I.J. Information Technology and Computer Science, **2016, 10, 72-81**
Published Online October 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2016.10.09

A Review on the Knowledge Representation Models and its Implications

Hepsiba Mabel V
SCSE, VIT University, Chennai, TN, India
E-mail: hepsiba.mabel@vit.ac.in

Justus Selwyn
SCSE, VIT University, Chennai, TN, India
E-mail: justus.s@vit.ac.in

***Abstract*—**Data and Information has seen exponential growth in the past few years which has led to its importance in processing it in the creation of knowledge. Representing knowledge in a required format is the need for the building a knowledgebase (KB) for Expert Systems. In this paper we carried a survey on the knowledge representation models that will help us choose a suitable model for designing and developing a KB. A detailed study is conducted on six models and comparison of the models on some non-functional attributes are carried out to enable knowledge workers to decide on the model selection.

***Index Terms*—**Knowledge representation, knowledgebase, intelligent systems, object-relational data modeling.

I. INTRODUCTION

The exponential growth in data, information, databases and repositories has created a demand for newer techniques and tools that can transform data into useful information and knowledge [15]. In the earlier days, we were in need of generating tools, techniques and models to acquire information from various internal and external resources. Now we have plenty of techniques to acquire data from everywhere and anywhere through several means and ways. Data and information are available in various forms and accessed and manipulated according to the systems' requirements. However, intelligent, expert and decision making systems demand higher level of data and processed information, which the world calls as knowledge.

Hence our intension is to retrieve the processed information 'knowledge' from the volumes of data repositories and storage systems. Knowledge retrieval cannot be directly done from the data or information. Therefore, the data and the information need to be represented in some model, so that knowledge acquisition is made possible.

Our concern here is how to represent the knowledge using symbols or tree-like structures and how to get it manipulated automatically. Here came the generation of OSM, CSM and CFRM, the models that we have generated in our earlier research works. Though we have a plenty of knowledge representation models [11], [15], [16], it is mandatory to review the models in order to analyze its features, functionalities, approaches and functional relevancy with other models.

A. Scope of the work

The representation models that we have proposed already meets 1) an object-based model that will represent simple, atomic concepts in the knowledgebase, 2) a conceptual graph based model that will handle the complexity of the concepts and its relationships, and 3) a special model of knowledge unit, which can handle dynamic processing of smaller, simpler KUs. Computations on these KUs from collection objects, random storage & retrieval from the working memory or secondary memory are efficiently handled by this third model. These knowledge representation models are evaluated for its performance, efficiency in computations and their reliability.

Formalizing and Validation of these models will provide an authenticity for the knowledge users and workers to very well accept them in their day-to-day practice of developing better knowledge based intelligent system

Research works on knowledge engineering and knowledge management have taken new heights in the past decade, resulting in so many smart phones/devices, intelligent expert systems and many research institutions and labs in particular for KM have been set in India and other countries. The scope of this Knowledge Era will continue to raise many innovative inventions in the next few decades, and works like ours on the integration of such pieces in KM will add treasure for the BOK of KE and KM.

These representation models will be the functional units for each layer in the knowledgebase. At every layer in the knowledgebase the represented knowledge units will serve its functional specifications.

B. Outline of the paper

Section II details on the various knowledge representation approaches that have evolved over the years. Section III gives a detailed review on the knowledge representation models. Comparison on the models is given in Section IV and with a brief conclusion on Section V.

II. Knowledge Representation Approaches

Knowledge units are the basic building blocks in the knowledgebase (KB) and they need to be represented in a more formalized models. In this section, we will present an overview on the forms of knowledge along with their definition as representation models.

A. *Knowledge Concepts*

Making decisions is the most important and complex task in our lives. It has been given to humans alone naturally, and human's decision all through ages have led us to where we are now – relations, feelings, intelligence, independence, and what more, advancement in technology. Decision making is preceded by so many complex activities like exposure to understanding, comprehending, inferring, thinking and finally concluding to take decisions [1]. These activities involve intelligence and it has been provided by nature to us, so we voluntarily do these activities to decide on something. To arrive at a decision, usually there are more than one choices, or paths to process, infer and conclude. Hence we process several concepts, situations, parameters and many other external and internal factors to conclude on a decision [2, 3]. The many things that are stored and processed are considered to be true beliefs, used in resolving to a conclusion or decision. These true beliefs are called Knowledge [10].

B. *KU: Building Block*

Conventional system development approaches will involve 1) Processing of data and information, 2) Algorithms 3) Control Sequence 4) System understanding 5) Expertise of the developing team in the technology [4, 5]. However in designing and developing an expert system in a domain, and making it work as effective or more effective than human, is to first understand the design requirement of such system, how it should work, and the deliverables. The main conceptual, working source of an ES is Knowledge [6]. Algorithmic procedures don't handle knowledge, and hence symbolic representation of knowledge is required. Knowledge is represented as symbols, letters words, objects, processes, instances, relationships etc. Instead of simply representing data in a datastructure as in conventional programming, knowledge based ES requires a specialized structure, which we will name as Knowledge Unit (KU) [7]. A KU has the advantage of representing a concept along with its data, instances, facts, rules, associations, relationships, other relevant concepts. The most important thing about KU is, it itself is returned to the user as Knowledge, as results of resolutions and conclusions.

