A SOA-based Approach for Personalizing Learning

Razieh Niazi & Nick Cercone

Soft Computing Project

April 2010

I. Introduction

Michio Kaku in 1998 elaborated how science will revolutionize the planet in the future. In his vision, he talked about changing everything around us including the nature of commerce, the wealth of nations, work, play and the way we communicate[1].Indeed, we will have an "intelligent planet" in which we can communicate with objects.

In the last decade, we have observed many changes in technologies. We have seen technological advances in people communication and business services. In Web 2.0, we have endeavored several transformations from electronic services like e-commerce, e-banking, e-shopping to mobile services like m-banking and m-learning. We are also introduced to social webs. In the next generation of Web, Web 3.0, we have wisdom and intelligence. Indeed, instead of data, we have knowledge, inference engines and consequence actions.

One of the emerging trends in the next generation of the Web is personalization and integration. Personalization is ability to adapt based on users' need, and desire. Personalization can be applied in services, search queries, user profiles, learning materials and so on. Integration is ability to adapt to different environments. A good example of integration is web service which can be integrated into any platform and is language independent.

Personalization in learning is an emerging trend in education bringing attention of many researchers. A personalized learning model is built based on the learners' needs, ability and background and it is different from individualized learning. E-learning systems which are widely used today are sort of individualized learning in this sense that they are available anytime, and anywhere. Hence, students can learn the materials with their own speed. However, they are suffering from the lack of personalization, knowledge neighborhoods and knowledge discovery.

There are several issues with the current approaches in learning systems:

1) Knowledge is not limited to the learning materials provided by instructors. In the current learning systems, knowledge is presented as a series of lectures in various formats. However, there are huge

amount of information available on the Web, libraries, social networks, and knowledge repositories. The question is how they can be referenced and discovered?

2) Knowledge includes not only factual but also heuristic information which is more experiential. There are huge amount of heuristic knowledge is available on the Web like discussion panels, social networks and forums. We need a way to reference these sorts of knowledge.

3) Regarding the availability and accessibility of huge amount of information on the Web, the bottleneck is knowledge discovery. The current state of knowledge discovery is elaborated most by search engines which work based on indexing.

4) Personalized learning is about having personalized lesson based on learners need, ability and background. In order to build customized lesson, we need to aggregate learning objects. However, the granularity level of learning object is too coarse.

5) There are many definitions on Learning Objects. The most popular definition defines a learning objects as small units of learning resources which are self-contained, reusable and able to aggregate and combined [2, 3, 4]. We need to design reusable learning object which can be easily aggregated to make a personalized lesson. Reusability of learning object means "placing a learning object in a context other than that for which it was designed" [11].

To summarize issues mentioned above, personalized learning process bottlenecks are identified in 4 categories: knowledge directories, knowledge discovery, knowledge neighborhood and the granularity level of learning objects.

Knowledge discovery as the basis for personalized learning is a difficult task due to the following reasons:

1. Information overload: Most problems which we have are finding a path through the huge amount of information currently available on the Web. Evaluating this information needs skills [1].
2. Information authentication: When we talk about the reusability of learning object, the question we face is whether we are allowed to reuse materials designed by authors. We need to find a way of authenticating learning objects.
3. Given the amount of information available, the problem of matching learners to materials relevant to his or her needs, becomes more and more required [1].

The term "Knowledge neighborhoods" is defined as "locations on the Web where communities collaborate to create and use representations of their knowledge" [1]. We need a way to connect knowledge neighborhoods together.

We need Knowledge directories to locate knowledge neighborhoods. We need a representation model to present knowledge directories.

In this paper, we initiate a solution for personalized learning and building a customize lesson. Indeed, we address all the issues mentioned earlier. Our approach is Service-Oriented Architecture (SOA) based. We borrow the concepts defined in SOA and introduce new concepts. We also define a granularity level for learning object enabling them to aggregate and combine. To perform aggregation, we use a rough-based approach which is based on indiscernability of learning object. We also provide a proof of concept implementation.

The rest of the paper is organized as follows: In Section II, an overview of Service-Oriented Architecture (SOA) is presented. Related works are represented in Section III. We also describe how our work is different from others. In Section IV, we initiate our solution for personalized learning, knowledge discovery, learning object granularity, and aggregation. In Section V, a proof of concept implementation is described. Finally, in Section VI, we have conclusion.

II. Service Oriented Architecture

Service oriented Architecture is an emerging paradigm for distributed computing in which the design, implementation and delivery of software is changed. SOA has three main components: a provider, a consumer and a registry. Providers publish their services on registries where consumers discover and invoke them (See Diagram 1).

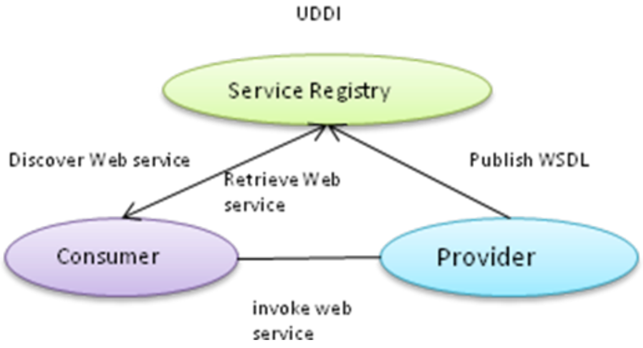


Diagram 1: Web service Architecture based on SOA

Web service architecture model is a basis for SOA architecture [10]. In this model, services are considered as reusable methods or a combination of methods which function together to provide a service. Services are available on the Web and are published in a registry named Universal Description Discovery and Integration (UDDI). A consumer discovers services on UDDI and retrieves Web service Description Language (WSDL) from UDDI. A WSDL is an XML file in which a collection of endpoints and operations are described. Operations are service methods. Using WSDL, a consumer is able to invoke a web service remotely.

Service-Oriented computing is well adapted and wildly used in industry. Many companies like Amazon, ebay, Google and Yahoo expose their web services and are consumed by third parties.

III. Related Work

We divide the related work into two sections: Learning Object and Personalized Learning

A. Learning Object

In the current state of learning systems, a Learning Object (LO) is a document, a photograph, a video clip, a java applet or any other object used for online learning [6]. LO are required to be found and discovered by learners and referenced by authors. In order to search LOs, descriptions about their characteristics are required. These characteristics are represented as metadata. There are various standards. In this section, we present the important ones:

- SCORM: SCORM stands for Sharable Content Object Reference Model, initiated by Advanced Distributed Learning (ADL) specification group. The current design of SCORM has resulted in [10]: (1) the slow pace of designing learning objects; (2) high cost developing of learning objects; and (3) not able to be tailored to individual needs

- LOM: LOM stands for IEEE Learning Object Metadata. It is a data model encoded in XML and used to describe learning objects. It was developed by IEEE supports reusability of learning objects, aids discoverability and facilitates interoperability in the context of online learning management systems

The major issue with the current state of LOs is the granularity level. The current granularity level is a document or a lecture in a different format such as word, PDF, or PowerPoint . The granularity level is too coarse. It does not allow reusability and aggregation of learning objects and build a personalized lesson.

A concept is described in two dimensions including intention and extention [5]. Intention is about a set of concept' attribute and values. In other words, it specifies the inter-relations between elements of a concept. In contrast, extention is about a set of objects that belongs to a concept. Indeed, it identifies the intra-relations between different objects. The metadata designed currently support extention of an object. It provides data about characteristics of a learning object, however, we still have problem with the granularity of the learning object.

B. Personalized Learning

Most researches done in personalized learning consider pedagogical aspects of personalization. A few research focus on practical aspects of personalized learning [13], though they mostly focus on building students models. Our approach is totally different from the related work, since we are looking at the problem from different point of view. Our approach is not only about students but also about knowledge discovery on the Web in general, granularity of learning objects and bringing semantic in user enquiries. Consider, for instance, a user named John is learning about a new topic such as Fuzzy Logic. He read learning materials, however, he still has problems in understanding the membership functions and how they are employed in control systems. He read tutorials and searches the Web with the certain keywords, the material he gets almost point to a certain things, similar together unless he does many queries with different keywords in different resources to see if the materials are useful. Now, consider knowledge discovery as a service. We send our request in a complete sentence not keywords. We say what we want specifically and the service on the Web gives us exactly the materials in the certain point.

IV. Initiate Solution

We initiate a SOA-based approach for personalizing learning. In this approach, we borrow the concept in Web Service architecture model and introduce new concept. Our approach is summarized in several tasks:

- Building granular learning objects
- Publish LO
- Discovery of LO
- Building customized lesson

In the following sections, we will describe each task in detail:

A. Building granular learning objects

The success key in having customized lesson is having an appropriate granularity level. We consider writing as a problem solving task. Each paragraph is expressed by a single idea containing several sentences. A point-of-view can be described by joining several ideas in several paragraphs. Initially, we consider a paragraph as a granule in our process. However, it can be changed based on the author's annotation. Hence, the granularization has two states:

- Initiate state: In this state, we consider a paragraph as a granular.
- Annotation state: authors annotate text based on their writing style. Annotation is done in the following scheme:

  @ principal(TopicName) Schema (1)

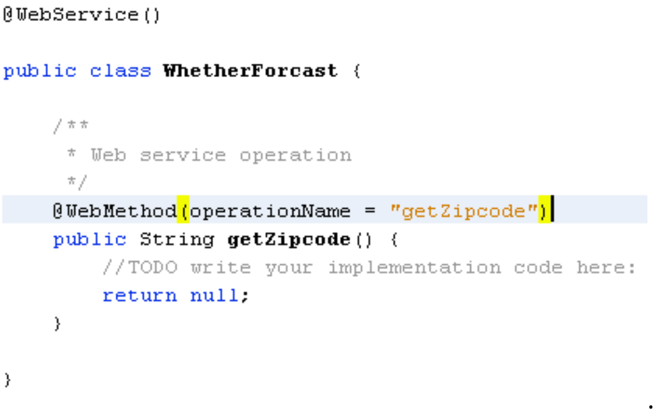
  TopicName is a term describing the basic idea in the paragraph (See Example 1).

  Principal describes whether the paragraph is about a concept, a usage, an advantage, a disadvantage and so on. Overall, it identifies what aspect of the term is discussed.

  Example 1: *Fuzzy logic is used in system control and analysis design, because it shortens the time for engineering development and sometimes, in the case of highly complex systems, is the only way to solve the problem.*

  Example 1 shows a paragraph describing the usage of Fuzzy Logic. If we want to specify the basic idea of the paragraph in one term, we say, Fuzzy Logic. "TopicName" is "Fuzzy Logic" and “principal” is “Usage”. Hence, based on Schema (1), we can annotate the paragraph as “@Usage(Fuzzy Logic)” .

