

## MOBILE ONTOLOGY DESIGN FOR SEMANTIC WEB: A CASE STUDY

*W. Aisha Banu, P. Sheik Abdul Khader and R. Shriram*

School of Information and Computing Sciences,

B.S. Abdur Rahman University, Chennai, India.

Email : (aisha, paskhader and shriram)@bsauniv.ac.in

### ABSTRACT

*The era of ubiquitous computing and communication has started to change the dynamics of design of systems in a big way. Mobile computing has become the norm and solutions that can automatically understand user intent are the need of the hour. For this, ontology design for mobile computing is crucial. Designing ontology systems that are tailored towards heterogeneous mobile environments imposes demands such as data management, reasoning systems and memory efficiency. The systems must be expressive to include the dimensions of the context, scalable and focused towards the users. In this paper we discuss the design of the ontology for mobile environment and a semantic encoding system that can manage the data. The experimental results show the promise of the work and applicability to various operations in the domain.*

**Keywords:** *Mobile computing, Mobile ontologies, Ontologies, Semantic Web.*

### 1.0 INTRODUCTION

The domain of mobile computing has evolved by leaps and bounds in the past few years with the growing capabilities of the phone based devices and the trend towards reduced cost of communication. The enormous popularity of mobile devices can have a profound effect on the lifestyle of the users, unleashing new levels of connectivity and personal mobility. Web access from mobile phones has been growing rapidly as a result of this and new areas of research have opened up. Mobile phone web usage is increasingly identified as distinct from conventional web usage from computers due to the problems like disconnected operation, small screen sizes, low processing speed, need for content adaptation and limited support for application development. These challenges pose considerable issues to the research and development effort. The keys are to find methods that overcome the limitations of the domain and still incorporate the advances in web intelligence research.

The domain also exhibits characteristics like personalized operation, importance of context, low tolerance to information overload, task directed operation, desire for trust, importance to user interaction design and a heterogeneous environment of operation. The mobile phone user has a lower tolerance to information overload due to the limitation in the usability aspects. Hence there is a need for fast and relevant information access algorithms with a tolerance for accuracy. In other words, the best fit approaches are less favored and the trend is for fast re-ranking. While a lot of the above problems can be overcome with the advent of technology with its emphasis on devices with advanced capabilities, the fact remains that as in the computing environment, the need is for usable and semantic systems. These systems demand solutions that will predict the needs of the users and retrieve information that is closest to the user intent. This work explains the design of an ontology for web applications for mobile devices and explains a novel semantic encoding technique to ensure the scalability of operation. The work was validated both empirically and by users and the results are presented here. The organization of the rest of the paper is as follows.

The state of the art in the mobile ontology systems and their applications is examined in the next section along with a discussion on the trends in the domain. The design of the ontology for mobile device web applications is given in Section 3. Section 4 explains the semantic encoding algorithm. Section 5 discusses the results and the validation of the system and Section 6 concludes the paper and explains the future research directions.

### 2.0 LITERATURE REVIEW

One of the key issues in understanding the user intent is to personalize the context of users and move away from the one size fits all approach of the conventional web. The need is for the system to understand the query of the users and process the query in terms of the global information. For this, the ontology [1] can be used. The ontology is a shared explicit specification of a concept in a universal form. The ontology enables us to understand not just the meaning of a term, but also explore the linkages of the term in the global and local context. Thus, for understanding