C. *Forms of Knowledge*

Based on their existence in the real-world system, knowledge is classified into three major forms: Tacit Knowledge, Explicit Knowledge, and Implicit Knowledge. Though there are several other forms proposed over the years, these three hold good for knowledge representation.

1) Tacit Knowledge

The inexpressible form of knowledge that resides in the human brain is called the tacit knowledge. This is the original form of knowledge, and is intellectual in nature. It is obtained mainly through learning and experiences. Its form is unstructured and cannot be generalized to a knowledge domain or ontology [8, 9]. The key challenge is to acquire this form of knowledge through knowledge acquisition, capturing and knowledge sharing. Consider for example: A driver says "I know to drive a car", is a claim that he has the knowledge in driving a car. If he were asked to share it, he can express it in a bulleted set of steps, and one cannot drive a car by following these steps. Hence the knowledge of the driver to drive a car is within him implicitly, and it cannot be expressed. Another thought is, "knowing to drive a car" raises another question, how much does he know, what is the degree of knowing? These things are impossible to be expressed in any form, and it is called the tacit knowledge.

2) Explicit Knowledge

A form of knowledge that can be expressed in any of the format is called explicit knowledge [10]. Its form is structured and can be represented using a language construct and can be generalized in an ontology or in a domain. The explicit form of knowledge can be expressed in many ways, including plain text, documents, spread sheets, databases, web-forms, images, audio, video, tweets, posts, short-messages, and the list grows. However, the expressed knowledge in these forms is categorized into structured and unstructured. Though unstructured, they can be processed and knowledge can be acquired. For example: an audio file of a Lecture can be processed and its contents can be organized into a structured form resulting in a collection of concepts/knowledge. Similarly, videos, images, tweets and posts, can be processed and knowledge can be acquired to be expressed in a required format. Explicit knowledge sharing between knowledge-based systems is deemed lossless, due to the representability. Knowledge representation and reasoning are dealing with explicit knowledge [11].

3) Implicit Knowledge

The existence of this knowledge in the system is unknown. However, it is used randomly without knowing its involvement in the system. Mostly, implicit knowledge is inexpressible, and sometimes explicit knowledge that remains dormant are also termed implicit [12]. For example: "lifting the accelerator and holding the clutches while shifting gears in a vehicle" is an implicit knowledge. Since this action happens voluntarily it is categorized as implicit. This form of knowledge cannot be processed to be represented, hence left untouched.

However, finding the availability of knowledge in the knowledgebase is a key challenge.

These forms of knowledge keep revolving in the knowledge management cycle within an organization. The tacit knowledge in the human brain gained over learning and experience is transformed into explicit knowledge through the process of documentation and other forms of knowledge representation [1]. The implicit form of knowledge is within the tacit form, but can be acquired and extracted through entailment procedures.

D. Concrete knowledge, and dependent knowledge

The concept of knowledge representation is as philosophical and complex as that of knowledge. Representation is a relationship between two domains: one being the concrete or base or immediate domain and the other takes the second place of depending domain. For example: "Brilliant Student" is a representation of a concept. Here, obviously "Student" is the concrete, immediate concept and "Brilliant" is a depending concept on "Student". The definition of "Student" will be concrete, correct, unambiguous, and immediate to access. But the definition of "Brilliant" takes a relevance or dependence on the concept to which it is being related. Hence the representation "Brilliant Student" carries semantic and pragmatic in the given instance.

Representation is conventionally done by symbols and shapes. "An index finger pointing to right in a restaurant and written WASH" represents the 'hand wash area' is at the right side. Examples of symbols representation in road rules are explanatory. Similarly representation of knowledge involves *formal symbols*, *propositions*, *subjects* and *entities* in the given representation ontology or domain.

E. Logic Representation Models

1) First-Order-Logic

It's a knowledge representation technique which has three notions for its valid representation named Syntax, Semantics and Pragmatics. It has logical and non-logical symbols. Logical symbols include punctuation, connective and variables. Non-functional symbols include functional and predicate symbols. Knowledge Units are represented as functional symbols and predicate symbols. The representation with these symbols will be meaningless if it is not interpreted.

A FOL representation is added meaning as a function of interpretation of function and predicate symbols. An interpretation T of FOL is a pair of D x I where D is the domain and I is the interpretation mapping of non-logical symbols to the domain D. For every predicate symbol P of n-ary, then I(P) ϕ +*D*H *D*H *D*… ,

These interpretations hold good and consistent for the given identified domains, and the mapping will hold true for that domain D. In order to have it more real-world, the existence of these predicates, interpretations and the existential property of the domain has to be explicitly represented. In short, the semantic of the predicate or function symbols in a given instance is represented. For example consider: ∀ (x).Dog(x) ∈ Mammal(x)

The existential identity of x is the instantiation of the predicate symbol Dog and Mammal. If Dog ("Jimmy") is in our knowledgebase (KB), then it can be logically concluded that Jimmy is a Mammal.

