Fuzzy Concept Set Based Organizational Memory as a Quasi Non-Semantic Component within the INFRAWEBS Framework

Galya Andonova
Active Solutions Ltd.
Sofia, Bulgaria
galya.andonova@activesolutions.bg

Gennady Agre
Institute of Information Technologies - BAS,
Sofia, Bulgaria
agre@iinf.bas.bg

H Joachim Nern
Aspasia Systems Ltd.
Duesseldorf,
Germany
nern@aspasia-systems.de

Andrei Boyanov
Active Solutions Ltd.
Sofia, Bulgaria
andrei.boyanov@activesolutions.bg

Abstract

The intention of this article is to describe the use of fuzzy logic techniques applied to an Organizational Memory (OM) – a module of the INFRAWEBS framework [1] dedicated to the collection, organization, refinement, and distribution of entity specific knowledge. The module is based on the new streams and approaches in the area of Fuzzy Concept Set (FCS) Modeling, especially Fuzzy Concept Matching (FCM) [2]. The paper gives an overview about the INFRAWBS conceptual model as well as presents the automatic object feature vector generation, objects classification and clustering mechanisms, as well as techniques for retrieving similar objects and ontology based semantic objects descriptions.

Keywords: Semantic Web Service, WSMO, WSDL, WSML, Ontology, Fuzzy matching, Classification, Clustering.

1 INFRAWEBS Framework

The current technology based on SOAP, WSDL and UDDI addresses only syntactical aspects of a Web service. On this basis just “rigid” (static) services can be designed and invoked. Rigid and non-dynamical means, that they cannot adapt and reflect changes (or disturbances in terms of automatic control) of the action environment without human intervention. Web services based on above mentioned technologies are not “really” automatically driven – their action cycles depend mainly on semi-automatic structures, due to missing closed loop features.

Accordingly new streams and project initiatives are developed to increase the degree of automation and the flexibility of service providing and service requesting activities by applying soft computing as well as Fuzzy Logic approaches [3, 5, 6, 7, 8, 9, 12].

The INFRAWEBS framework represents an application-oriented software “tool set” for creating, maintaining and executing WSMO-based [10] Semantic Web Services within the whole life cycle. The main users of the INFRAWEBS will be Web service providers as well as semantic Web service application developers, who register their applications into the system. The INFRAWEBS provides a loosely-coupled set of adaptable system components and tools to analyze design and maintain Semantic Web Services (SWS) in the whole service life cycle: discovery, selection, execution and monitoring. The specific approach follows strictly the bottom up paradigm - it combines and couples knowledge management issues (representing the basis) with SWS issues (representing the top of service advertising and offering environments).

The open platform developed within the INFRAWEBS project consists of coupled and linked Semantic Web Units (SWU), whereby each unit provides tools and system components to analyze, design and maintain Web services realized as semantic Web services within the whole life cycle.

The overall design of the INFRAWEBS (Figure. 1) is structured in three main layers:

1) Knowledge management layer for handling service related knowledge artifacts realized as an organizational memory coupled to semantic information routing components (OM & SIR) [1];
2) **Service development layer** for creating and maintaining SWS embedded in a semantic based interoperable middleware, consisting of Semantic Web Service Designer & Composer, Distributed Semantic Web Service Registries, and an agent based discovery module (Semantic Web Service Unit -SWU);

3) **Service deployment layer** for the execution and monitoring of Semantic Web Services exploiting closed loop feedback information (Quality of Service brokering) provided for monitoring and execution issues.

- **Information structures** for effective discovering, storing and retrieving both semantic and non-semantic information needed for creating and maintaining Semantic Web Services
- **Tools** for creating and maintaining Semantic Web Services and SWS Applications
- **Library of Methods** used for creating and maintaining Semantic Web Services

The INFRAWEBS Environment provides means for communicating with different kinds of INFRAWEBS users and other INFRAWEBS Semantic Web Units as well as for executing SWS ensuring security and privacy of these operations.

2 **INFRAWEBS Conceptual Model**

The conceptual model of the INFRAWEBS (Figure 2) is aimed to serve the following categories of users:

**Web Service Provider** – a provider of an existing Web service, who would like to convert it to the Semantic Web service and to publish it.

**Web Service Broker** – an entity, who would like to create and publish a Semantic Web Service with some desired functionality via composition of several existing Semantic Web services.

**Web Service Application Provider** – an entity, who would like to design an own application based on Semantic Web Service Technology.

**Web Service Application User** – an “ordinary” end-user of a Web Service Application, who would like to use INFRAWEBS for retrieving a Web Service able to satisfy the request (goal).

The INFRAWEBS Conceptual Model consists of two main elements: the INFRAWEBS Semantic Web Unit and the INFRAWEBS Environment. The INFRAWEBS Semantic Web Unit provides users with:

- **Tools** for creating and maintaining Semantic Web Services
- **Library of Methods** used for creating and maintaining Semantic Web Services

The INFRAWEBS Environment provides means for communicating with different kinds of INFRAWEBS users and other INFRAWEBS Semantic Web Units as well as for executing SWS ensuring security and privacy of these operations.

All SWU components are organized in two directions:

- Problem solving based on semantic information (or Logic-based problem solving)
- Problem solving based on non-semantic information (similarity-based problem solving)

From these points of view the SWU proposes:

**Information structures** for storing and retrieving semantic and non-semantic data that are DSWS-R (Distributed Semantic Web Service Repository) and the Similarity-based OM:

- **DSWS-R** is aimed at effective storing and retrieving all elements of Semantic Web according to the WSMO Framework: Goals, Ontologies, SWS and Mediators written in WSML.
- **Similarity-based OM** contains non-logical representation of the same (or some of the) elements (so called “knowledge objects”)
as well as other data e.g. graphical models of SWS and “natural language” templates for WSMO Goals.

**Tools for creating and maintaining both semantic and non-semantic data:**

- **CBR based Service Designer** is aimed at designing a WSMO-based Semantic Web service from an existing non-semantic Web service.
- **CBR based Service Composer** is aimed at creating a Semantic Web service through composition of existing WSMO-based semantic Web services.
- **CBR based Goal Editor** is aimed at creating predefined WSMO-based goals and their “natural language” templates needed for designing SWS-based applications.
- **CBR based Recommender tool** is a similarity-based tool facilitating operation of all INFRAWEBS “semantic-based” tools by utilizing “past experience”. Non-semantic data stored in OM is used as the problem description for determining the most similar solution (SWS or its graphical model) to the current problem.

**Methods used during problem-solving**

- Logic-based discovery.
- Application-specific decision-support methods used for service composition, compensation, monitoring etc.
- Ontology key words based discovery
- Several methods for calculating similarity and/or assignments – structural, linguistic, statistic etc.

Semantic and non-semantic components of INFRAWEBS are interconnected by a Semantic Information Router (SIR), which is responsible for:

- Locating all resources needed for problem solving either in the local SWU or outside.
- Creation of non-semantic content (knowledge objects) by means of the semantic content stored in the DWSW-R.
- Creating an effective system of indexes allowing fast communication between semantic and non-semantic modules of SWU.

The objective of this paper is to present main design principles of the INFRAWEBS Organizational Memory (FCS-OM), which uses the Fuzzy Concept Set modeling to solve the following tasks:

- Classification and retrieval of semantic and non-semantic knowledge;
- Implementation of a case-based memory providing solutions for natural language and ontology-based queries;
- Non-semantic mapping of semantic web services to the given requester goals as an implementation of the first step of the service discovery process;
- Provision of similarity calculation methods;

3 **FCS-OM Design and Concepts**

The FCS-OM represents a similarity based organizational memory and a case based recommender tool. The component classifies the system knowledge objects using fuzzy logic techniques as well as statistical and linguistic algorithms as fuzzy matching functions.

3.1 **Main Elements**

The system knowledge is represented by semantic and non-semantic knowledge objects. Semantic knowledge objects are the WSML descriptions [17] of semantic web services, ontologies, goals etc. WSDL files are non-semantic system knowledge objects containing non-semantic web services descriptions.

Knowledge objects are classified to a given pre-configured taxonomy composed out of classification subjects. Each knowledge object of the organizational memory is categorized by similarity- (and relationship-) coefficients determining the relation between objects and subjects regarding the given taxonomy. FCS-OM module uses different types of system dictionaries – knowledge domain keywords, generic keywords, abbreviations, stop terms and synonyms. These dictionaries are used by the algorithms for creating a (meta data) feature vector representation of knowledge objects and “centroids”, whereby the so called “centroids” are formalized meta data representations of subjects. The (meta data) centroids are synthetically generated objects representing the quasi centre of the set of objects.
assigned to a subject (see following section). The module maintains also a list of keywords recommendations that are extracted automatically out of each new object stored in the system. Accordingly each keyword proposition is processed and added to one of the system dictionaries.

The FCS-OM supports multi-dimensional classification of knowledge objects. For this purpose multiple dictionary sets are also supported:

- Ontology based dictionaries
- Natural language dictionaries

A fuzzy mapping between these two kinds of dictionaries is maintained by the system and is used for retrieving semantic information by a natural language query.

**Centroids** are the representatives of each set of objects. Each centroid is composed of centroid feature vector (CFV) and mapped to a classification subject given by the system taxonomy. Multiple layers of centroids are maintained by the component to allow the multi-dimensional classification of objects.

Every knowledge object is represented by an object feature vector (OFV). An OFV is a list of weighted keywords that is created automatically by a filtering process. Multiple layers of OFVs are supported by the system, whereas different methods in each layer are used.

The component combines semantic and non-semantic classification of objects. Additionally the semantic objects are classified by different semantic criteria. This imposes the necessity of multiple classification layers, where knowledge objects are represented by different OFVs and classified to the layers centroids.

### 3.2 Generation of Centroids and OFVs

The FCS-OM supports two types of centroids:

- centroids based on natural language words;
- centroids based on ontology terms.

The centroids based on natural language words are generated by a clustering process. These centroids are dynamically reconfigured by the system in the case a “new” object is stored in the objects’ collection.

The centroids based on ontology terms are automatically created from the ontologies stored in the DSWS-R component of the INFRAWEBS system. They are reconfigured automatically in the case a new ontology is stored.

The knowledge object feature vector generation is performed automatically during the process of adding the object to the objects’ collection. The feature vector creation is achieved by a filtering process. The filtering of generic objects consists of automatic extraction of object’s metadata and relevant keywords. The OFVs generated out of generic knowledge objects are non-semantic object feature vectors (NS-OFV) The metadata processing considers the meaning of different metadata fields. Keywords from different metadata fields are weighted in different way.

The filtering and normalization is performed as a fuzzy matching procedure. Subsequently the generated feature vector undergoes a classification process, which is performed using statistical similarity and assignment approaches for determining relationship functions among the objects themselves and pretended subject term vectors (centroids, representing the subjects), to which the objects are assigned and allocated.

A further applied fuzzy matching process using the predefined dictionaries of standard keywords and knowledge-domain terms extracts and weights implicitly terms and normalize the term vector to the given sets of standard keywords and knowledge-domain terms.

Creation of a semantic objects feature vector (S-OFV) is performed by extracting ontology concepts out of WSML files. The Semantic objects feature vector consists of weighted ontology concepts. The system dictionaries used for these feature vectors are populated with ontology concepts too.

### 3.3 Similarity and Assignment Function Calculations

Similarity calculation methods are used as fuzzy matching functions. Several methods for similarity calculation are provided by the FCS-OM:

- Statistical methods for calculation of the similarity between two objects;
- Linguistic methods for calculating the similarity between terms;
• Combined methods for complex similarity calculations between ontology based and natural language based OFVs.

The result of the assignment function and similarity calculation yields in a similarity resp. assignment function matrix, which couples the processed objects by similarity coefficients (Figure 3).

Figure 3: Calculation of a knowledge content object similarity and assignment matrix

3.4 Objects clustering

A three stage bottom up clustering is processed by using stepwise the MinMax-Algorithm and processing the assignment matrix of the objects given in the collection.

In Figure 4a the membership function $\mu$ is given over the centroids of the median sets $ms$ (R, Q, I, J … G, Z, N; 2nd layer in the subject tree). $FS_{root}$ gives the fuzzy root concept set, consisting of fuzzy median concept sets, whereby $\mu_{root}(ms)$ gives the assignment function of the fuzzy median concept centroids. Marked in bold is the assignment function $\mu_k(ms)$ of the fuzzy concept median set $FS_k$ (K).

The set $FS_k$ marked in bold with the assignment function $\mu_k(ms)$ is dissolved in Figure 4b as a set with the function $\mu$ (bs) dependent on the fuzzy concept bottom sets bs. Marked in dark bold is the element set $FS_{k01}$ with the assignment function $\mu_{k01}(e)$ dependent on the elements (knowledge content object, concept) in the bottom set $K_0$.

In Figure 4c the element set $FS_{k01}$ (marked in bold) with the assignment function $\mu_{k01}(e)$ is dissolved over the elements of the bottom sets. Marked in dark bold is exemplarily the word set $FS_{k01,6}$ with the assignment function $\mu_{wk01,6}$ dependent on the words in the element set.

The word set $FS_{k01,6}$ (marked in dark bold) with the assignment function $\mu_{wk01,6}$ is dissolved in Figure 4d over the words of the elements. The assignment coefficients of the singular words (Ko1,6.1, Ko1,6.2, … Ko1,6.9) to the given element Ko1,6 is equal to the discrete points of $\mu_{wk01,6}$.

The visualizations in Figure 4b, c, d illustrate the capability of the OM to determine the overall assignment (or membership) of a single “word” to the overall knowledge stored in the OM – the capability to identify the degree of membership of a single word to the overall knowledge (local similarity / global similarity – local membership / global membership).

Defined vice versa: the OM has the capability to identify a global fuzzy root set of concepts (global knowledge), consisting of fuzzy median sets of concepts (centroids), consisting of fuzzy sets of elements, consisting of sets of (atomic) words.

3.5 Objects classification

Objects are classified to subjects of the classification subject’s tree based on processing the assignment functions. Each subject is characterized and represented by centroids, whereby the centroids are formalized as feature vectors (CFV), which are structural identical with the object feature vectors (OFV).

During the classification procedure every similarity resp. assignment coefficient between an object and a given centroid is classified in one of three similarity sets – min, middle or max. The min set contains objects with minimal similarity coefficients between them and the subject, the max set contains objects with maximum similarity coefficients between them and the subject and the middle set contains objects with middle coefficients.

4 Using FCS-OM as a Case-based Memory

The FCS-OM acts as a case based memory and recommender tool in the context of the INFRAWEBs system [18]. The component uses the objects classification and clustering for determining appropriate objects for different kind of case-based queries.

4.1 Query by imported ontologies

This is the simplest type of queries, which requests for existing services described by the same (or almost the same, “quite similar”) set of (imported) ontologies.
Figure 4a, b, c, d: Three stage bottom up clustering of concept sets yielding in: root set, medion sets, element sets, word sets.
The rationality behind this is an assumption that services, which used the same set of ontologies, are likely to belong to the same domain. Practically such an option is equivalent to browsing capabilities of available semantic services from the same problem domain and may be useful for receiving some initial ideas of how the desired service capability should look like. It is not expected to receive a precise selectivity but some kind of fuzzy one from such type of the query.

4.2 “Natural language” queries
“Natural language” queries are intended for searching existing semantic services, which descriptions contain words (terms) specified in the query. Since natural language is used mainly for describing nonfunctional properties of a semantic service (e.g. title, publisher, description, etc.) such a query will be matched against such service properties. The queries may be unstructured or structured. In the first case matching is performed against all nonfunctional properties of the service, while in the second case the similarity is evaluated only using nonfunctional properties specified in the query. For example, it can be requested for the “similar” services created by a concrete organization (service publisher) or written in a concrete natural language or even created by a concrete person (service nonfunctional property “contributor”).

In order to guarantee the compatibility of the retrieved service with the already selected set of ontologies, such “natural language” queries are automatically considered in conjunction with the first type of queries (queries by a set of ontologies).

4.3 Ontology-based queries
The mere a-priori information available to the user before starting to create the description of the service’s axiom is an initial set of ontologies, which was selected as the most appropriated for semantic description of the Web service during the generation process of the service grounding. So a single way to represent more clearly the “meaning” of the new axiom is to describe it as a set of concepts and relations that user thinks to include in its description. Since an axiom can exist only as a named part of a Semantic service description, the meaning of such a query is to find a set of existing Semantic Web services, which capability descriptions contain the set of ontological concepts specified in the query.

4.3.1. Unstructured queries
Even in this case it is possible to formulate a set of queries with a different meaning:
- Query for a semantic service, which capability description as a whole contains (or is similar to) a specified set of ontological concepts.
- Query for a semantic service containing one or several axioms, which descriptions separately contain (or similar to) a specified set of ontological concepts.

In the first case the similarity is measured based on the overall ontology concepts accumulated from all axioms participated in the service capability description. In other words the query is matched against a compound axiom constructed by merging all service capability axioms. The most similar service will be a service, which capability axioms have the highest average similarity (overall similarity) with the query.

In the second case the query is matched against each capability axiom (individually). The most similar service will be the one containing an axiom, which has the highest similarity with the query.

4.3.2. Structured queries
The next natural step is, to allow the user to construct structured queries. Such queries are specified not only by the ontological concepts the service capability description should have, but also considering which part of such a description (i.e. postconditions, assumption etc.) should have which set of ontological words.

For example, it will be possible to retrieve an existing semantic service, whereas the preconditions are the most similar to the set of ontological words specified in the query; or to find such a service which preconditions are similar to one ontological word set and which postconditions are similar to another set of such words defined in the query. In such case the most similar service will be the service with the highest average aggregated similarity (overall similarity) to the structural query.

5 INFRAWEBS Application
The INFRAWEBS Consortium is developing a semantic application performing a Frequent
Flyer Program (FFP) [19] in which the customers can create and reuse travel packages. The application is built upon a Service Oriented Architecture, accessing, discovering, composing and invoking Semantic Web Services for the management of the Travel Packages. The composition of semantic services is driven by Choreography, using the Web Service Modeling Ontology (WSMO see http://www.wsмо.org) as a framework to describe both the service capability and the service behavior. The prototype implementation of the FFP is expected to be realized until the end of 2006.

Acknowledgement
This work is supported by the EC funded IST project INFRAWEBS (IST FP6-511723) within the framework FP 6.

6 References