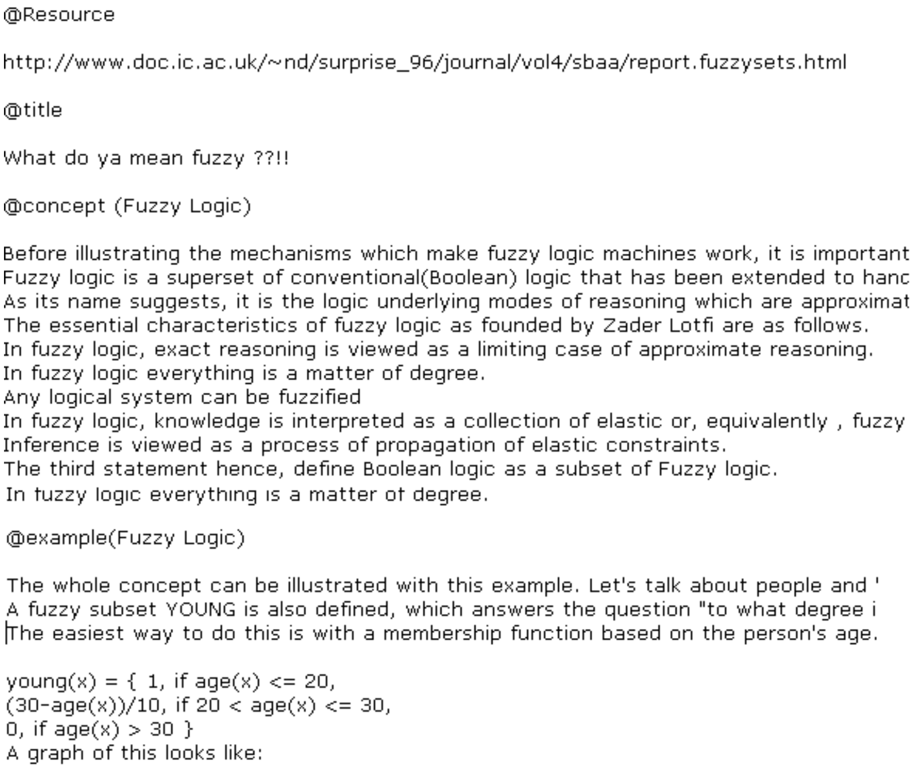
Annotating text by authors is neither a difficult task nor a burden, since they should provide two parameters: Topic Name and Principal. By designing tools, this task can be done as easily as possible. We use annotation in software programming a lot. Figure 2 shows a sample example of annotation task supported by NetBeans IDE.



```
@WebService()

public class WhetherForcast {

    /**
     * Web service operation
     */
    @WebMethod(operationName = "getZipcode")
    public String getZipcode() {
        //TODO write your implementation code here:
        return null;
    }

}
```

Figure 2. An annotated web service in NetBeans

We annotated some text based on Schema (1) manually. Figure 3 demonstrates an example:



@Resource

http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.fuzzysets.html

@title

What do ya mean fuzzy ??!!

@concept (Fuzzy Logic)

Before illustrating the mechanisms which make fuzzy logic machines work, it is important
Fuzzy logic is a superset of conventional(Boolean) logic that has been extended to hanc
As its name suggests, it is the logic underlying modes of reasoning which are approximat
The essential characteristics of fuzzy logic as founded by Zader Lotfi are as follows.
In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.
In fuzzy logic everything is a matter of degree.
Any logical system can be fuzzified
In fuzzy logic, knowledge is interpreted as a collection of elastic or, equivalently , fuzzy
Inference is viewed as a process of propagation of elastic constraints.
The third statement hence, define Boolean logic as a subset of Fuzzy logic.
In tuzzy logic everything is a matter ot degree.

@example(Fuzzy Logic)

The whole concept can be illustrated with this example. Let's talk about people and '
A fuzzy subset YOUNG is also defined, which answers the question "to what degree i
The easiest way to do this is with a membership function based on the person's age.

young(x) = { 1, if age(x) <= 20,
(30-age(x))/10, if 20 < age(x) <= 30,
0, if age(x) > 30 }
A graph of this looks like:

Figure 3. Annotated text done manually based on Schema (1)

In addition to Schema (1), there are other annotation tags as follows:

- @Resource: it specifies the original source of a learning content (URL) developed by an author.
- @title: it specifies the titles in a text
- @image: it specifies the address of an image in a text

Several tasks are performed on annotated text. Diagram 2 shows the processing steps:

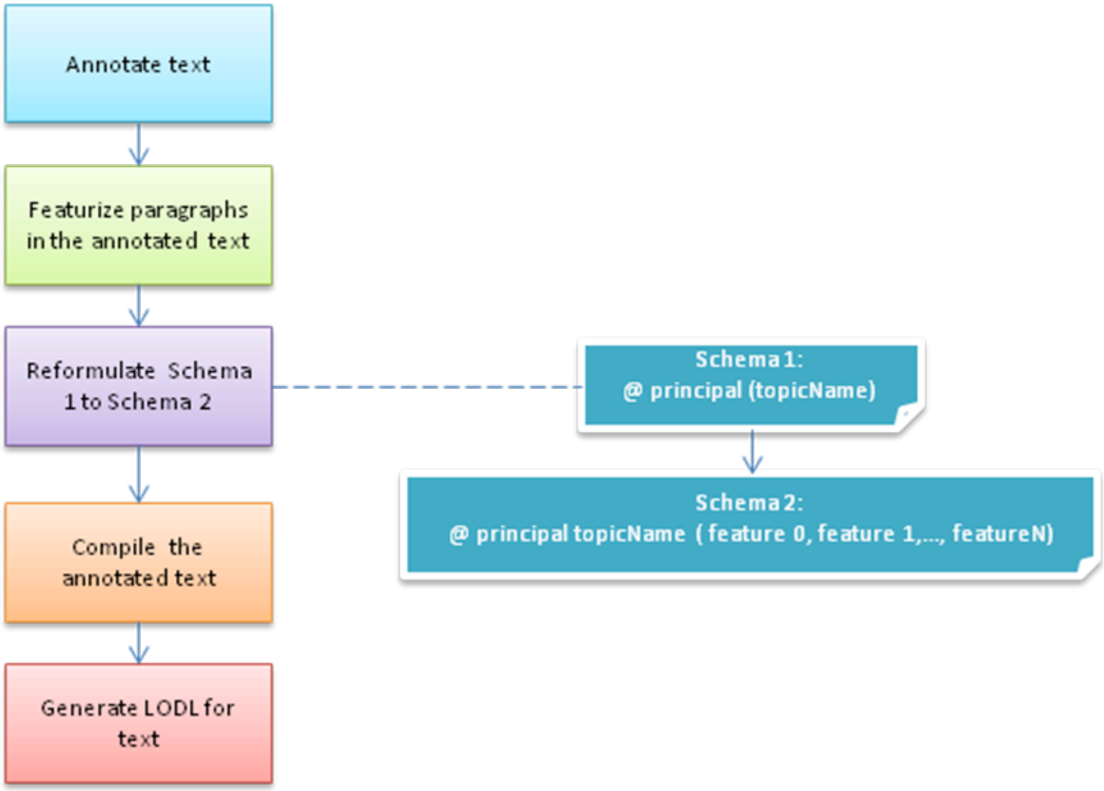

Annotate text
Featurize paragraphs in the annotated text
Reformulate Schema 1 to Schema 2
Compile the annotated text
Generate LODL for text
Schema 1:
@ principal (topicName)
Schema 2:
@ principal topicName (feature 0, feature 1,..., featureN)

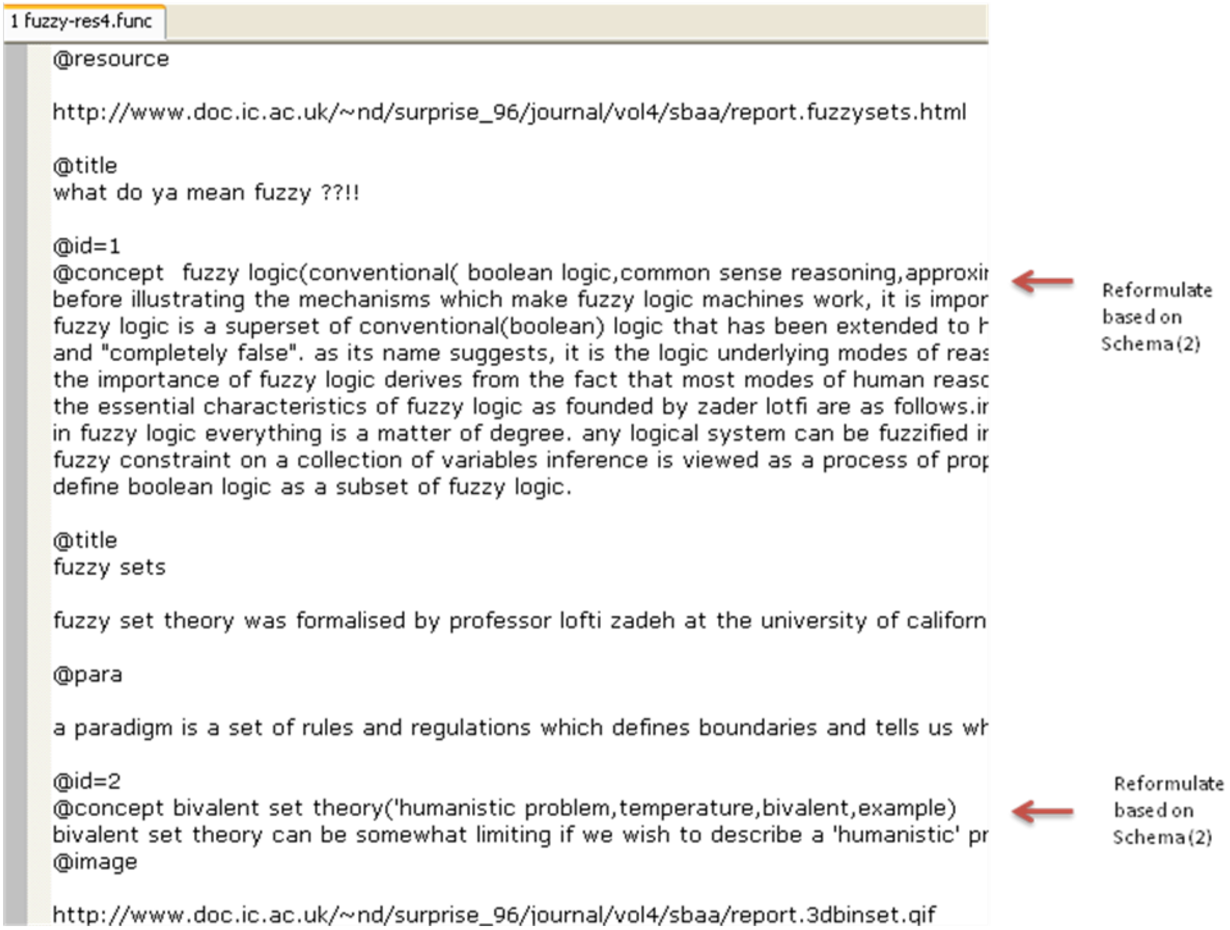
Diagram 2: processing tasks for building LO granules

As demonstrated in Diagram 2, annotated text are featurized. The featurization algorithm is described in Section B. Then, the annotated text is compiled by our program and a file with the extension ".func" is generated . This file presents the functional representation of the paragraphs based on Schema 2.

In the compiling process, the annotate text represented by Schema (1) converts to a functional model built based on Schema 2. The following shows the schema model:

@ Principal TopicName(feature0, feature 1, feature2,….,featureN) Schema (2)

In this way, each paragraph is characterized by its topic, its principal and a series of features. Features in a context (i.e: paragraph) are terms either unigram or collocations. Figure 4 shows the compiled file generated by our application:



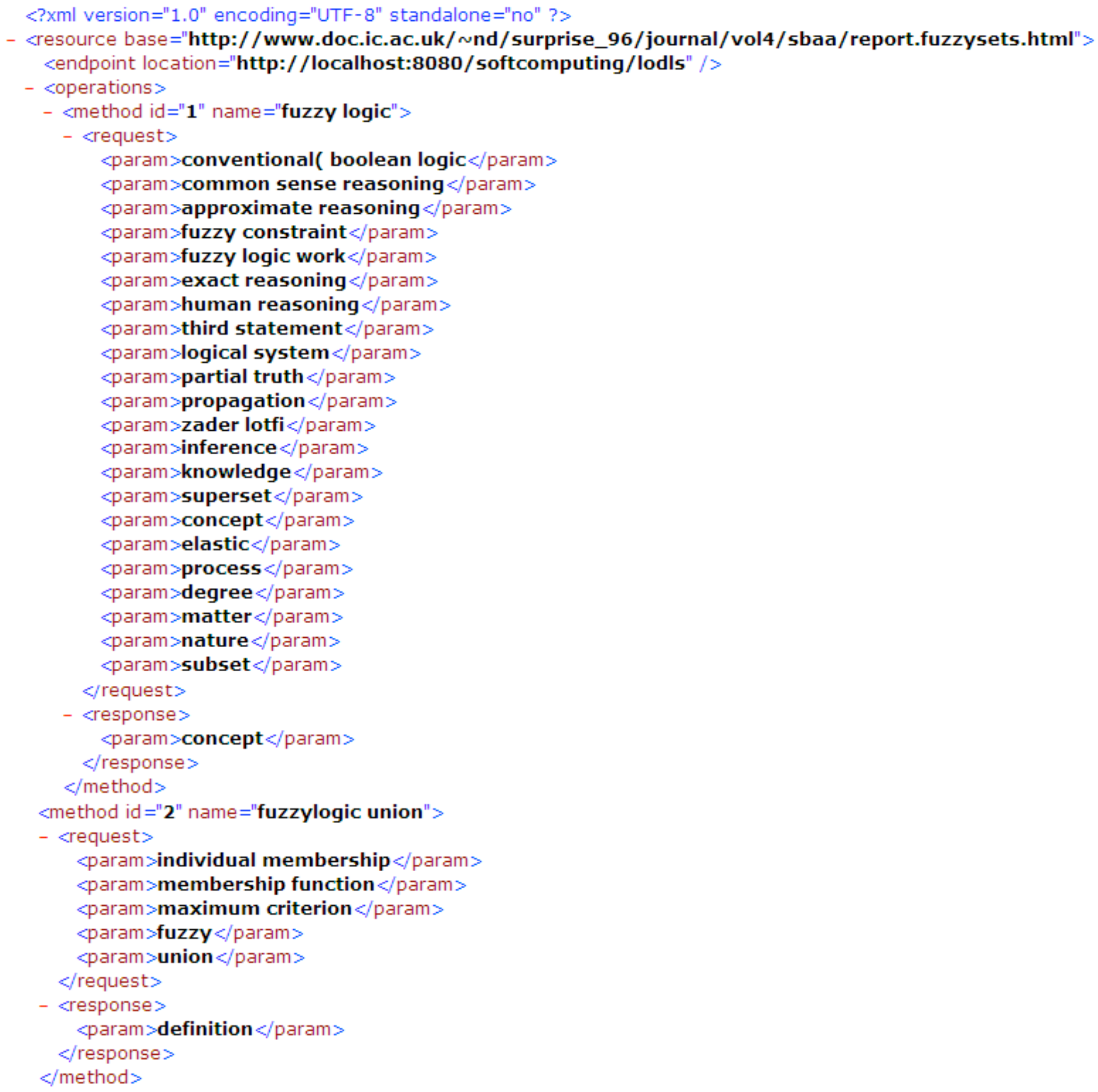
```
@resource

http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.fuzzysets.html

@title
what do ya mean fuzzy ??!!

@id=1
@concept  fuzzy logic(conventional( boolean logic,common sense reasoning,approxir
before illustrating the mechanisms which make fuzzy logic machines work, it is impor
fuzzy logic is a superset of conventional(boolean) logic that has been extended to h
and "completely false". as its name suggests, it is the logic underlying modes of reas
the importance of fuzzy logic derives from the fact that most modes of human reasc
the essential characteristics of fuzzy logic as founded by zader lotfi are as follows.ir
in fuzzy logic everything is a matter of degree. any logical system can be fuzzified ir
fuzzy constraint on a collection of variables inference is viewed as a process of prop
define boolean logic as a subset of fuzzy logic.

@title
fuzzy sets

fuzzy set theory was formalised by professor lofti zadeh at the university of californ

@para

a paradigm is a set of rules and regulations which defines boundaries and tells us wh

@id=2
@concept bivalent set theory('humanistic problem,temperature,bivalent,example)
bivalent set theory can be somewhat limiting if we wish to describe a 'humanistic' pr
@image

http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.3dbinset.gif
```

Reformulate based on Schema(2)

Reformulate based on Schema(2)

Figure 4: Compiled file; fussy-res4.func

In order to be able to discover granules in a text, we propose a description language named Learning Object Description Language (LODL). LODL is an xml file and similar to the WSDL in the Web service architecture. A brief description about WSDL presents in Section II. LODL is generated for each text (learning content) during the compilation process. LODL is the functional representation of paragraphs (annotated text) in a learning material. Figure 5 shows a sample LODL generated by our program:



```
  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <resource base="http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.fuzzysets.html">
    <endpoint location="http://localhost:8080/softcomputing/lodls" />
  - <operations>
    - <method id="1" name="fuzzy logic">
      - <request>
          <param>conventional( boolean logic</param>
          <param>common sense reasoning</param>
          <param>approximate reasoning</param>
          <param>fuzzy constraint</param>
          <param>fuzzy logic work</param>
          <param>exact reasoning</param>
          <param>human reasoning</param>
          <param>third statement</param>
          <param>logical system</param>
          <param>partial truth</param>
          <param>propagation</param>
          <param>zader lotfi</param>
          <param>inference</param>
          <param>knowledge</param>
          <param>superset</param>
          <param>concept</param>
          <param>elastic</param>
          <param>process</param>
          <param>degree</param>
          <param>matter</param>
          <param>nature</param>
          <param>subset</param>
        </request>
      - <response>
          <param>concept</param>
        </response>
      </method>
    <method id="2" name="fuzzylogic union">
    - <request>
        <param>individual membership</param>
        <param>membership function</param>
        <param>maximum criterion</param>
        <param>fuzzy</param>
        <param>union</param>
      </request>
    - <response>
        <param>definition</param>
      </response>
    </method>
```

Figure 5: A sample LODL file generated by our program

B. *Featurizing Paragraphs*

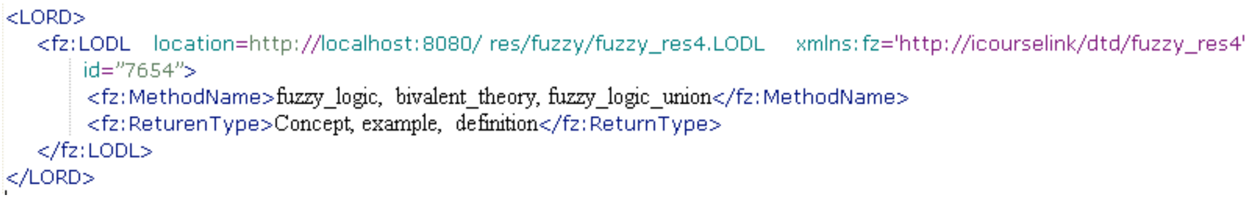
Diagram 3 shows the proposed solution to extract features. As demonstrated in Diagram 3, to extract features, given a free style text, sentences are detected and words are tokenized. We perform POS Tagging on tokens. The most common POS tags in English in the Word-level are: noun, verb, adjective, adverb, pronoun, preposition, conjunction and interjection. For the feature extraction, we are only interested in nouns and proper nouns in a noun phrase.

In order to obtain semantic orientation of words and find out how words are inter-related together, we employ chunkers. For the feature extraction, we use the same algorithm initiated in [12].

C. Publishing LO

Since our approach is service-oriented based, to have a reference to learning objects, we need to publish them to a registry. We borrow the concept of Universal Description Discovery and Integration (UDDI) and introduce a new concept named Learning Object Repository and Directory (LORD). LORD is an xml-based registry in which we store LODLs. In this way, we support reusability of LOs and provide a solution to reference LOs described in LODLs. Table 1 shows the tag elements in a LORD:

Table 1: A sample example of LORD with one entry, generated by our program



```
<LORD>
   <fz:LODL   location=http://localhost:8080/ res/fuzzy/fuzzy_res4.LODL    xmlns:fz='http://icourselink/dtd/fuzzy_res4'
        id="7654">
         <fz:MethodName>fuzzy_logic,  bivalent_theory, fuzzy_logic_union</fz:MethodName>
         <fz:ReturenType>Concept, example,  definition</fz:ReturnType>
   </fz:LODL>
</LORD>
```

D. Discovery of LO & Build a customize lesson

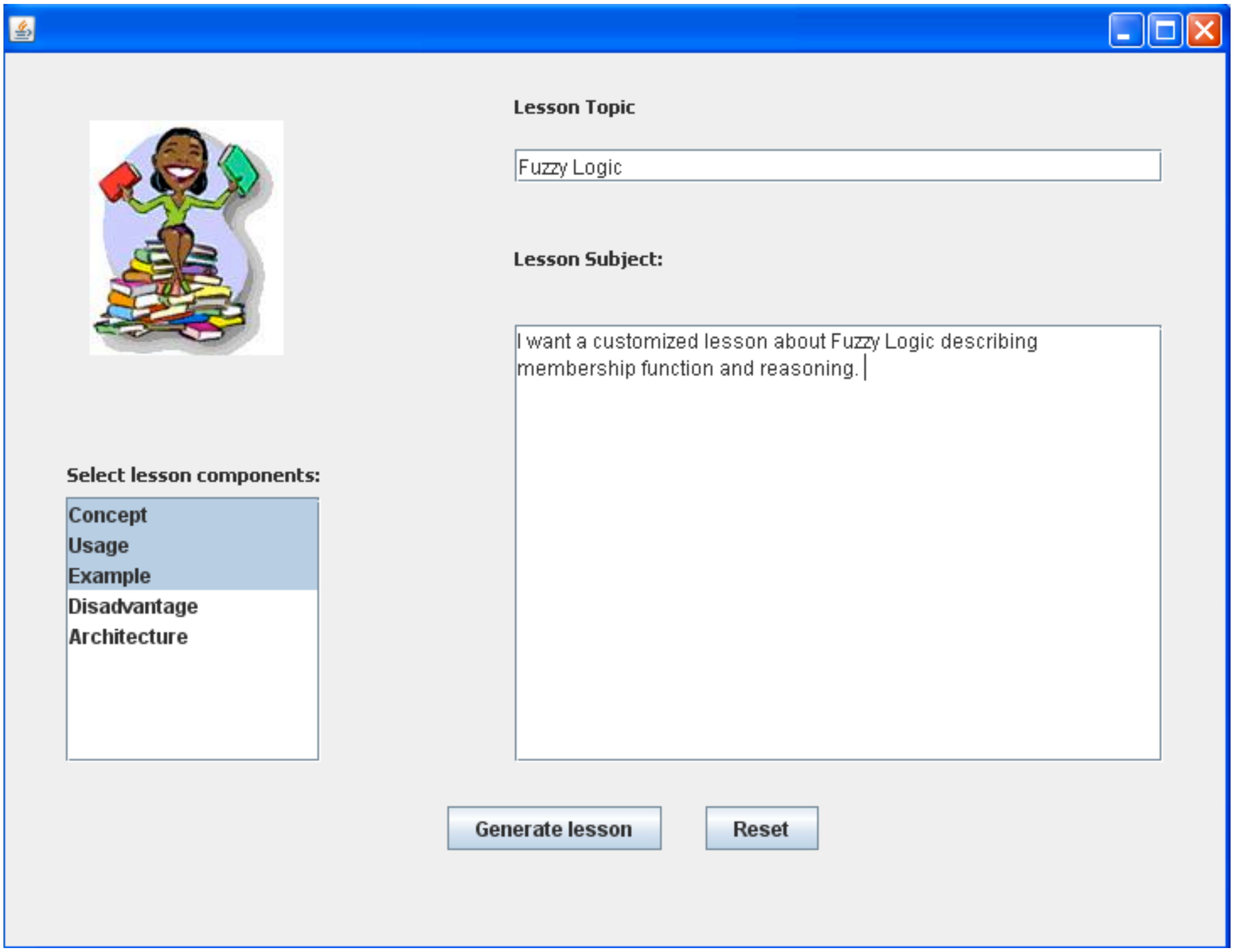
Discovery of LO is done by searching in LORD and finding relevant LODLs with respect to the learners learning path. Initially, a learning path is a set of parameters identified by a learner. In a learning path, basically it is asked about the following elements:

- Principals: principals are various parts in a customized lesson that a learner wants to have. We already discussed about principals in Section A.