the user intent, the need of the hour is to understand the intent by using the ontology and using this understanding to deliver information to the users. Ontologies for mobile phones and the domain is an emerging area of research and is perhaps the key towards delivering relevant information to the users. A comprehensive survey on the state of the art in Ubiquitous computing and Semantic web encompassing the different dimensions of mobile ontologies was given in [3]. In the paper, the challenges for the future have been outlined as a) support for search and indexing within the ontologies, b) extraction and segmentation of ontology parts and c) support for plug-ins from other sources. Of this, the first two aspects are covered in this work. As the design is data centric, the support for data plug-ins from external sources can be easily built in later. There have been initiatives aimed at mobile ontology research. [4] is an example of the initiative by the European Union in this regard. The work in [1] outlines context directed database oriented ontology. Our work is designed with this in mind. The ontology structure for a mobile phone has been described in [5]. This structure has been explained in terms of the OWL and the RDF structure so that it is machine readable. [6] proposes a low-complexity version of Prediction by Partial Matching (PPM). A hash table technique is employed in this method to design the data structure for data and context model. The major focus of [7] is the Two-Level Word Replacing Transformation (TWRT). This scheme can use up to two dictionaries, which are dynamically selected before the actual preprocessing starts. The mobile ontology for semantic web is crucial is it can bridge the structured and unstructured knowledge in a single platform and can be used with advanced reasoning systems for operation. The mobile ontologies also act as a bridge [2] between the service platforms and mobile communications. A mobile ontology of a user can incorporate the personal preferences (people, place, context, preferences, etc.) in an expressive manner. For example a hierarchical organization of circle of people can be organized into friends, family, colleagues, collaborators, acquaintances, service providers, service requesters, consumers, etc. each of whom will have a meaning to the individual users. The ontology can then help the software interfaces towards user centric operations which can gauge the intent and predict the operations based on the intent. Thus the design of the ontology is crucial in making the mobile phone as a computing and communication media. Ontology design for mobile phones is a distinct problem in semantic computing in that the limitations of the domain must be taken into account. An expressive structure with reliance on huge knowledge base may not be implementable in a mobile phone based environment. Also, the nature of the services that use the ontology must be taken into account and emphasis must be to maintain the completeness while also tailoring the system towards the services that use the system.

### 3.0 MOBILE ONTOLOGY DESIGN

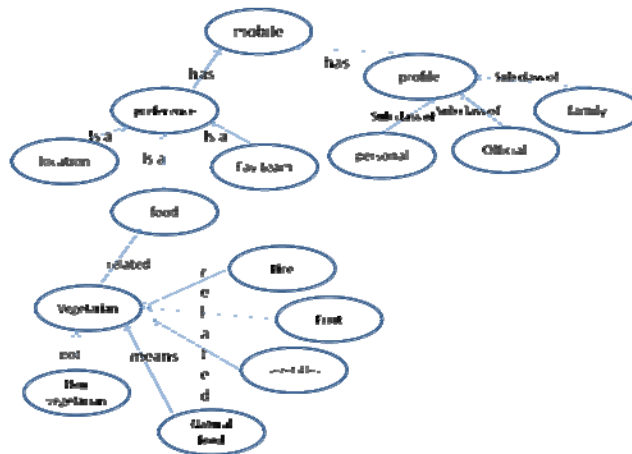
The design of the mobile ontology is governed by the context of operation of the users. This work proposes the event-condition-action formalism as a basis for the design of the ontology. The basic aspect of the design is that it must allow the expressiveness of the language at the same time be implementable in the mobile computing system. We propose a meta language-database centric formalism for the ontology representation. The core aspect of the work is in the use of tag indicators for expressiveness.

#### 3.1. Basic Structure

The core structure is represented in the form of a database which can be retrieved by access languages. Expression operators are applied on the strings that are stored to give meaning to the content. A portion of the Ontological structure for mobility is given in this section. The basic unit of the ontology is a term. An important aspect is the distinction between the terms and words. The mobile ontology considers the basic unit as a term not a single word. Hence 'semantic' and 'web' are different from 'semantic web'. & operator represents the class hierarchy. Applying & before the string will denote that the string is a super class of the next stored elements. For example \* operator represents the related terms. Applying \* before the string will denote that the string's related terms are the next stored elements. ^ operator unites the keywords. + operator represents the relevant terms. - operator represents the opposites. ## and # operators represent the subclass hierarchy in generalized cases. Also in case the terms need to be connected to each other with distinct connector labels, then the ## terms can be applied without the # terms. Due to space constraints a full description of the ontology is shortened. The sample ontology is shown in Table 1 and Figure 1.

