a category, but the editor-in-chief has the sole responsibility for their classification. Journal articles are classified as follows:

Scientific articles:

1. Original scientific paper (giving the previously unpublished results of the author's own research based on management methods).
2. Survey paper (giving an original, detailed and critical view of a research problem or an area to which the author has made a contribution visible through his self-citation);
3. Short or preliminary communication (original management paper of full format but of a smaller extent or of a preliminary character);
4. Scientific critique or forum (discussion on a particular scientific topic, based exclusively on management argumentation) and commentaries. Exceptionally, in particular areas, a scientific paper in the Journal can be in a form of a monograph or a critical edition of scientific data (historical, archival, lexicographic, bibliographic, data survey, etc.) which were unknown or hardly accessible for scientific research.

Professional articles:

1. Professional paper (contribution offering experience useful for improvement of professional practice but not necessarily based on scientific methods);
2. Informative contribution (editorial, commentary, etc.);
3. Review (of a book, software, case study, scientific event, etc.)

Language

The article should be in English. The grammar and style of the article should be of good quality. The systematized text should be without abbreviations (except standard ones). All measurements must be in SI units. The sequence of formulae is denoted in Arabic numerals in parentheses on the right-hand side.

Abstract and Summary

An abstract is a concise informative presentation of the article content for fast and accurate Evaluation of its relevance. It is both in the Editorial Office's and the author's best interest for an abstract to contain terms often used for indexing and article search. The abstract describes the purpose of the study and the methods, outlines the findings and state the conclusions. A 100- to 250-Word abstract should be placed between the title and the keywords with the body text to follow. Besides an abstract are advised to have a summary in English, at the end of the article, after the Reference list. The summary should be structured and long up to 1/10 of the article length (it is more extensive than the abstract).

Keywords

Keywords are terms or phrases showing adequately the article content for indexing and search purposes. They should be allocated heaving in mind widely accepted international sources (index, dictionary or thesaurus), such as the Web of Science keyword list for science in general. The higher their usage frequency is the better. Up to 10 keywords immediately follow the abstract and the summary, in respective languages.

Acknowledgements

The name and the number of the project or programmed within which the article was realized is given in a separate note at the bottom of the first page together with the name of the institution which financially supported the project or programmed.

Tables and Illustrations

All the captions should be in the original language as well as in English, together with the texts in illustrations if possible. Tables are typed in the same style as the text and are denoted by numerals at the top. Photographs and drawings, placed appropriately in the text, should be clear, precise and suitable for reproduction. Drawings should be created in Word or Corel.

Citation in the Text

Citation in the text must be uniform. When citing references in the text, use the reference number set in square brackets from the Reference list at the end of the article.

Footnotes

Footnotes are given at the bottom of the page with the text they refer to. They can contain less relevant details, additional explanations or used sources (e.g. scientific material, manuals). They cannot replace the cited literature.
The article should be accompanied with a cover letter with the information about the author(s): surname, middle initial, first name, and citizen personal number, rank, title, e-mail address, and affiliation address, home address including municipality, phone number in the office and at home (or a mobile phone number). The cover letter should state the type of the article and tell which illustrations are original and which are not.

Notes: