Ontology based Rule Base Fuzzy Cognitive Maps

Alejandro Peña*, Humberto Sossa** and Francisco Gutierrez*

WOLNM1, UPIICSA2 & CIC3 - National Polytechnic Institute2,3, México

*31 Julio 1859 # 1099B Leyes Reforma, DF 09310 México
apenaa@ipn.mx
atornes@cic.ipn.mx

**Juan de Dios Batiz s/n esq. Miguel Othón, Zacatenco, DF 07738 México
hsossa@cic.ipn.mx

Abstract: This work proposes a framework for designing and deploying Ontologies oriented to describe Rule Base Fuzzy Cognitive Maps. The approach takes into account the foundations of the Fuzzy Cognitive Maps and the baseline of the Ontologies. With these underlying elements, a specification of a conceptualization about the modeled domain is depicted. Thus, the concepts and causal relationships that fit fuzzy Rule Bases are stated. As a result, it is organized a semantic repository that is encoded into Ontology Web Language. The administration of the Ontology is in charge of an Ontology Agent. This manager is responsible for updating the meta-definitions and instances that compound the Ontology. In addition, the Ontology Agent carries out the inferences for answering the queries requested by a community of agents. This set of agents recreates a Multi-Agent System that is deployed in the Internet by means of Web Services. The whole system is oriented to generate the structure of Fuzzy Cognitive Maps.

Key words: Fuzzy Causal Reasoning, Fuzzy Cognitive Maps, Fuzzy Rule Base, Ontology.

INTRODUCTION

A Cognitive Map is a mental model about a specific problem domain that is analyzed from the cause-effect perspective. The entities of the domain are objects and events that are stated as concepts. Stimulus and inhibitions between couples of concepts are considered causal relations [Peña, et al. 05] and [Peña, et al. 06]. Fuzzy Cognitive Maps use linguistic terms to identify the state of the concepts and the intensity of the causal influences.

Due to the plenty of fuzzy values stemmed from the universe of discourse attached to the concepts, it is necessary a rule base for each causal relation. As a consequence, it is required an appropriate repository for stating the meta-description and the instantiation of the elements of the Fuzzy Cognitive Map. This knowledge structure must be grounded on the axioms that specify the intended meaning of such vocabulary.

In addition, the model needs an engine that searches components and achieves inheritance inferences. So as a solution, in this work is proposed the use of Ontologies for setting the formal specifications of the terms in the domain, and it is introduced a framework for their development.

Thus, the organization of the paper is as follows: In the first section is presented a profile of the Rule-Base Fuzzy Cognitive Maps. In the second section is pointed out the formal model of the Ontologies. In the third section is outlined the framework for carrying out an Ontology. In the fourth section is described the approach and illustrated its outcomes. In the conclusions are resumed some experiences of the application and it is identified the work to be done.

1. Rule Base Fuzzy Cognitive Maps Profile

The Rule-Base Fuzzy Cognitive Maps (RB-FCM’s), proposed by Carvalho [Carvalho 01], are an approach for modeling the evolution and stability of the entities that compound a domain of study. The RB-FCM’s simulate system dynamics from a qualitative and causal point of view. The main assumption is that: As a result of the activation of a given concept, is perturbed the state of those entities that are causal interrelated with it as effect concepts. A RB-FCM is a digraph, whose vertices correspond to the concepts as nodes, and its arcs to causal relations. Direct causal relationships between two concepts, \( a \rightarrow z \), are sketched by one arc. Indirect relations, \( a \rightarrow b \rightarrow z \), are drawn by paths composed by intermediary concepts that join a cause concept with an effect one.
Feedback relations, $a \rightarrow b$. $\rightarrow z \rightarrow h$, are identified through arcs that depart from one concept to any of its cause concepts. With these kinds of cause-effect relations is trigged a fuzzy-causal inference mechanism that simulates tendencies and final states of the entities of the domain of study.

The concepts are managed as linguistic variables that are instantiated by linguistic terms. According to the nature of the entities, concepts depict variations or levels. For instance, given the entity inflation its variation concept could be increase-high, whilst its level concept is low. Each fuzzy value is stated by a membership function that outcome a fuzzy set.

The fuzzy set owns dimensional properties, as the ones illustrated in Figure 1, which correspond respectively to: a) shape, b) area, c) axis of central mass, d) support set, e) support set length, f) core set, g) core set length, h) interior base length, i) exterior base length, j) interior declination, and k) exterior declination.

The fuzzy set owns dimensional properties, as the ones illustrated in Figure 1, which correspond respectively to: a) shape, b) area, c) axis of central mass, d) support set, e) support set length, f) core set, g) core set length, h) interior base length, i) exterior base length, j) interior declination, and k) exterior declination.

The linguistic terms associated to a given concept are arranged according to their meaning into the context of the universe of discourse (UoD). The UoD corresponds to the x-axis of the plane graph attached to the concept. The absissa is sketched by a scale of discrete points into the range $[-1, 1]$. Thus, fuzzy sets labeled with the highest intensities are allocated at the end of the x-axis. As the intensity of the fuzzy value drops, its fuzzy set is positioned closer to the central point of the UoD as in Figure 2.

Prior begin the simulation, it is measured the initial state of the concepts according to their nature. Afterwards, the initial value is normalized into the scale $[-1, 1]$ in order to assign the point in the x-axis that corresponds it. As a result, it is identified the fuzzy set that reveals the degree of incertitude. In addition, it is estimated the membership degree for the linguistic term, according to the membership function of the fuzzy set. This membership value is a real number in the range $[0, 1]$, which corresponds to the scale for the ordinate y-axis of the concept’s graph, as in Figure 3.

The causal relations are stated by rules, whose antecedent is a cause concept and its consequent is an effect one. Thus, according to the linguistic term that holds the cause concept, a consequent fuzzy value is associated to the effect concept. When the relation imposes a level or a variation value on the effect concept’s state, it is called a fuzzy influence relation (FIR); otherwise it is a fuzzy causal relation (FCR).

The FCR does not estimate the real value of the concept as it does a derivate; only it express the qualitative perturbation that supposes it has occurred. Also, an accumulative effect is achieved on a given concept when it is biased simultaneously by several cause concepts.

Regarding to the FIR a strengthen effect is applied on the effect concept when it is simultaneously instantiated with the same linguistic term by several FIR’s. The mathematical foundations about these two fuzzy relations are detailed in [Carvalho 01].

2. Ontology Formal Model

According to Guarino [Guarino 88] the underlying principles for an Ontology formal model are: Conceptualization and Ontology. Where, conceptualization lays in the Aristotle’s definition given for Ontology that claims: The Ontology is always the same independently of the language used to describe it. The second principle is related to the meaning that the Artificial Intelligence gives to the Ontology, as: An engineering artifact constituted by a vocabulary with a set of explicit assumptions regarding to the meaning of the words.
Wherefore, conceptualization is concerned with the formal structure of reality as perceived and organized by an agent. Whereas, the Ontology is a vocabulary, that depicts the intended meaning of each word. As a consequence, in despite of having different vocabularies, two Ontologies can share the same conceptualization. So an Ontology is a specification of a conceptualization.

The conceptualizations [Genesereth and Wilson 04], are defined as a world structure: \(<D, R>\), where \(D\) is a domain and \(R\) is a set of relations on \(D\) that reflects a particular state of affairs. These relations are considered as conceptual relations defined on a domain space, whose structure is \(<D, W>\), where \(W\) is the set of all relevant states of affairs of such domain. Thus, a conceptual relation \(p^\alpha\) of arity \(n\) on \(<D, W>\) is a total function \(p^\alpha: W \to (2^n)^\alpha\) from \(W\) into the set of all \(n\)-ary relations on \(D\). Given a generic conceptual relation \(p\), the set \(Ep=\{p(w) \mid w \in W\}\) contain the admissible extensions of \(p\). So a conceptualization for \(D\) is defined as a tuple: \(C=<D, W, R>\), where \(R\) is a set of conceptual relations on \(<D, W>\). Therefore a conceptualization is a set of conceptual relations defined on a domain space.

A conceptualization is able to define many world structures \(<D, R>\), called intended world structures, as follows: Let \(C=<D, W, R>\) be a conceptualization for each possible world \(w \in W\). The corresponding world structure to \(C\) is \(S_c=<D, R_c>\), where \(R_c=\{p(w) \mid p \in R\}\) is the set of extensions of the elements of \(R\). So that, the intended world structures of \(C\) is the set \(S_c=\{S_wc \mid w \in W\}\).

Given a logical language \(L\) with vocabulary \(V\), it is defined a model for \(L\) as a structure \(<S, I>\), where \(S=<D, R>\) is a world structure and \(I: V \to D, A.R\) is an interpretation function that assigns elements of \(D\) to constant symbols of \(V\), and elements of \(R\) to predicate symbols of \(V\).

Furthermore, an ontological commitment for \(L\) is the intentional interpretation by means of a structure \(<C, \zeta>\), where \(C=<D, W, R>\) is a conceptualization, and \(\zeta: V \to D, A.R\) is a function that assigns elements of \(D\) to constant symbols of \(V\), and elements of \(R\) to predicate symbols of \(V\). Thus, if \(K=<C, \zeta>\) is an ontological commitment for \(L\), it means that \(L\) commits to \(C\) by means of \(K\), while \(C\) is the underlying conceptualization of \(K\).

Thus, given a language \(L\) with vocabulary \(V\) and an ontological commitment \(K=<C, \zeta>\) for \(L\), a model \(<S, I>\) will be compatible with \(K\), if \(S \subseteq S_c\), and for each constant \(c\), \(I(c)=\zeta(c)\), and for each predicate symbol \(p\), \(I\) maps such predicate into an admissible extension of \(\zeta(p)\). For instance, there exists a conceptual relation \(p\) and a world \(w\) such that \(\zeta(p) = p^\alpha\), \(p(w) = I(p)\). As a result, the set \(I_{k}(L)\) of all models of \(L\) that are compatible with \(K\), will be called the set of intended models of \(L\) according to \(K\).

Given a language \(L\) with ontological commitment \(K\), an Ontology for \(L\) is: A set of axioms designed in a way such that, the set of its models approximates as best as possible to the set of intended models of \(L\) according to \(K\). Thus, an Ontology can depict a conceptualization only in a very indirect way.

Therefore, an Ontology \(O\) for a language \(L\) approximates a conceptualization \(C\), if there exists an ontological commitment \(<C, \zeta>\), such that the intended models of \(L\), according to \(K\), are included in the models of \(O\). Finally, it is said that an Ontology commits to \(C\) if it has been designed with the purpose of characterizing \(C\) and approximates \(C\). A language \(L\) commits to an Ontology \(O\) if it commits to some conceptualization \(C\) such that \(O\) agrees on \(C\). Wherefore from a logical point of view, an Ontology is a logical accounting for the intended meaning of a formal vocabulary.

3. Ontology Framework

The framework for building an Ontology is organized into three stages: Design and Deployment. The first stage is based on Ontology Development 101 [Noy and McGuinness 04]. The second stage uses the FIPA Ontology Agent specifications [FIPA 01] and the Gaia Methodology [Wooldridge et al. 00] in order to manage the Ontology repositories. The third stage corresponds to the integration of the platform for deploying the Ontology based on the Ontology Web Language [OWL 06] and Web Services paradigm [Short 02]

3.1. Ontology Design

In this section are identified the seven steps for designing an Ontology. Furthermore, such steps are illustrated according the Rule Base Fuzzy Cognitive Maps domain, as follows:

1) Identification of the domain and its scope. This task is carried out at responding to competency questions, as: For what it is going to use the Ontology?, and what are the type of inferences to be done?. In this case, the Ontology contains the meta-description of classes and properties for defining the elements of the Fuzzy Cognitive Maps; and it is required a mechanism that deals with multiple inferences.

2) Consider the reuse of existing Ontologies. This task aims for taking advantage of available repositories. However, in this case there is a lack of this kind of Ontologies. So this work offers a contribution for the Cognitive Maps field.

3) Identify the main terms. The task abstracts the most representative classes and attributes in the domain. In the application, there are three types of terms: Meta, as _Id and _Fuzzy_Set; main, as Concept, Relation, and Linguistic Term; and basic, as Membership Function and Fuzzy Rule Base.
4) Definition of classes and hierarchies. This task is accomplished through top-down, bottom-up, or hybrid strategies. As a result, a graphical schema is achieved with the types of terms previously identified, as the one illustrated in Figure 4 where the inheritance relations are drawn by directed arcs. Also, in the Fig. 5 appears OWL code for defining classes as Fuzzy Rule Base. Where the element rdfs: subClassOf, with its sub-element owl:Class owns the attribute rdfs:about that identifies the super-class _Identification.

![Figure 4. Ontology Schema’s Cognitive Map](image)

![Figure 5. Ontology Schema’s Cognitive Map](image)

5) Set the internal structure of the concepts. According to the multiple-inheritance paradigm, the properties are attached to the appropriate class in order to deal with legacies and exceptions. Thus, based on the three types of terms, the properties are defined and inherited, as the OWL code illustrated in Figure 6 where appears the definition of the property area, whose element owl:Class has the attribute rdfs:about that identifies the class _Fuzzy_Set, as the owner class.

![Figure 6. Property Description](image)

6) Characterization of the properties. This task defines facets as value types, cardinality, and default values. In the application, this description is done by means of attributes, i.e., in Figure 6 is set the property area through: rdfs:Cardinality and rdfs:resource, whose respectively values are "#single" and "..#float".

7) Creation of instances. After the meta-data definition, the declaration of the classes’ instances is done with the particular values associated to the properties. For instance, in the application the class Fuzzy Rule Base is instantiated as an Asymmetric_ Relation, as appears in Fig. 7.

![Figure 7. Class instantiation](image)

Finally, the definition of classes and the organization of the hierarchy deal with some issues like: Ensuring that the class hierarchy is correct, number and position of the siblings in a class hierarchy, conflicts stemmed from multiple inheritance, dimension of the hierarchy, and the imbalance produced when a new class or an instance is added to the Ontology.
3.2. Ontology Agent Development

The building of an Ontology Agent is fully detailed in our paper [Peña, et al. 06]. There, it is pointed out that the Ontology Agent administrates the Ontologies, as a middleware between the application agents and the Ontology. The Ontology Agent catches the events and messages that demand some task or any components of the Ontologies. Next, it proceeds to interpret the requests, accomplishes the tasks and returns the answer. The collaboration among agents is achieved by an Ontology-based protocol, which defines speech acts and protocols. Before an Agent sends a message to others, the elements defined are encoded inside a message. When an Agent receives a message, besides interpret its content, requires knowledge of domain to fulfill its job.

The Ontology Agent is available in the Multi-Agent System as a federated service, by its registration in public directories like Universal Description, Discovery and Integration (UDDI). So that, any agent can request to the directory manager for the profile of the Ontology Agent that offers a specific service. The manager interprets the request, seeks the data of the Ontology Agent that matches the constraints, and returns the profile to the requester agent. Afterwards, the user agent encodes messages with the data of the Ontologies and the services required. Next, it sends a message to the Ontology Agent, which achieves the task and returns the results.

Regarding to the development of the Ontology Agent two stages are done: Analysis and Design. In the Analysis stage is gained an understanding of the system and its structure by means of two models: Roles and Interaction. The Design stage depicts how the community of agents work together to fulfill the system goals. The Design outcomes three models: Agent, Services, and Acquaintance.

The Roles Model is composed by role schemas that are stated by four properties: Permissions, activities, protocols and responsibilities. Permissions are the rights granted to the role for managing information resources. Activities are private actions attached to the role without interacting with others. Protocols set functionalities where the role interacts with other roles. Responsibilities set the functionality of the role thru liveness and safety properties. These attributes are triggered respectively, when commit certain environmental conditions and normal invariant conditions are hold.

The Interaction Model depicts the relations among roles in a Multi-Agent System through a set of protocol definitions. These definitions are pattern interactions externalized by messages interchanges. A Protocol Definition consists of six attributes: 1) Purpose of the interaction. 2) Initiator role that starts the interaction. 3) Responder role that achieves the functionality. 4) Input information submitted. 5) Output information generated. 6) Functionality performed during the interaction.

The Agent Model defines types of agents that are composed by sets of roles. So it is possible to package several related roles in a specific agent type.

The Service Model sets the functions that the agent is engaged to achieve after the reception of the request or the occurrence of the condition that triggers the service. Thus, for each service carried out by the agent are depicted the following attributes: Services, Inputs, Outputs, Pre-conditions, and Post-conditions. The services are derived form the Roles Model, the inputs and outputs coming from the Interaction Model, and the pre-conditions and post-conditions depict constraints on services.

The Acquaintance Model draws the communication flow among the agent types, trying to identify potential bottlenecks that could cause problems at run-time.

3.3. Platform

Essentially, two elements are required for deploying the repository and the managing of the Ontology into the Web: The OWL and the Web Services paradigm. OWL presents information and process the content of information. OWL is based on the recommendations of W3C regarding to the Semantic Web, XML language, XML Schema, RDF data model and the RDF Schema. OWL facilitates machine interoperability of Web content providing vocabulary with a formal semantics.

A Web Service is a code that offers specific functionalities to a community of programs through the Internet. The client applications access the service by the use of an interface that invokes a particular activity on behalf of the client. A Web Service provides interoperability, Internet friendliness, typed interfaces, ability to leverage Internet standards, support for any language and distributed component infrastructure.

The facilities required for federating Web Services are organized into five functional tiers: 1) Transport, supported by Hyper Text Transfer Protocol (HTTP). 2) Encoding, described by Extended Markup Language (XML). 3) Message format edited by Simple Object Access Protocol (SOAP). 4) Description of the functionality by Web Services Description Language (WSDL). 5) Publication of services by UDDI.

4. Approach and Outcomes

In this section is outlined the Ontology based Rule Base Fuzzy Cognitive Maps approach through: The description of the Multi-Agent System, the identification of the code used for deploying Web Services, and the operation of the Ontology Agent.

The Multi-Agent System owns three types of components: Web Interface, agents and Ontologies. The user layer is played by a Web Interface that is implemented by an Active Server Page.
In the middle-tier appear the agents. These agents are deployed as Web Services for providing functionalities as: Interface, Ontology managing, encode and decode of messages. The back-end level corresponds to the Ontology repository integrated by the Ontologies and its respective manager. The Web Interface offers the facilities for operating the Ontology application through the windows illustrated in the Figures 8, 9 and 10.

Figure 8. Ontology Class definition

Figure 9. Ontology Class definition

Figure 10. Instance definition
In the Figure 8 appears a window devoted to insert a class definition, e.g., a *Fuzzy_Rule_Base* definition. Whilst, in the Figure 9 is stated the definition of a property called *fuzzy_rule*, with text and multiple as range and cardinality attributes. Finally, in the Figure 10, is inserted an instance, called *Causal Fuzzy Rule Base* for the *Fuzzy_Rule_Base* class, where the value of *teaching → learning*, is attached to the *fuzzy_rule* property of the instance. This application can be referenced thru: [http://www.wolnm.org/SEBW/ME/AO/waInterface/wfInterface.aspx.cs](http://www.wolnm.org/SEBW/ME/AO/waInterface/wfInterface.aspx.cs).

An agent interface is attached to the Web Interface through the code edited in Figure 11. The section defines the Web Service corresponding to the agent interface, in the language Microsoft® C®#®. Whilst in the Figure 12 is shared the Web Service reference.

```
// Web Service Class Declaration
namespace wsInterfaceAgent {
  public class wsInterfaceAgent :
    // Web Service Method Declaration
    [WebMethod]
    public string wmInterfaceAgent(Input-parameters, out output-parameters) { .... return value }

Figure 11. Web Service Declaration
```

```
// Web Service Reference
namespace waInterface { public class WebForm1 : System.Web.UI.Page }

// Namespace: host;
// class: wsInterfaceAgent is the Interface Agent;
// Instance: gInterfaceAgent
host.wsInterfaceAgent gInterface = new host.wsInterfaceAgent();

// Web Service invocation;
// Web Service method: wmInterfaceAgent
gInterface.wmInterfaceAgent
  (Input-parameters, out output-parameters);

Figure 12. Web Service Reference
```

The communication among agents is expressed by messages edited in XML. The schema is based on an attribute-value style, distributed along a structure of elements and sub-elements with tags and values. The request schema is depicted in the Figure 13.

```
<?xml version="1.0" encoding="utf-8" ?>
<messageMAS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <request>
    <Header_Message>
      <Sender_Agent>wfInterfaceAgent</Sender_Agent>
    </Header_Message>
    <Body_Message>
      <Command>Query_if</Command>
      <Head_elemento>Class</Head_elemento>
      <Body_Note>
        <Note>Request</Note>
      </Body_Note>
    </Body_Message>
  </request>
</messageMAS>
```

**Figure 13. Message Schema**

The code owns a *Namespace* and a root labeled by the *request* tag. The three main elements are: *Header_Message, Body_Message, and Body_Note*. The elements have sub-elements as: *Sender-Agent, Command, and Note* respectively. In order to access the elements of the message, the decode agent navigates thru the structure. Besides, it interprets the elements and sub-elements, and recovers the values. These components are stored in a data structure declared in a class.

The Ontology Agent catches, interprets and forwards the requests to the Ontology manager of the Ontology demanded. Also, the Ontology Agent supervises the process, gets the outcome and sends the results to the requester agent. The code for processing a *request* and *command* appears in the Figure 14.

```
// Call to decode the request message received
void processMessage(){
  if (rDecodeAgent.RMessage.Equals("request"))
    processRequest();
  // Identifies the command of the message in order
  // to process it
  void processRequest() {
    if (rDecodeAgent.RCommand.Equals("Insert"))
      pInsert();
    else if (rDecodeAgent.RCommand.Equals("Query_if"))
      pQuery_if();
    else if (rDecodeAgent.RCommand.Equals("Insert_Instance"))
      pInsert();

Figure 14. Message process and request methods
```
The Ontology is encoded in OWL by the use of three types of elements: Classes, properties, and instances. These elements are identified by the tags: owl:Class, owl:DatatypeProperty, and the name of the class, i.e., _Fuzzy_Set.

5. Conclusion

The Ontology offers a sound and flexible support for setting the semantics of the structure and vocabulary of the elements of a domain. Thus, in this paper was introduced the contribution of the Ontologies to the field of the Rule Base Fuzzy Cognitive Maps. Based on their formal profile of the Ontologies and Cognitive Maps, was stated a framework for building an Ontology administration environment.

The proposal sets the process for developing an Ontology. Also, it states the specifications and stages for deploying an Ontology Agent. Furthermore, were identified the two main resources for federating the Ontologies services by means of OWL and Web Services. Finally, were sketched some outcomes and the code used in the application.

As further work, it is considered the enhancement of the Ontology for describing the cognitive, personality and learning preferences of the students. As well, it is planed to develop a mechanism that automatically generates the Fuzzy Cognitive Maps based on the support of the Ontology.

ACKNOWLEDGMENT

First author states that this work was inspired in a special way for my Father, my brother Jesus and my Helper as part of the research projects of World Outreach Light to the Nations Ministries (WOLNM). Also this work was partially supported by UPIICSA-IPN, CIC-IPN, COFAA-IPN, COTEPABE-IPN, CONACYT and Microsoft México; Through grants: CONACYT 182329, IPN: SIP-20060064, COFAA DEBEC/138/06, COTEPABE PL-430/05.

REFERENCES