2) Working Memory & Framesets

A knowledgebase (KB) is viewed as a collection of facts and rules. The facts are the atomic representation of concepts, and the rules act upon these facts, to make it meaningful and help in resolutions. If x and y are symbols, then Parent(y, x) leads to a resolved conclusion ChildOf(x,y). These designs of KB are preferred in some Knowledge Management System, where facts and rules are given prime consideration and they are called as Production systems (PS). The facts and rules are represented as framesets in a PS. A collection of these frames are stored in a temporary volatile memory called Working Memory (WM). WM supports the knowledge-based system (KBS) as an intermediate between the KB and the front-end application. The entries in a WM are called working memory element (WME), and they represent the facts and the rules in a more consistent manner.

The rules acting over the facts are called production rules, and they are of the form

IF conditions **THEN** actions

The conditions are tests that take the current state of the KB, and the actions are set of modifications done on the KB. The basic three operations of a PS are: 1) recognize rules that are applicable, 2) resolve the conflict among the rules found in first step, and 3) update working memory by performing the consequent actions. The basic building blocks of a working memory are the working memory element (WME), and it is of the form:

<type attribute1:value1 attribute2:value2 ….. >

Where type is a concept or a knowledge unit, attribute is the property of the concept and value is assigned to the attribute. An example of a WME is:

<person name:John age:23 eyecolor:blue…>

<student roll_no:11PHD08 registered_course:"Research Methodology"…>

<staff emp_id:195613 department:"Computer Science" specializat:on:"cloud computing"… >

Production Rules are also applied in the WME as follows: <student CGPA: {<8 ϖ >5 }>

This WME execution returns students whose GCPA score is between 8 and 5. The three actions that are applied on a WME are: ADD, MODIFY and REMOVE elements of WM. These actions are added to the value in the WME wherever required.

Framesets are a representation of objects or a WME in a working memory. Each WME or a concept is represented as a frame. A frame is of the form:

(Frame-name
<:slot-name1:filler1>
<:slot-name2:filler2>
...)

Frame names are the name of a concept or knowledge-unit, and the slot-name is the attribute and the filler are the values assigned to the attributes. All represented frames are connected through two types of relations: IS-A and INSTANCE-OF.

The frameset representation involves action parts which are added as fillers to slots, they are: IF-ADDED, IF-NEEDED and IF-REMOVED. If a value is added to a frame as a slot, then the actions or computations following IF-ADDED are executed, if a value or a set of values need to be added or modified then the actions or computations following IF-NEEDED are executed, and if a value or a slot is removed, then the actions or computations following IF-REMOVED are executed.

3) Description Logic (DL)

In addition to these above discussed representation logic, Description Logic (DL) has the power to represent nouns and noun phrases. DL implements a knowledge unit as a three-term representation, concept, roles and constants. Atomic concepts such as Parent are given as d, roles such as Father are given as r, and constants such as “John” as c. John is a Father, and hence he is a Parent. Using the keywords of DL, some examples of DL are

Pediatrician = [AND Doctor [EXISTS 7 YearsOfPractice] [FILLS SpecialistDegree Pediatrics] [ALL Degrees [FILLS Recognized AIMA]]

Surgeon = [AND Doctor [EXISTS 4 YearsOfPractice [FILLS SpecialistDegree Surgery] [ALL Degrees [FILLS Recognized AIMA] [FILLS MinCGPA 7.5]]]

These two definitions of the concepts Pediatrician and Surgeon can be entailed to form a new concept PediatricSurgeon, whose definition will go as follows:

PediatricianSurgeon = [AND Doctor [EXISTS 7 YearsOfPractice [FILLS SpecialistDegree Pediatrics] [FILLS SpecialistDegree Surgery] [ALL Degrees [FILLS Recognized AIMA] [FILLS MinCGPA 7.5]]]

Concept-inheritance is same as that of object-oriented programming, entailments and interpretations are same as that of FOL. Entailments in FOL are done after the given sentence or a formula is converted to a CNF form, similarly in DL a concept definition has to be normalized before it is entailed to get a new concept.

F. The Need for Representation Models

Now these knowledge processes are used in knowledge representation techniques. Capturing information of structured, unstructured and semi-structured, manipulating them to acquire knowledge through representation techniques are really challenging. However, the steps involved in knowledge representation are:

1. Processing the higher-level data or information (tokenizing)
2. Ontology-based clausial classification (Clauses entailments)
3. Logical validation of the information (FOL)
4. Defining the logic and the knowledge unit (DL)

Hence all the above discussed knowledge representation approaches are used in designing and building knowledgebase.

III. Review of Knowledge Representation Models

Having said the intricacies of the representation models and their context in the knowledgebase, this section presents a detailed review on the types of six major representation models. These models can be perceived in the context of its application and implementation in building a KB.

A. Graph based Representation Model for GIS

To represent the spatial and non-special data, a graph based representation model is prescribed. This model can then be used to generate the data set comprising both the types of data along with relationship among them. Then a data mining technique can be used to improve results by considering the both the spatial and non-spatial data along with spatial relationship represented in the graph-based representation model [15].

Each node will be assigned for an object, in which the attributes describing the object, relationships with other objects will be defined. To improve the knowledge extraction, all these elements should be considered for an object. These attributes and relationships will be really helpful to analyse the spatial elements. In this model, collection of vertices and edges will be used to represent the spatial objects along with its attributes. The total number of vertices used in the graph can be calculated by

$$n+\sum_{i=1}^{n} num-of-attr-per-obj\left(\mathrm{n_i}\right)$$

where n is the total number of spatial objects used in the collection.

Similarly, each edge will represent the binary relationships among the objects. The total number of edges used in the graph can be calculated by,

$$n+\sum_{i=1}^{n} num-of-attr-per-obj\left(n_i\right)+num-of-rel-btwn-objects$$

This type of model helps in representing the objects and its relations with other objects and gives a great impact in the result of data mining processes. As this model is designed only for GIS, it will be used to identify the most important characteristics of geometric objects located at some distance from some specific risky regions.(eg: to analyse whether they will withstand in the atmospheric conditions, to analyse the evacuative routes etc).

B. *Conceptual Graphs (CG)*

This is a connected bipartite graph in which number of concepts and its relation with other concepts can be represented [9][11]. In Conceptual graph, the concepts will be represented by boxes and the conceptual relations can be represented using ellipses. Conceptual nodes will have arcs to conceptually related nodes thus creating the bipartite graph. Conceptual nodes represent the concrete objects, e.g Teacher, Student, etc. To represent a relation of arity of an n conceptual relation node, it needs n arcs.

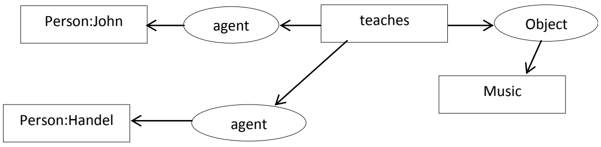

Fig.1. Model of a CG

Here each concept node represents an individual and specifies the type of each individual. If an object is not known, another identifier can also be used in the place of the name of the object. The unique identifier can be generated by comprising a hash symbol followed by a numeric value. In the above example if suppose, we don't know the name of the teacher, we can give him an Identifier (eg: #50603) as his identity. This kind of identifier in the graph is called individual marker [11].In addition to this type of identifier, another one named 'generic marker' can also be used with an '*' symbol to represent an unspecified individual [11].

C. *Nodes of Knowledge Representation (NoK)*

This is the model designed for representing the knowledge given in a sentence or in a group of sentences. This model takes its input from documents, databases, encyclopaedias, business processes and business documents. This model uses two basic elements, nodes and links. To represent different contexts, collection of context nodes is used.

A special node called process node is used in this model to represent the knowledge described in the sentence. Moreover an array of process nodes can also be used to represent the knowledge expressed in a sequence of sentences. The objective of this model is generating a knowledge network for text based sentences [16].

Ex: *John teaches music*

According to the context node 'teaches', "The teacher teaches music" is true. According to the process node 'teaches', "John teaches music" is true. But here we have a connection with the process node 'conducts' which gives the answer to the question "Why does John conduct the exam?". According to the process node 'conducts' the statement "John conducts exam" is true. Therefore when we combine all the previous statements into one, we get the answer "John teaches music because he conducts the music exam". The graphical representation of this example could be translated to the following statements.

"The teacher teaches music". "John is a teacher"

"John teaches music as he conducts the examination"

"Studying of violin can be categorized as studying of music", "Violin belongs to music group"

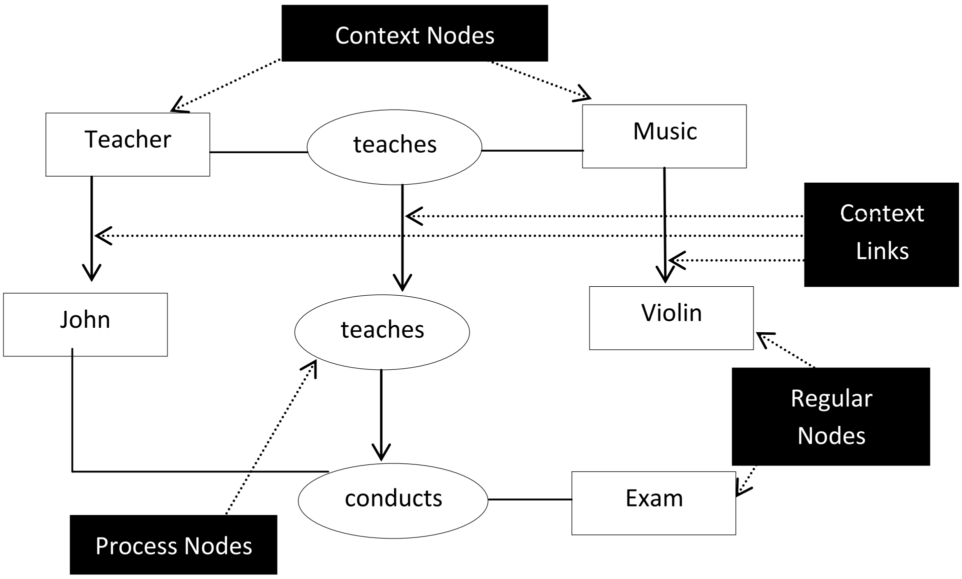

Fig.2. NoK Example

D. *Object Model: Object Structured Model (OSM)*

The object structured model is defined as follows:

The OSM modeling environment is an object-based ontology with knowledge acquisition/representation activities that are widely used for domain modeling and simple KU representations. OSM can be represented in four labeled structure

OSM = {knowledge, <c, s, f, a>} where c-classes, s - slots, f - facets, and a - axioms.

Classes are the representation of the context. Slots describe properties or attributes of classes. Facets describe constraints associated with a slot. Additional constraints to a relation can be specified with Axioms. An OSM knowledge base includes the ontology and individual instances of classes with specific values for slots. Slots express the properties of classes and instances, such as name of an employee.

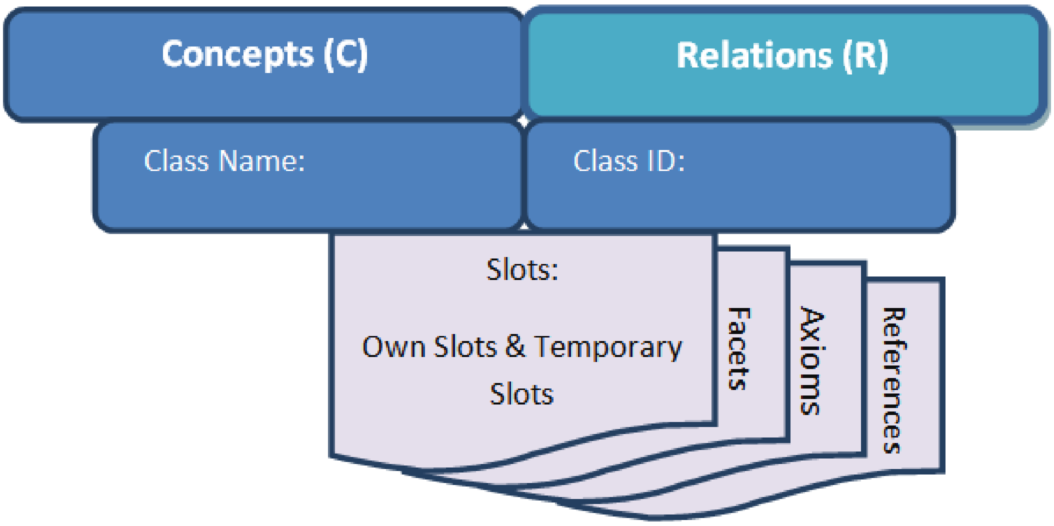

Fig.3. Functional Model of OSM

The two slots: own slots describe properties of an object represented by that frame and temporary slots describe properties that an instance of a class can possess. The number of values that can be associated with a slot, restrictions on the value of the slot can be specified using Facets. A class can acquire own slots only by being an instance of a meta-class that has those slots as temporary slots.

Consider for example: "John teaches music". This is a concept. 'John' is a human entity and can be represented as Person_t. However as of in OSM, John is an object (class instance). 'Music' is another entity which literally cannot be defined in this context. Hence, we represent it also as an object (class instance). The representation of this concept in OSM is given in table 1.

E. Conceptual Model: Concept-Structured Model (CSM)

Definition: A conceptual graph representation of a knowledge unit is a paired structure with concept and referent. Concept has the context of the knowledge content and referent has the links to the conceptually related knowledge unit.

This is a widely preferred and applied model for knowledge representation in the building of knowledge management systems. Since this representation is in close association with the natural language, practitioners choose this model for representing the entities conceptually. Knowledge is perceived as a concept, and it is strongly argued and agreed that any knowledge could be represented in the notion of concepts, because knowledge itself is a conceptual understanding of the human brain [13].

According to John F Sowa [11], "A finite, connected, undirected, bipartite graph with nodes of one type called concepts and nodes of the other type called conceptual relations". These concepts are the nodes and the conceptual relations are the arcs in the conceptual graphs.

Table 1. OSM construct for a concept

Class Name	John_teaches_music
Class ID	XXX_XXX_XX
Concepts	John→Person_t, music→skill_set_t
Relationships	Teaches, knows, trains
Slots	Person_t→name, Person_t→location Person_t→available_timings Skill_set_t→theory Skill_set_t→instrument Skill_set_t→experience
Facets	Person_t→available_timings *of* Person_t→name *in the* Person_t→location Skill_set_t→experience of Person_t→name *in teaching* Skill_set_t→theory *and* Skill_set_t→instrument
Axioms	Skill_set_t→instrument *includes* piano, violin, guitar *only*
References	John→teacher, artist, performer music→Classical, carnatic, gazal, Hindustani

Conceptual graphs address in terms of concepts and its attributes. A concept can be an entity, event or an action. Every concept has its own attributes and is instantiated with instances.

The functional model of CSM given in Fig. 4is arrived at after a careful study of the dynamisms of conceptual graphs. Each node in the given conceptual graph G will be an instantiation of this functional model. In order to instantiate this CSM model, we identified an object-relational (O-R) model presented in [17].

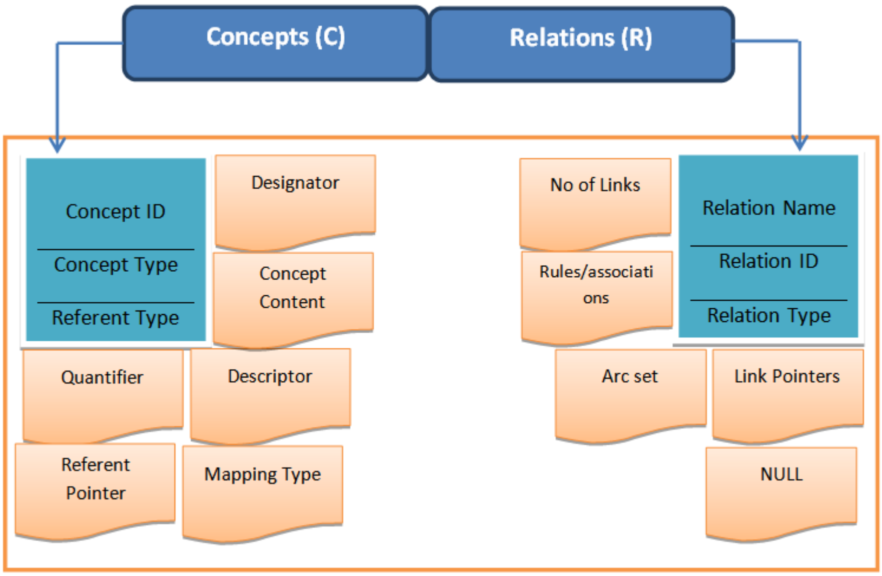

Fig.4. Functional Model of CSM

Object relational model is chosen because of its three-fold functionality which is the basic requirement for our functional model of CSM. The three fold structure of an object relational model is given as:

O-R Model = {<attributes, functions>, <relations>}

The O-R model is a powerful implementation platform which combines the traits of object-oriented programming and relational concepts programming concepts of RDBMS [14]. Hence it is a class as in OOP, but added to it is the 'relation' part that relates the classes conceptually. Hence the implementation of the concept is to represent it as a data structure and get stored in a relational database.

F. Storage Model: Concept Function Relation Model (CFRM)

Definition: The concept part of every KU is associated with a function, and coupled with conceptual relation with other representation structures. This model holds the basic concepts that are collected and stored in collection class_types, which are made relational to other class_types.

By stating this model, we attempt to design a platform onto which the already proposed models OSM and CSM can reside and get stored in the lower storage architectures. This model proposes three-layered architecture which binds the OSM objects and the CSM objects together. This binding will enable storage and retrieval of the objects more efficiently than when they are managed as standalone objects. The three phases of CFR model is given below.

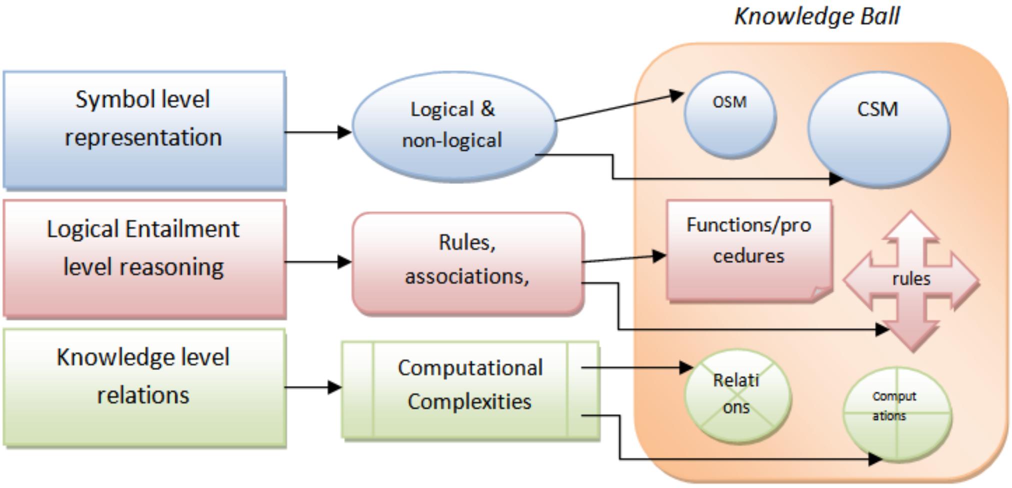
Knowledge Ball
Symbol level representation
Logical & non-logical
OSM
CSM
Logical Entailment level reasoning
Rules, associations,
Functions/procedures
rules
Knowledge level relations
Computational Complexities
Relations
Computations

Fig.5. Functional Model of CFRM

A knowledge unit before it is stored in a schema will have to be modeled in this way as shown above, where the three levels of representation are necessary. When a concept is acquired, it represents only a data or a set of relevant data, or sometimes trivial information. This concept is added with rules, associations and axioms in the forms of functions and procedures, giving semantic meaning to the acquired concept. Finally, the semantic concept is stuffed with relations to make it pragmatic, to lead to conceptual connection with other conceptually relevant, related similar or dissimilar concepts. The computational complexities in storing these KUs and their retrieval are also included at the knowledge level.

Consider the following example:

"*Every Student has an option to select a minimum of 5 subjects and a max of 7 subjects in a semester*"

Student is a concrete knowledge in this statement and covers a wide variety of their categories which includes, undergraduate, postgraduate, doctorates and etc. the rule is applied to the Student KU. The rule is the 'Option' of choosing the subjects. Now subjects are collection types having 100s of subjects being offered. Out from the subject pool each student will have knowledge collection types, if and only if the collection has minimum of 5 subjects and a maximum of 7 subjects in a given instance, a semester, for a given Student. This can be represented in O-R modelling as shown in Table 2.

These representation formulas and terms can be easily codified to class_types in object-relational databases (ORDB) without any ambiguity or incorrectness. The following class_types can be created corresponding to the representation symbols given Table 2. This whole bunch of representation is the object that we are trying to store in the knowledgebase.

Create class_type Student_t	x
Create collection_type subject_Pool_t	f(x)
Create class_type subject_t	s
Define rule subject_Pool_t→minimum	Q
Define rule subject_Pool_t→maximum	R
Define procedure_t subject_selection	P
Create/Define relation Student_t→ subject_Pool_t	T

Table 2. An example of CFRM representation

	Symbol level		**Entailment level**		**Knowledge level**	
	Logical	**Non-Logical**	**Rules**	**Functions**	**Relations**	**Computations**
Interpretation	Student	collectionOfSubjects subjectPool	Option minOption(5) maxOption(7)	selectionFromSubjectPool()	Student→subjectPool subjectPool→Program	Mapping, storage & retrieval of objects from pools for every student
Representation	$\forall(X)$, $\exists(X)$	f(s_1, s_2,…, s_n)	$\exists(X)$.[Q(f(s)) ϖ R(f(s))]	$\exists(X)$.P(f(s_1, s_2,…, s_n))	$\forall(X)$. T(f(s_1, s_2,…, s_n).T(p))	Algorithms, procedures' complexity
Symbols explanation	x is Students	f is a non-logical symbol representing the collection of subjects {s_1, s_2,…, s_n}	Q is a rule to select minimum, R is a rule to select maximum	P is predicate symbol that represents the selection	T is the predicate symbol that represent the relations	-

Table 3. Comparison of the knowledge representation models

	GIS	**CG**	**NoK**	**OSM**	**CSM**	**CFRM**
KU Forms	Explicit: YES Implicit: NO	Explicit: YES Implicit: YES	Explicit: YES Implicit: NO	Explicit: YES Implicit: NO	Explicit: YES Implicit: YES	Explicit: YES Implicit: NO
Implementation	Imaging data modeling	Volatile structures	User-defined Data Structures	OOP, Framesets	Object-Relational Data Modeling	Object-Relational Data Modeling
Usability	In spatial systems	in logical entailments	In relational logics	In codification	In logical entailments	In storage & retrieval
Scalability	Widely scalable	Widely scalable & flexible	Not scalable	Scalable	Widely scalable	Exponentially scalable
Shareable/Reuse	Partial	Complete	Partial	Complete	Complete	Complete

IV. Comparison of Representation Models

After the study of these models for using them in the development of Knowledgebased systems, the comparison of these knowledge representation models could provide a deep insight in designing a knowledgebase for Expert Systems. We considered the six representation models: Graph-based model (GIS), Conceptual Graph (CG), NoK, OSM, CSM, CFRM.

Comparisons of these representation models are based on some of non-functional attributes of these models. Though these models are validated empirically and mathematically in their original works, we are trying to study them in the terms of its applicability in designing a knowledgebase. Some of the important attributes of comparison we have chosen are their KU forms, platforms of implementation, usability in systems, scalability & flexibility in terms of growth and adaptability, knowledge sharing and reuse of the models.

The research on knowledge representation models since the late 1990s has given GIS, CG, NoK, Framesets, Petrinets, and several other tools too. In the early 20th century, enhanced works on object-based approaches can be found. From object-orientation data modeling evolved the powerful data model that combined object-orientation and relational database concepts, called object-relational data (ORD) modeling. The features of OOD like abstraction, inheritance, polymorphism, encapsulation, and the features of relational data modeling like normalization, integrity, and relational aspects are combined. This ORD formed basis for many other knowledge representation models like the one we proposed earlier, OSM, CSM and CFRM.

V. Conclusion

Presently, we can find works going on knowledge representation in terms on Bigdata representations, and data and information flowing from Internet of things (IoT). Now knowledge workers have greater challenges to deal with these vast, variety data and information flowing from many directions into our repositories. IoT is another wide area, where we have greater and real-time challenges to deal with data from electronic devices, those which could be unstructured or semi-structured.

This work on the review of knowledge representation models will lead knowledge workers and researchers to choose from these models or to propose newer models to manipulate and process data and information to acquire, extract, represent and codify knowledge. This ultimately leads building KB that can handle bigger data and information for Expert Systems.

Refernces

[1] Lu. J., M.A. Quaddus and R. Williams, "Developing a Knowledge-Based Multi-Objective Decision Support System", Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000.

[2] Molina. M., "Building a decision support system with a knowledge modeling tool", Journal of Decision Systems, Lavoisier, 2006.

[3] Pomerol.J., P. Brezillon and L. Pasquier, "Operational Knowledge Representation for Practical Decision Making, Proceedings of the 34th Hawaii International Conference on System Sciences, 2001

[4] Bartak, R. (1999). Expert Systems Based On Constraints. PhD thesis.

[5] Hart, A., "Knowledge Acquisition for Expert Systems", New York: McGraw-Hill, 1992.

[6] Fensel,D., J. Angele, and R. Struder, "The Knowledge Acquisition and Representation Language KARL", IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No. 4, July/August 1998.

[7] Alagarsamy K., Justus S and Iyakutti. K., "Implementation Specification of a SPI supportive Knowledge Management Tool", International Journal IET Software , Vol. 2, No. 2, pp. 123-133, April, 2008.

[8] Holsapple C, Joshi K., "A Formal Knowledge Management Ontology: Conduct, Activities, Resources and Influences". Journal of the American Society for Information Science and Technology, Vol. 55, No. 7, pp. 593–612, 2004.

[9] Jérôme. Nobécourt, Brigitte Biébow, "Md ωπ: A Modeling Language to Build a Formal Ontology in Either Description Logics or Conceptual Graphs", Knowledge Engineering and Knowledge Management Methods, Models, and Tools, Lecture Notes in Computer Science Volume 1937, pp 57-64, 2000.

[10] Ronald J. Brachman and Hector J.Levesque, "Knowledge representation and reasoning", 2nd edition, Elsevier publications, 2004.

[11] John F. Sowa, Conceptual Graphs. Handbook of Knowledge Representation, 2008

[12] Marvin Minsky. A framework for representing knowledge. In John Haugeland, editor, Mind Design, pages 95–128. MIT Press, Cambridge, MA, 1981

[13] Inui, "Mechanisms of Action Generation and Recognition in the Human Brain", Proc. Second International Conference on Informatics Research for Development of Knowledge Society Infrastructure, pp. 45-52, 2007.

[14] ISO / IEC 9075 Standard, Information Technology – Database Languages – SQL: 2003, International Organization for Standardization, 2003.

[15] Manuel Pech Palacio, David Sol and Jesus Gonzalez, "Graph-based Knowledge representation for GIS data", Proc.Fourth Mexican International Conference on Computer Science(ENC'03),0-76895-1915-6/03, 2003 IEEE

[16] Alen Jakupovic, Mile Pavlic, Ana Mestrovic and Vladan Jovanovic, "Comparison of Nodes of Knowledge Method with other graphical methods for Knowledge Representation", MIPRO 2013, May 20-24, 2013,Opatija, Croatia

[17] Justus S, Iyakkutti, "An Empirical Validation of the suite of metrics for Object-relational data modeling", International Journal of Intelligent Information and Database Systems, Vol 5, No, 1 pp. 49-80, 2011

Authors' Profiles

Hepsiba Mabel. V received her doctorate in computer science from Madurai Kamaraj University, Madurai, India. Her areas of research include, knowledge management systems, knowledge representation models and knowledge engineering.

She is a faculty of computer science and engineering and has been teaching software engineering related subjects and knowledge management courses. She is an Oracle certified Java programmer and has contributed in research works like EduNiversity.

Some of her research works are published in international conferences and journals. She is a member of ISTE and IAENG associations. Currently she is working as Senior Assistant Professor at VIT University.

Dr. S. Justus received his doctorate degree from Madurai Kamaraj University, Madurai, India. His research specializations include Object-relational data modeling, knowledge engineering and Big Data.

He has been into academic research and has published several of his research work results in International Journals and Conferences – including SwSTE in Israel and DASMA in Germany. He has also practitioner's experience while working with Software development companies.

He is a member of IEEE, ISTE, IAENG professional associations. Presently he is working as Associate Professor at VIT University, Chennai, and the chairperson for the Software Engineering Research Group (SERG) in VIT University.

How to cite this paper: Hepsiba Mabel V, Justus Selwyn, "A Review on the Knowledge Representation Models and its Implications", International Journal of Information Technology and Computer Science (IJITCS), Vol.8, No.10, pp.72-81, 2016. DOI: 10.5815/ijitcs.2016.10.09