- Lesson topic: this specifies a topic for the customized lesson and is about a term or concept on which a learner wants to have a customized lesson. For instance, we want a lesson about Fuzzy Logic. Fuzzy Logic is the topic.

- Lesson subject: In this element, the learner specifies on what aspects of a concept, the customized lesson should be, say " we want to have a lesson about the membership function and reasoning in fuzzy logic".

In Section E, we discuss about the searching strategy. Figure 6 shows a screenshot of the User Interface designed for the learner's learning path.



Lesson Topic

Fuzzy Logic

Lesson Subject:

I want a customized lesson about Fuzzy Logic describing membership function and reasoning.

Select lesson components:

- Concept
- Usage
- Example
- Disadvantage
- Architecture

Figure 6: A screenshot of the designed UI for the learner's learning path

E. The search method strategy

Many authors and companies develop learning materials. Since all learning materials are decomposited to granules, initially a paragraph, we have so many LOs in LORD. How can we find the most related LOs with respect to the learner's learning path? We use Rough set theory to partition similar objects since rough set is a successful methodology based on the discernability of objects.

Diagram 4 shows the discovery process of learning objects to build a customized lesson.

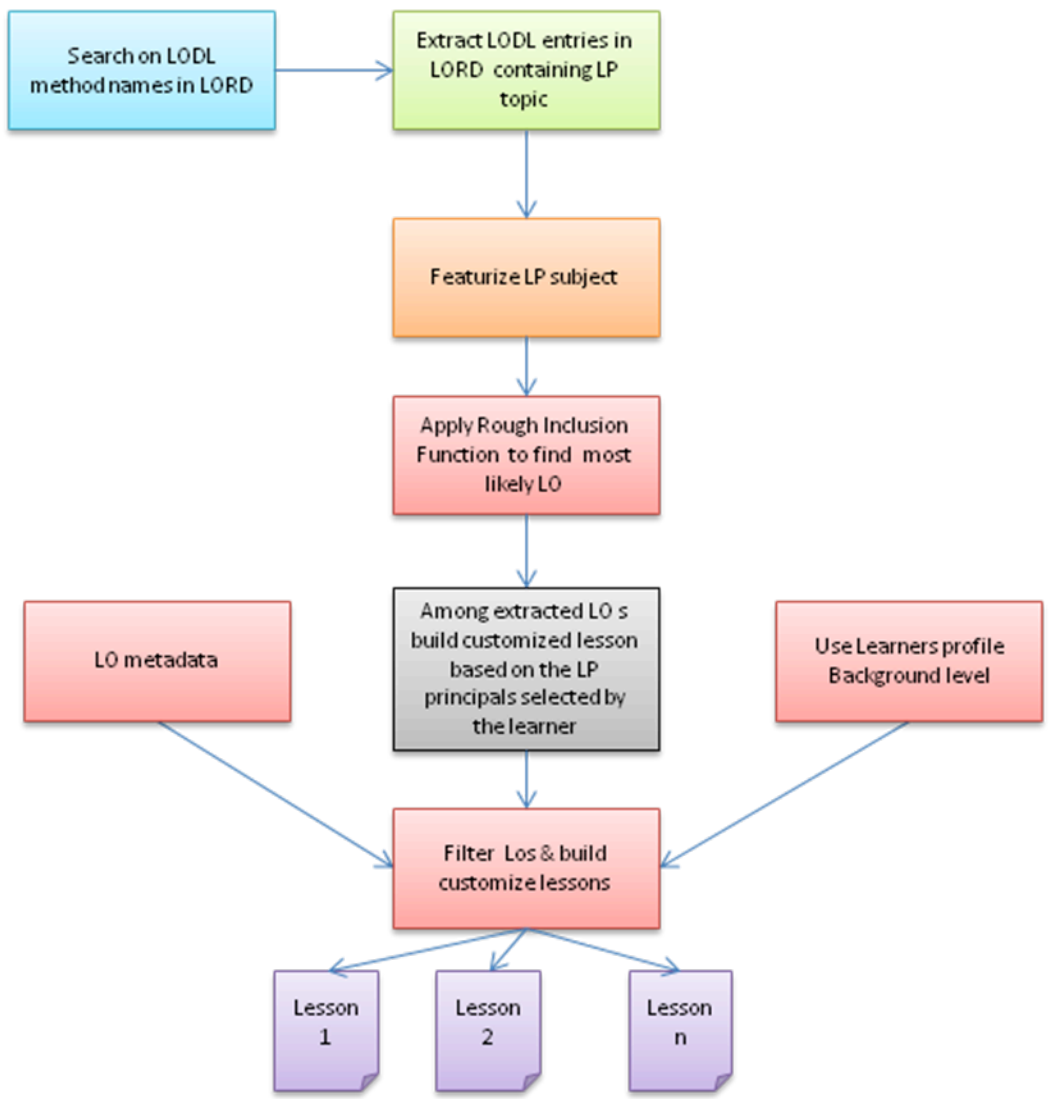

Search on LODL method names in LORD
Extract LODL entries in LORD containing LP topic
Featurize LP subject
Apply Rough Inclusion Function to find most likely LO
LO metadata
Among extracted LO s build customized lesson based on the LP principals selected by the learner
Use Learners profile Background level
Filter Los & build customize lessons
Lesson 1
Lesson 2
Lesson n

Diagram 4: Discovery process of learning objects to build a customized lesson

The search strategy has two steps:

- Search LORD: Finding LODLs entries in LORD which “MethodName” element matches learner’s learning path topic (See Table 1). As the result a list of LODL is extracted.
- Search extracted LODLs: In this step, the learner’s learning path lesson subject is featurized and features are compared with method elements (param tags) in LODL (See Figure 5). Given two sorts of features, Rough Inclusion Function is computed.

Rough Inclusion Function [7] defines the degree of inclusion of X in Y where X,Y $\subseteq$ U. In the simplest case, the standard rough inclusion function can be defined as follow:

$$V_{SRI}(X,Y) = \begin{cases} \frac{card(X \cap Y)}{card(X)} & \text{if } X \neq \varnothing \\ 1 & X = \varnothing \end{cases}$$

Formula (1)

The following example illustrates Formula (1):

Table 2: An example to illustrate Formula (1)

LODL id in LORD	Learning Object ID in LODL	X= user's learning path lesson subject	Y= param tags in LODLs	V_{SRI} (X, Y)
1	1	{membership function, partial truth, reasoning}	{boolean logic, common sense reasoning, approximate reasoning, fuzzy constraint, exact reasoning, human reasoning, logical system, partial truth, propagation, inference, knowledge, superset}	2/3 = 0.67
2	7	{membership function, partial truth, reasoning}	{abstraction originating, knowledge experience, classical logic, precise numeric, exact numeric, subjective, complex system}	0
3	4	{membership function, partial truth, reasoning}	continuous variable, temperature measurement, anti-lock brakes, membership functions, truth value,	1/3 = 0.34

As demonstrated in Table 2, the learning object with ids (LODL_id =1, LO_method_id = 1) has the highest V_{SRI} (X, Y) value and is selected as the most relevant learning object with respect to the learner's learning path.

V. Implementation

The program has been fully developed in java, J2EE. We use various techniques in developing the application as follows:

- J2EE: to develop the application

- JAXP: to parse XMLs. Also, we use Document Object Model (DOM) parser to write and read XMLs.

- OpenNLP [8]: this is a third party API which is used for featurization process in our program. OpenNLP is an organizational center for open source projects related to natural language processing. It provides Java-based libraries for POS tagging, chunking and more.

- Java UI-Swing: it is used to design user interfaces.

A high level architecture of the project is presented in Figure 7:

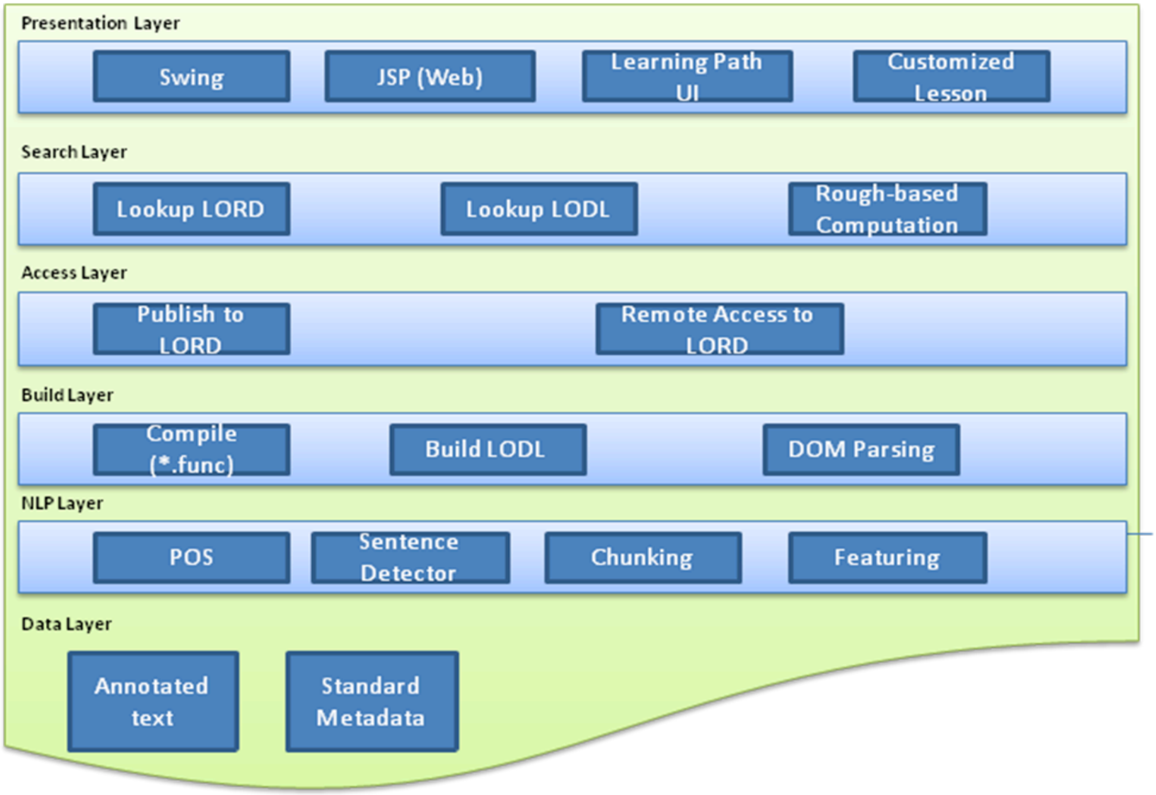


Figure 7: A high level architecture of our system

VI. Conclusion

In this paper, we initiated a SOA- based approach for personalizing learning. In this approach, learning objects can be discovered, referenced and because the granularity level is fine, they can be aggregated and combined to build a customized lesson. We applied a rough-based approach in discovery process to find most relevant learning objects to the learner's learning path. This method works based on discernability of objects and gives a higher score for having an inclusion in a set of features.

The initiate idea of the work came from Wiki where we have different elements from different resources combined together to represent a concept.

References:

[1] Stutt, A. and Motta, E. 2004. Semantic Learning Webs, Journal of Interactive Media in Education (JIME), 2004(10), pp. 253-285

[2] Learning Object Metadata, http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf (Accessed: April 4. 2011)

[3] Wiley, D. A. 2003. Learning objects: Difficulties and opportunities. Academic ADL Co-Lab News Report: No. 152-030406., http://opencontent.org/docs/lo_do.pdf (Accessed: April 10, 2011)

[4] Wiley, D. A. 2001. Instructional Use of Learning Objects*,* Agency for Instructional Technology, http://www.reusability.org/read/ (Accessed: April9, 2011)

[5] Sonamthiang, S. **,** Cercone, N., & Naruedomkul, K. 2010. Granular concept map. Transaction on Rough Sets (in press).

[6] McGreal, R. 2004. Learning objects: A practical definition. International Journal of Instructional Technology and Distance Learning (IJITDL), 9(1), http://www.itdl.org/journal/sep_04/article02.htm, (Accessed: April 8, 2011)

[7] Stepaniuk, J. 2008. Rough Granular Computing in Knowledge Discovery and Data Mining (1 ed.). Springer Publishing Company, Incorporated.

[8] OpenNLP, http://incubator.apache.org/opennlp/ (Accessed: December 15, 2010)

[9] Papazoglou, M. P. 2003. Service-Oriented Computing: Concepts, characteristics and directions. In WISE ’03: Proceedings of the Fourth International Conference on Web Information Systems Engineering, Washington, DC, USA, IEEE Computer Society, pp. 3.

[10]Romiszowski, R. 2007. Instructional Design, Learning Objects and SCORM revisited, TTS-Global Knowledge Factory, http://tts-global.com/blog/2007/07/01/instructional-design-learning-objects-and-scorm-revisited/ (Accessed: April 15, 2010)

[11]Wiley, D. 2004. The Reusability Paradox. Connexions. http://cnx.org/content/m11898/1.18/ (Accessed: April 15, 2010)

[12]Niazi, R., Cercone, N., and An, A. (2011) *A Computational Linguistics Approach to Feature-based Sentiment Analysis,* Submitted to the 2011 IEEE/WIC/ACM International Conference on Web Intelligence, to be held August 22 - 27, 2011, Lyon, France.

[13]Madhour, H., Wentland, M., and Fernandes, E. 2006. Semantic learning model and extended student model: towards an AHAM-based adaptive system. In Proceedings of the 5th IASTED international conference on Web-based education (WBE'06), V. Uskov (Ed.). ACTA Press, Anaheim, CA, USA, pp. 349-354.

APPENDIX

A. Project System Manual

The application has 3 main menu: File, Build and Configuration.

File Menu:

- Open New File: By clicking on this menu item, you are able to open an annotated text .
- “Save as…”: By clicking on this menu item, you can save your file with the same name or different name.

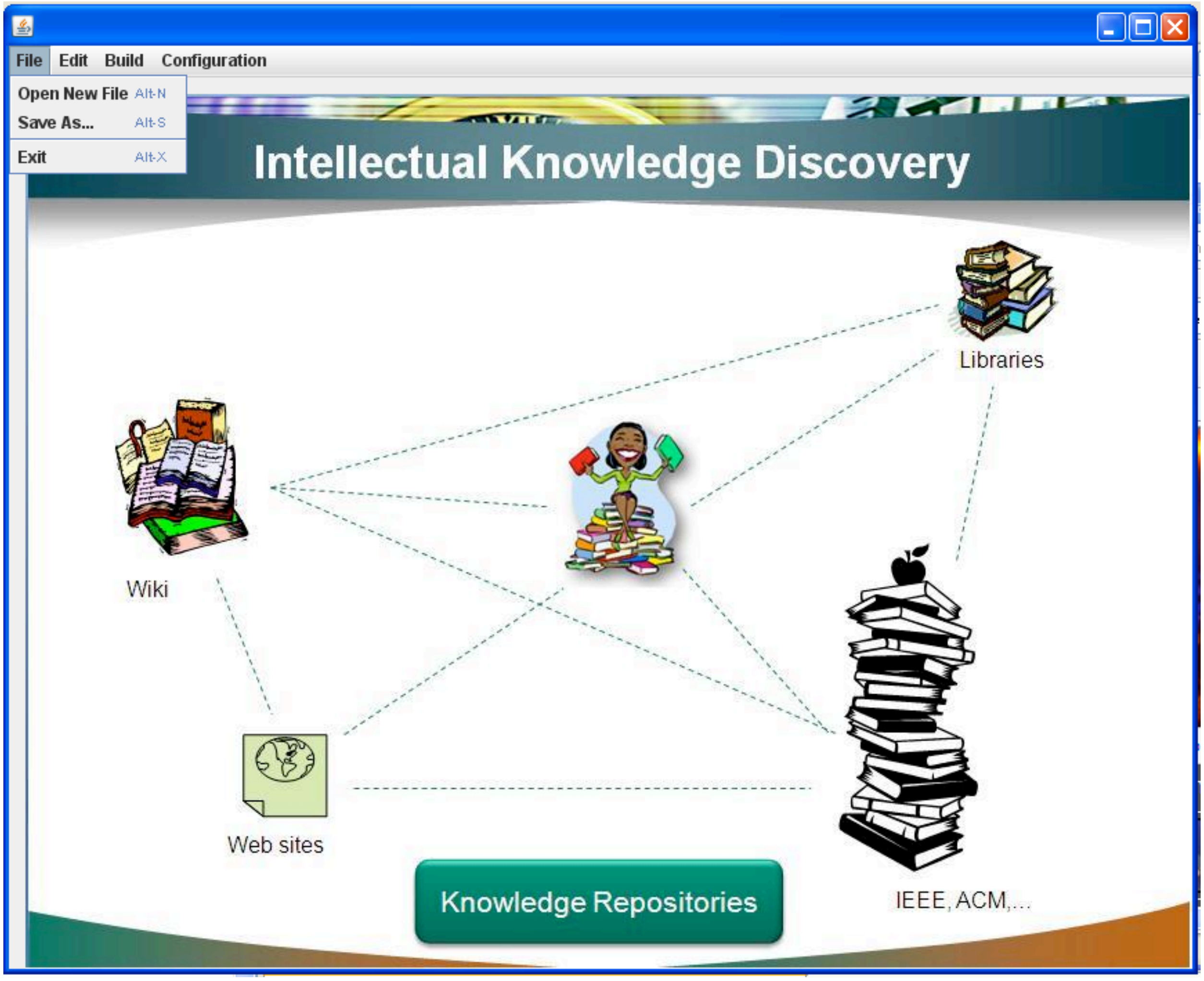


Fig. 1: File Menu

Build Menu:

It has two menu items:

- Compile the Annotated Text: Using this menu item, the annotated text is compiled and a file with the extension *.func is built. It also generates a LODL file for the annotated text opened. The LODL file is an XML file with a *.LODL extension.

- Build Customized Lesson: By clicking this menu item, another window open (See Fig.3)

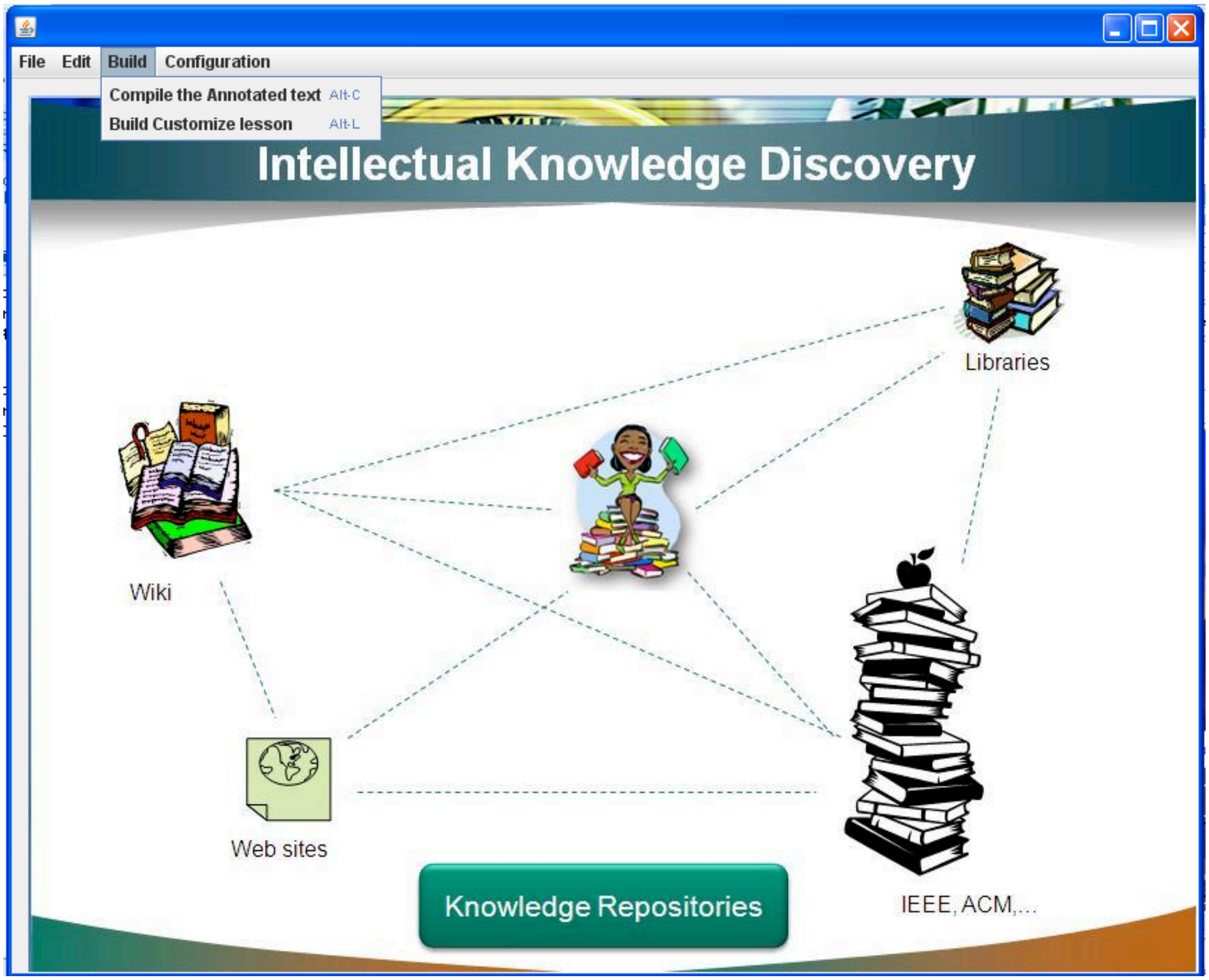
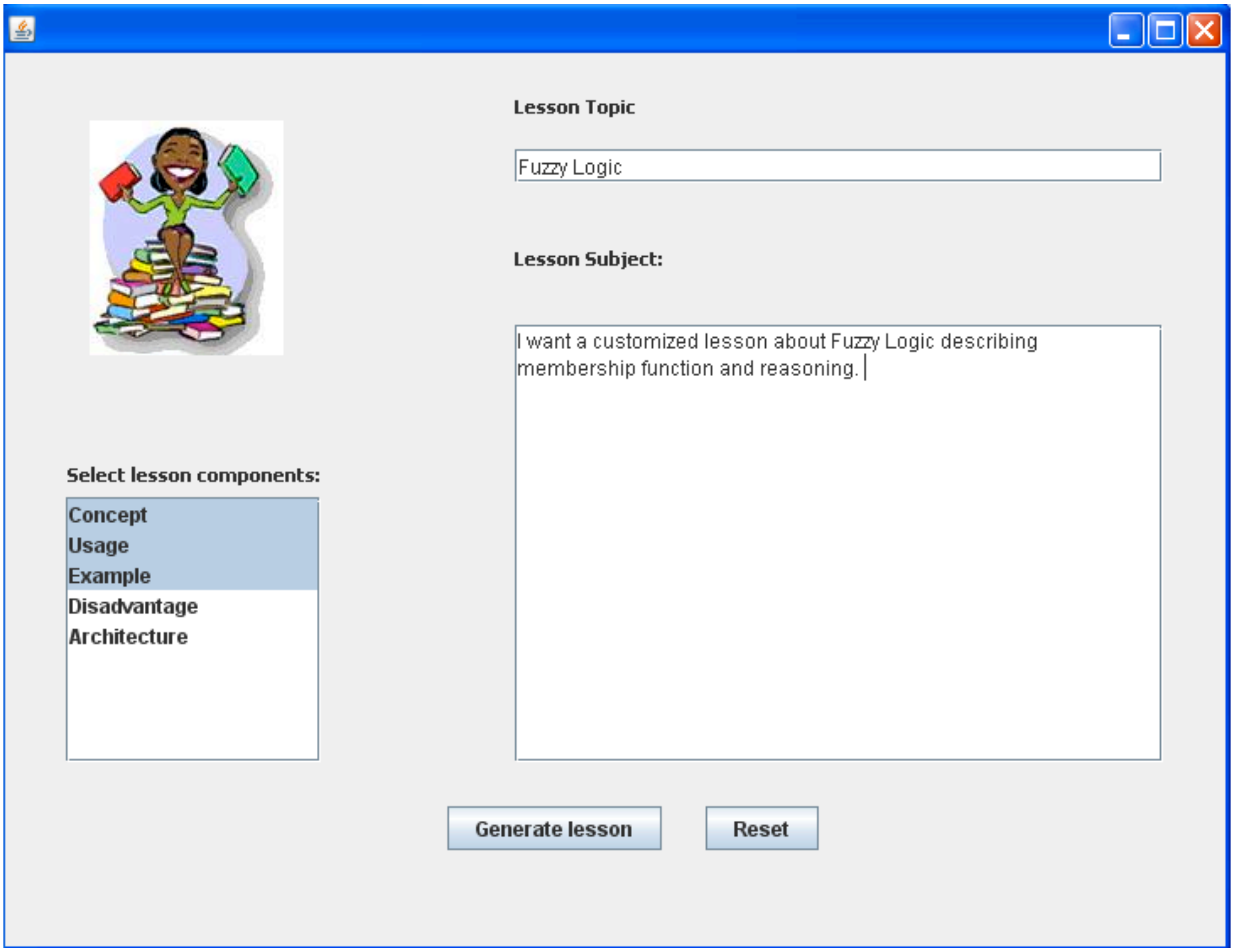

File
Edit
Build
Configuration
Compile the Annotated text Alt-C
Build Customize lesson Alt-L
Intellectual Knowledge Discovery
Libraries
Wiki
Web sites
IEEE, ACM,...
Knowledge Repositories

Fig. 2: Build Menu



Lesson Topic

Fuzzy Logic

Lesson Subject:

I want a customized lesson about Fuzzy Logic describing membership function and reasoning.

Select lesson components:

- Concept
- Usage
- Example
- Disadvantage
- Architecture

Fig. 3: Learning Path Window

In this window, you can enter your learning path containing Lesson topic, Lesson components and Lesson subject and then click on “Generate a lesson” button. It generates a customized lesson in a file with the extension *.gen. You can open this file in the program by clicking on File menu> Open New File.

B. *Executing the system*

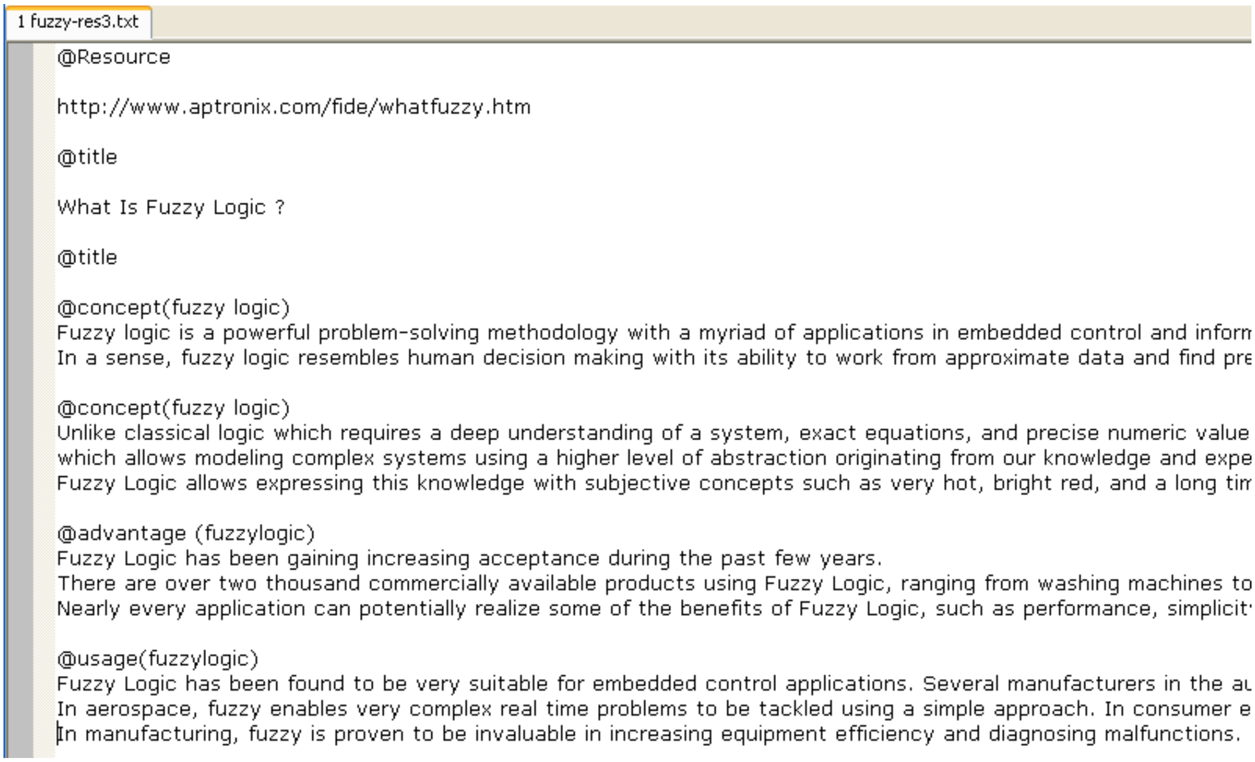
System has been written in Java and developed in NetBeans IDE. You can execute it by opening the project in NetBeans and pressing F6. Or if you want to run it through the command prompt, you should execute the following commands:

```
C:/> Javac  -classpath = .; c:\lib\opennlp.jar;    softcomputing .java

C:\> java    softcomputing
```

C. *Examples of Outputs*

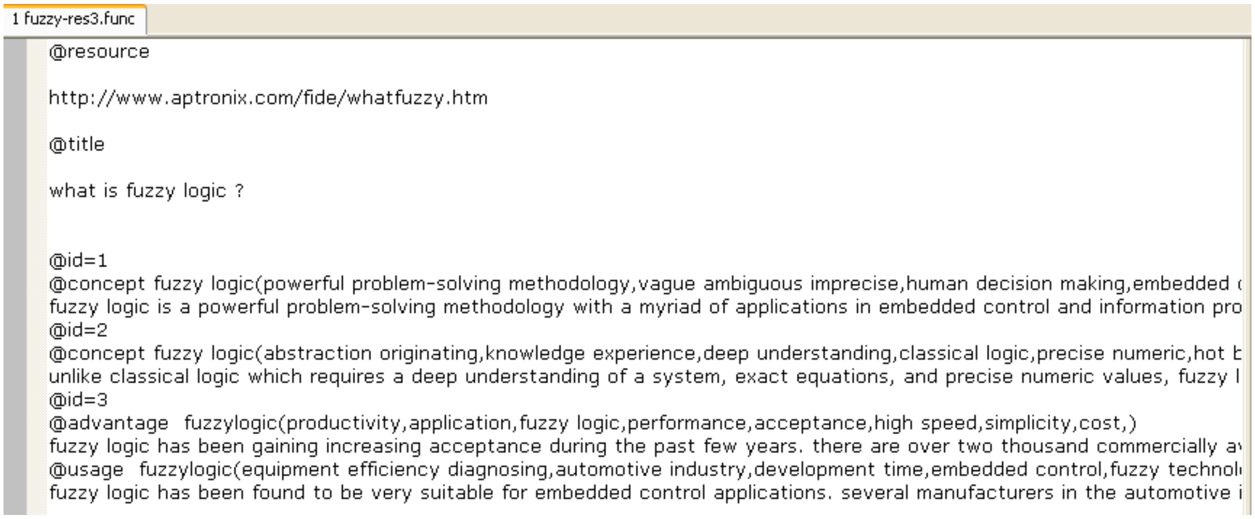
Sample examples of the program:

A text annotated manually:



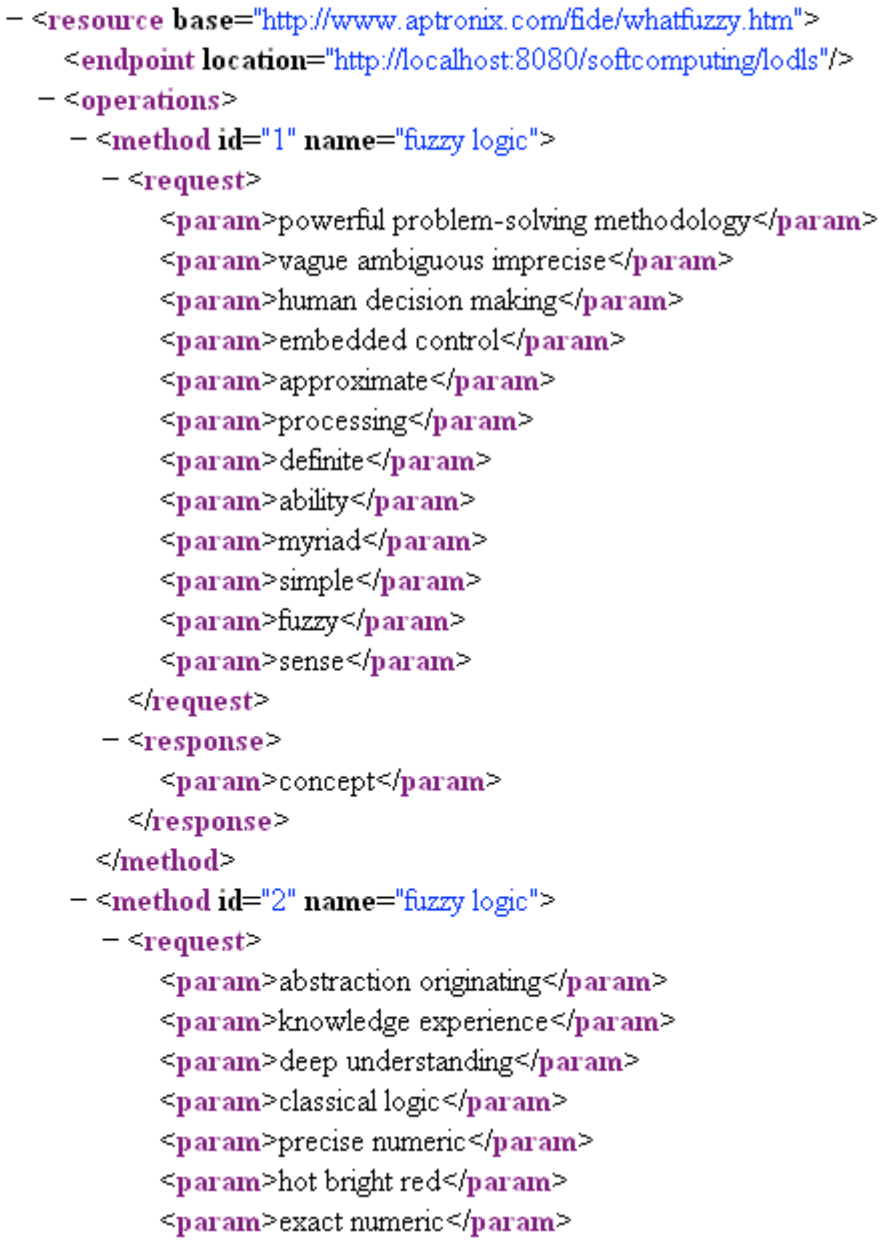
```
@Resource

http://www.aptronix.com/fide/whatfuzzy.htm

@title

What Is Fuzzy Logic ?

@title

@concept(fuzzy logic)
Fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and inform
In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find pre

@concept(fuzzy logic)
Unlike classical logic which requires a deep understanding of a system, exact equations, and precise numeric value
which allows modeling complex systems using a higher level of abstraction originating from our knowledge and expe
Fuzzy Logic allows expressing this knowledge with subjective concepts such as very hot, bright red, and a long tim

@advantage (fuzzylogic)
Fuzzy Logic has been gaining increasing acceptance during the past few years.
There are over two thousand commercially available products using Fuzzy Logic, ranging from washing machines to
Nearly every application can potentially realize some of the benefits of Fuzzy Logic, such as performance, simplicit

@usage(fuzzylogic)
Fuzzy Logic has been found to be very suitable for embedded control applications. Several manufacturers in the au
In aerospace, fuzzy enables very complex real time problems to be tackled using a simple approach. In consumer e
In manufacturing, fuzzy is proven to be invaluable in increasing equipment efficiency and diagnosing malfunctions.
```

A compile file generated with the extension *.func

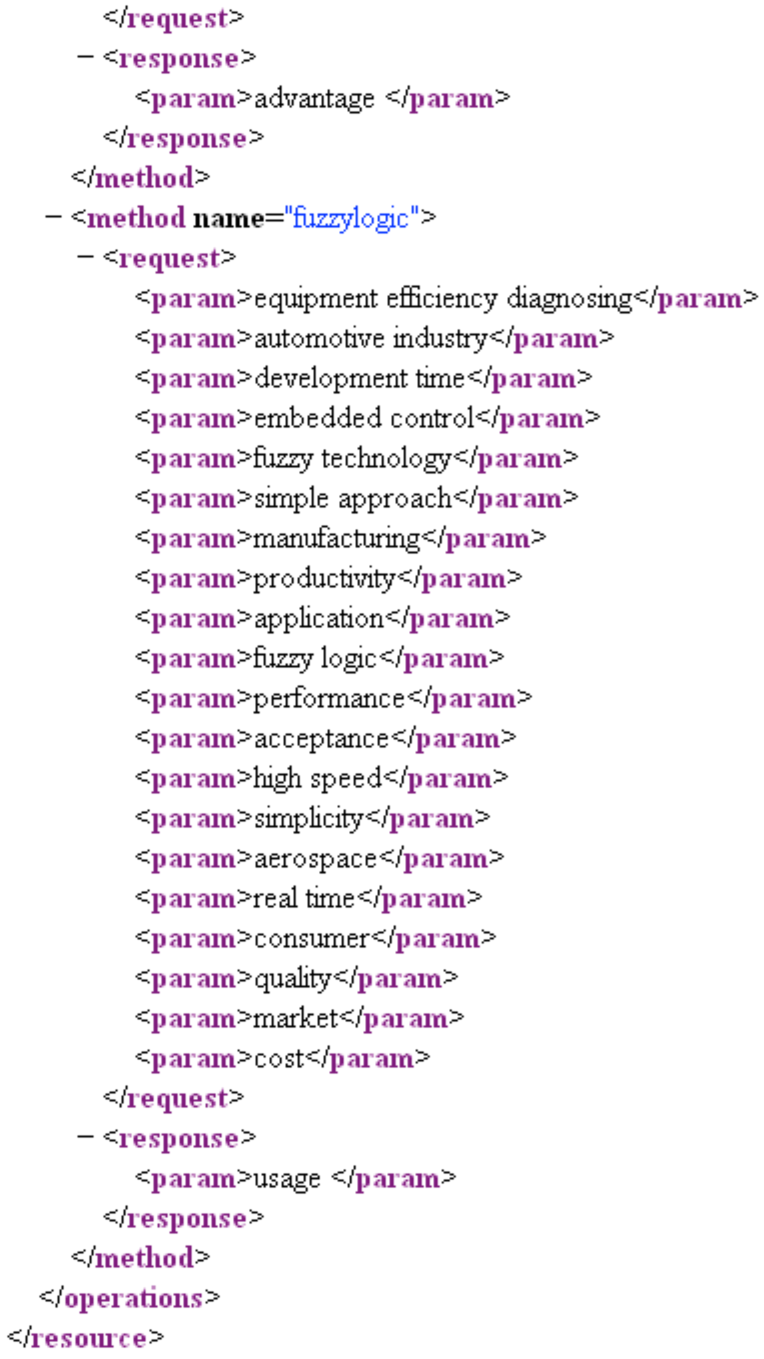


```
@resource

http://www.aptronix.com/fide/whatfuzzy.htm

@title

what is fuzzy logic ?


@id=1
@concept fuzzy logic(powerful problem-solving methodology,vague ambiguous imprecise,human decision making,embedded
fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information pro
@id=2
@concept fuzzy logic(abstraction originating,knowledge experience,deep understanding,classical logic,precise numeric,hot b
unlike classical logic which requires a deep understanding of a system, exact equations, and precise numeric values, fuzzy l
@id=3
@advantage  fuzzylogic(productivity,application,fuzzy logic,performance,acceptance,high speed,simplicity,cost,)
fuzzy logic has been gaining increasing acceptance during the past few years. there are over two thousand commercially a
@usage  fuzzylogic(equipment efficiency diagnosing,automotive industry,development time,embedded control,fuzzy technol
fuzzy logic has been found to be very suitable for embedded control applications. several manufacturers in the automotive i
```

A LODL generated by the program :

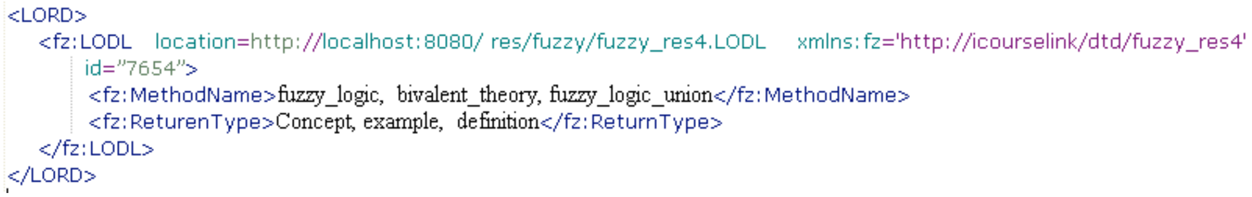


```
- <resource base="http://www.aptronix.com/fide/whatfuzzy.htm">
    <endpoint location="http://localhost:8080/softcomputing/lodls"/>
  - <operations>
    - <method id="1" name="fuzzy logic">
      - <request>
          <param>powerful problem-solving methodology</param>
          <param>vague ambiguous imprecise</param>
          <param>human decision making</param>
          <param>embedded control</param>
          <param>approximate</param>
          <param>processing</param>
          <param>definite</param>
          <param>ability</param>
          <param>myriad</param>
          <param>simple</param>
          <param>fuzzy</param>
          <param>sense</param>
        </request>
      - <response>
          <param>concept</param>
        </response>
      </method>
    - <method id="2" name="fuzzy logic">
      - <request>
          <param>abstraction originating</param>
          <param>knowledge experience</param>
          <param>deep understanding</param>
          <param>classical logic</param>
          <param>precise numeric</param>
          <param>hot bright red</param>
          <param>exact numeric</param>
    </request>
  − <response>
      <param>concept</param>
    </response>
  </method>
− <method id="3" name="fuzzylogic">
  − <request>
      <param>productivity</param>
      <param>application</param>
      <param>fuzzy logic</param>
      <param>performance</param>
      <param>acceptance</param>
      <param>high speed</param>
      <param>simplicity</param>
      <param>cost</param>
    </request>
  − <response>
      <param>advantage </param>
    </response>
  </method>
− <method name="fuzzylogic">
  − <request>
      <param>equipment efficiency diagnosing</param>
      <param>automotive industry</param>
      <param>development time</param>
      <param>embedded control</param>
      <param>fuzzy technology</param>
      <param>simple approach</param>
      <param>manufacturing</param>
      <param>productivity</param>
      <param>application</param>
      <param>fuzzy logic</param>
      <param>performance</param>
      <param>acceptance</param>
      <param>high speed</param>
```



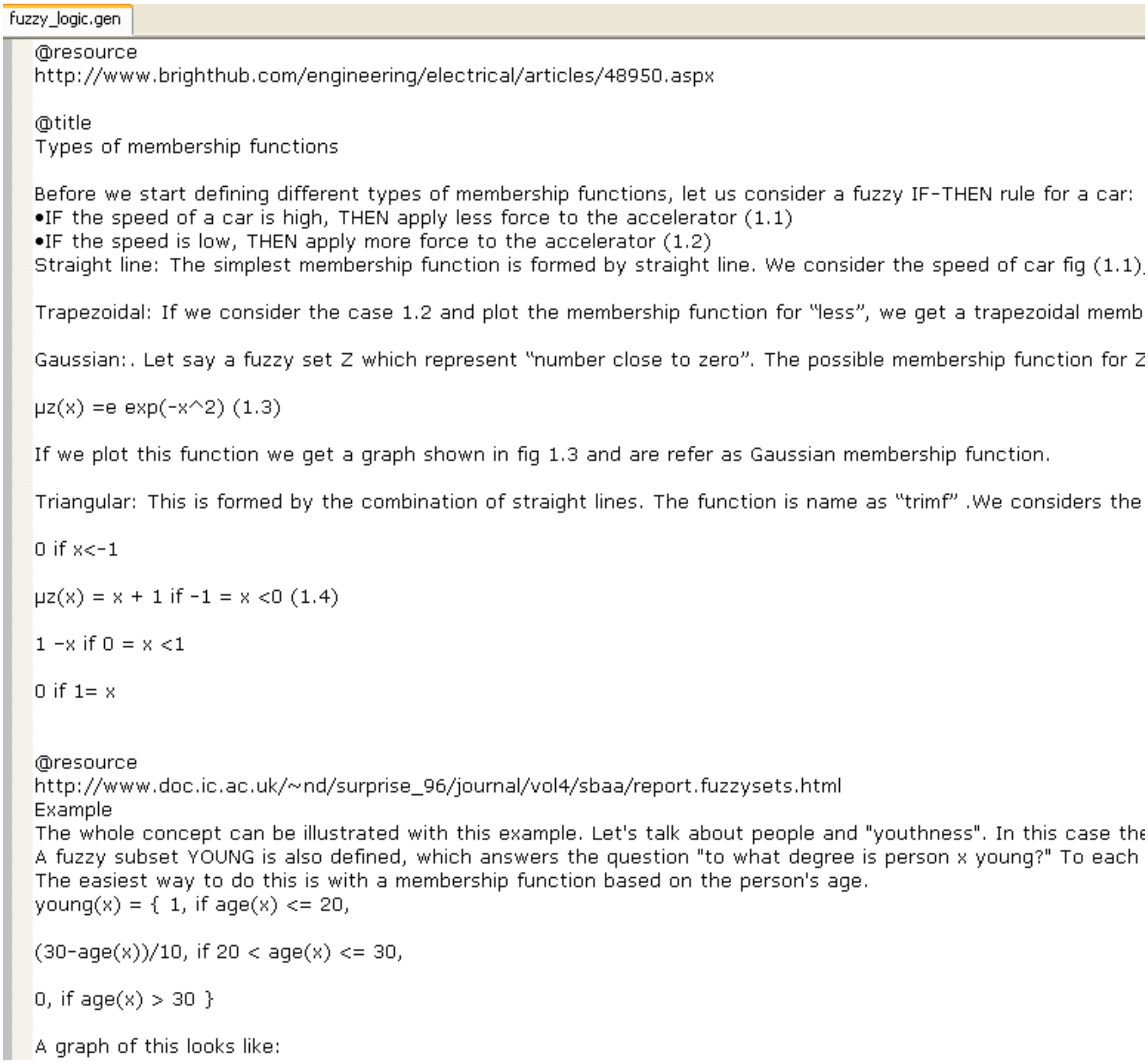
```
        </request>
      - <response>
          <param>advantage </param>
        </response>
      </method>
    - <method name="fuzzylogic">
      - <request>
          <param>equipment efficiency diagnosing</param>
          <param>automotive industry</param>
          <param>development time</param>
          <param>embedded control</param>
          <param>fuzzy technology</param>
          <param>simple approach</param>
          <param>manufacturing</param>
          <param>productivity</param>
          <param>application</param>
          <param>fuzzy logic</param>
          <param>performance</param>
          <param>acceptance</param>
          <param>high speed</param>
          <param>simplicity</param>
          <param>aerospace</param>
          <param>real time</param>
          <param>consumer</param>
          <param>quality</param>
          <param>market</param>
          <param>cost</param>
        </request>
      - <response>
          <param>usage </param>
        </response>
      </method>
    </operations>
  </resource>
```

A LORD generated by program in a serialized format and read by Wscite:



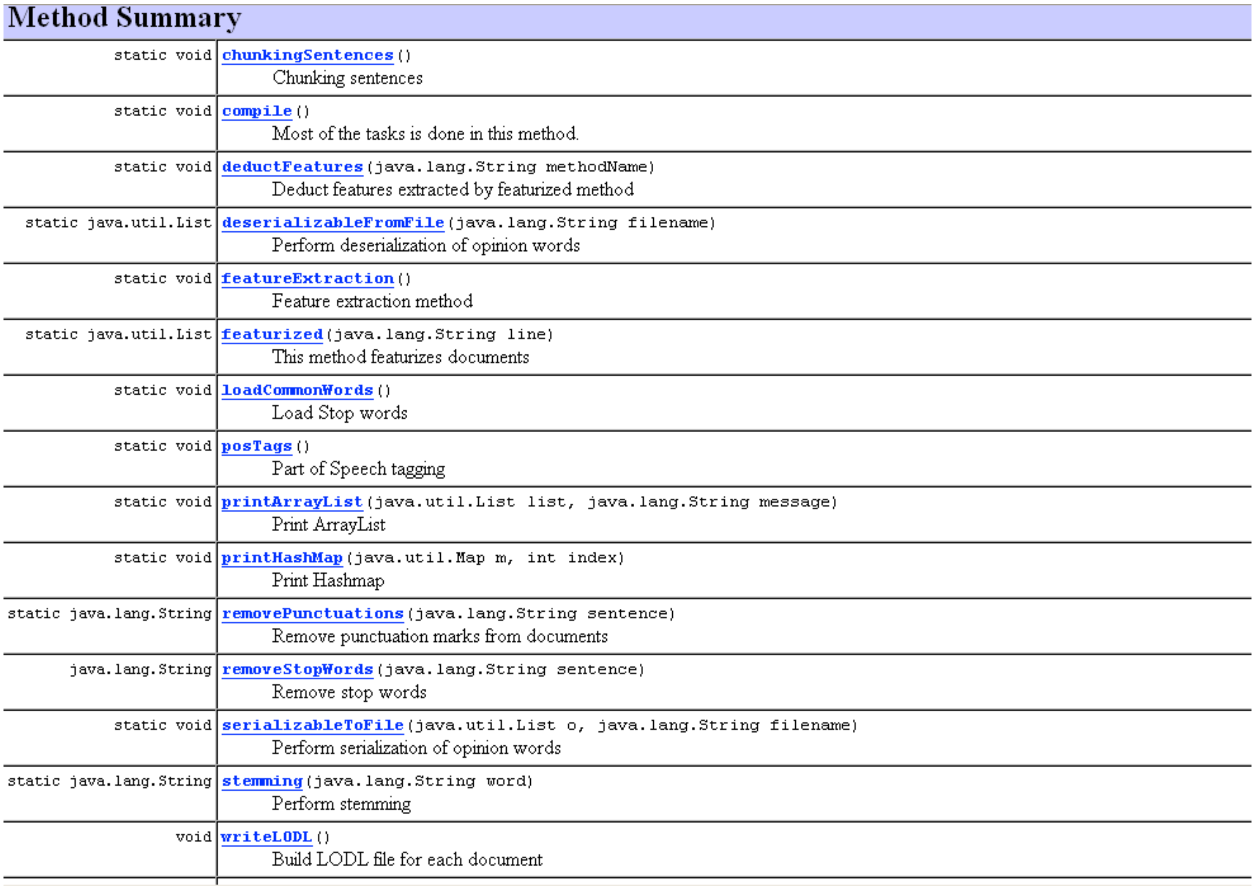
```
<LORD>
   <fz:LODL   location=http://localhost:8080/ res/fuzzy/fuzzy_res4.LODL    xmlns:fz='http://icourselink/dtd/fuzzy_res4'
       id="7654">
        <fz:MethodName>fuzzy_logic,  bivalent_theory, fuzzy_logic_union</fz:MethodName>
        <fz:ReturenType>Concept, example,  definition</fz:ReturnType>
   </fz:LODL>
</LORD>
```

A customized lesson generated by program (the most relevant learning object Ids are discovered by the search algorithm initiated in section IV-E.

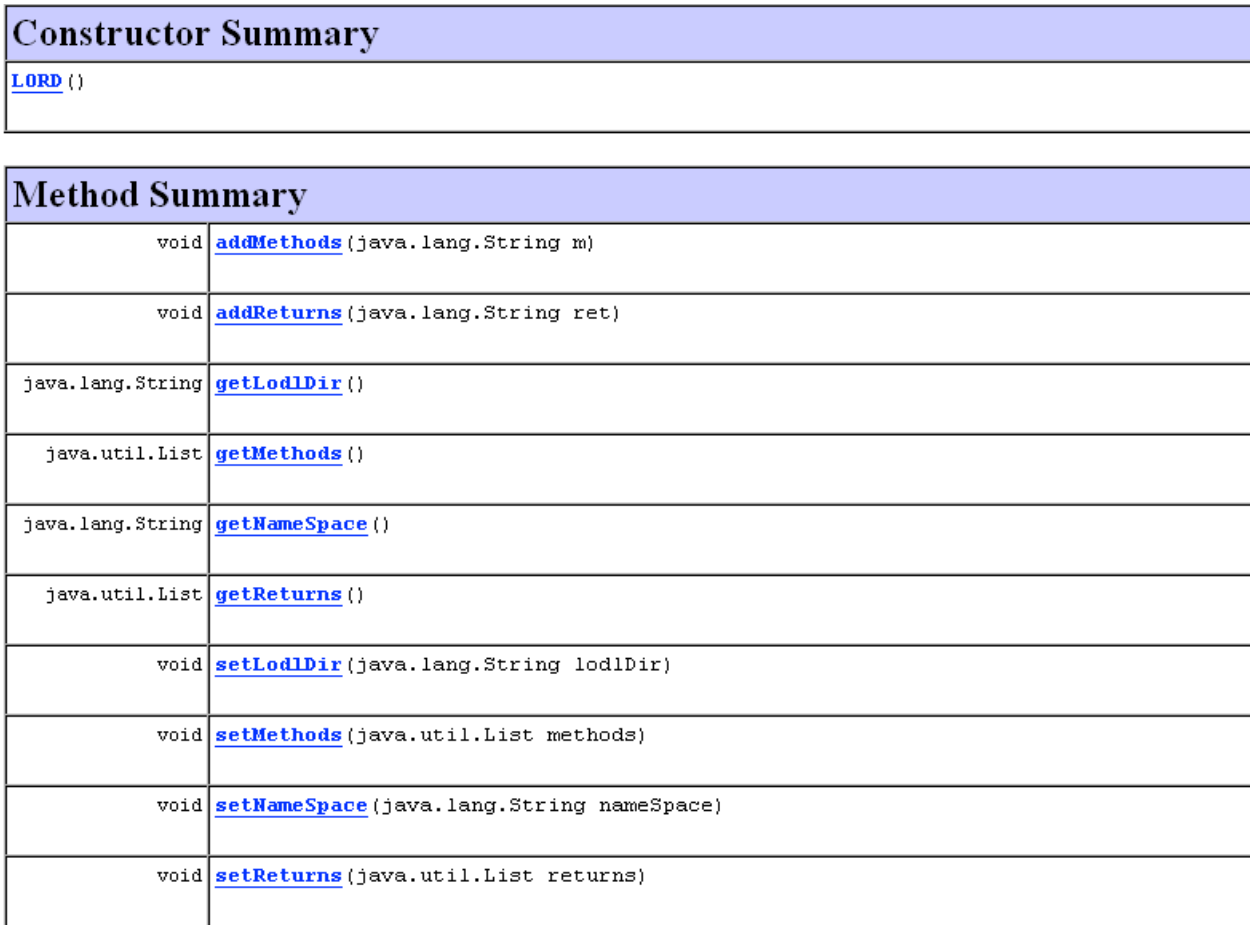


```
@resource
http://www.brighthub.com/engineering/electrical/articles/48950.aspx

@title
Types of membership functions

Before we start defining different types of membership functions, let us consider a fuzzy IF-THEN rule for a car:
•IF the speed of a car is high, THEN apply less force to the accelerator (1.1)
•IF the speed is low, THEN apply more force to the accelerator (1.2)
Straight line: The simplest membership function is formed by straight line. We consider the speed of car fig (1.1).

Trapezoidal: If we consider the case 1.2 and plot the membership function for “less”, we get a trapezoidal memb

Gaussian:. Let say a fuzzy set Z which represent “number close to zero”. The possible membership function for Z

µz(x) =e exp(-x^2) (1.3)

If we plot this function we get a graph shown in fig 1.3 and are refer as Gaussian membership function.

Triangular: This is formed by the combination of straight lines. The function is name as “trimf” .We considers the

0 if x<-1

µz(x) = x + 1 if -1 = x <0 (1.4)

1 –x if 0 = x <1

0 if 1= x


@resource
http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.fuzzysets.html
Example
The whole concept can be illustrated with this example. Let's talk about people and "youthness". In this case the
A fuzzy subset YOUNG is also defined, which answers the question "to what degree is person x young?" To each
The easiest way to do this is with a membership function based on the person's age.
young(x) = { 1, if age(x) <= 20,

(30-age(x))/10, if 20 < age(x) <= 30,

0, if age(x) > 30 }

A graph of this looks like:
```

D. Method Summary



Method Summary	
static void	chunkingSentences() Chunking sentences
static void	compile() Most of the tasks is done in this method.
static void	deductFeatures(java.lang.String methodName) Deduct features extracted by featurized method
static java.util.List	deserializableFromFile(java.lang.String filename) Perform deserialization of opinion words
static void	featureExtraction() Feature extraction method
static java.util.List	featurized(java.lang.String line) This method featurizes documents
static void	loadCommonWords() Load Stop words
static void	posTags() Part of Speech tagging
static void	printArrayList(java.util.List list, java.lang.String message) Print ArrayList
static void	printHashMap(java.util.Map m, int index) Print Hashmap
static java.lang.String	removePunctuations(java.lang.String sentence) Remove punctuation marks from documents
java.lang.String	removeStopWords(java.lang.String sentence) Remove stop words
static void	serializableToFile(java.util.List o, java.lang.String filename) Perform serialization of opinion words
static java.lang.String	stemming(java.lang.String word) Perform stemming
void	writeLODL() Build LODL file for each document



Constructor Summary

LORD()

Method Summary

void	**addMethods**(java.lang.String m)
void	**addReturns**(java.lang.String ret)
java.lang.String	**getLodlDir**()
java.util.List	**getMethods**()
java.lang.String	**getNameSpace**()
java.util.List	**getReturns**()
void	**setLodlDir**(java.lang.String lodlDir)
void	**setMethods**(java.util.List methods)
void	**setNameSpace**(java.lang.String nameSpace)
void	**setReturns**(java.util.List returns)