**Table 1.** Sample Ontology Design.

## Mobile	#has	Profile	Preference	
& Profile	Official &	Friends &	Family &	
## Preference	# Isa	Location	Food	Fav team
^ Food	Vegetarian			
Vegetarian	-Non vegetarian	Natural food		
*Vegetarian	Rice	Fruits	Vegetables	



**Fig. 1.** Sample Ontology for Table 1.

In case the term is not available in the ontology, then, the user is prompted to enter the values for the words. Once an entry is created, the entry is double indexed with default values filled. Meaning, if an entry is made that ‘apple’ is related to ‘computer’ and ‘fruit’, then automatically the entries for the term ‘computer’ and ‘fruit’ are updated. Thus the bi-conditional relationships are valid. This relationship is also applicable to the fuzzy terms ‘relevant’. Thus, the terms can be linked too. Computing, communication and computers are related to each other. Thus the entries for communication can be updated with computing and computers. Similarly the entries for the computers can also be updated. The user is prompted for the conflict cases. In this case, if a connector relationship already exists in the form of ‘can perform’ for ‘computing’ and ‘computers’, and now the user unknowingly tries to create the relationship ‘computers’ and ‘computing’ related to each other, then the user is prompted for the conflict of rules case. At the initialization of the operation, the ontology set populated with data will be used. Over time, the user’s data will dominate and the data will become personalized.

**3.2. Reasoning System**

The reasoning mechanism (Figure 2) is built to handle the rules as an event condition action rule ontology reasoning system. The overall operation of the system has the following inputs: a) the mobile ontology base, b) the event classes, c) the rules that act as input to the event classes and d) the actions that can be triggered as a result of the application of the events and the rules. The primary aspect of the system is the event. The events are standardized actions defined in the system. Finding the relationship distance, forming the related terms, and finding the super class of a term are examples of the events. Events can be global – with reference to the entire ontology or local with reference to the query term and its connected words alone. For the event, the rules are applied by the mobile ontology manager and the actions are undertaken. The pre-condition of the events (number of the terms needed for

processing, the nature of terms used, etc.) can be retrieved from the rule base and used for the subsequent processing of the commands. The reasoning system computes the inference as a simple tree based structure with the default values (operators like +, -, & etc.) supplying the slot and filler structure. For each event, the rules that can be applied can be retrieved by the reasoning system and based on the needs of the system, the appropriate actions are selected. The action clauses are called by the event and can be inherited from the super class.

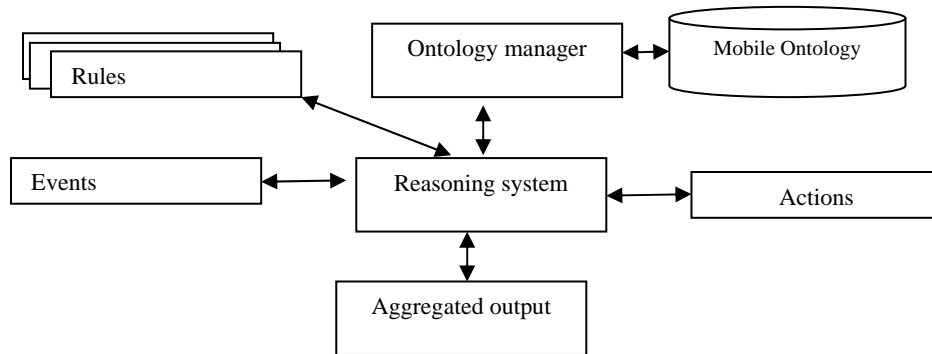


Fig. 2. Ontology Reasoning System.

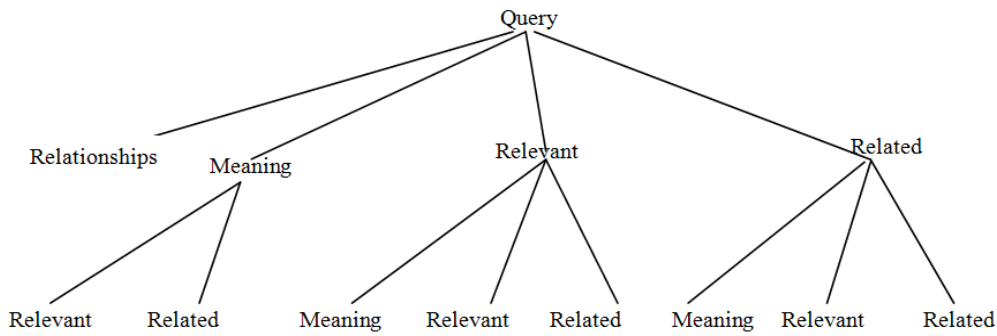
The reasoning system can couple various rules in a sequence or have them run in parallel. For example for finding the relevance of a snippet to a query, the reasoning system needs to find out the term relationship of each word in the snippet to the query using the ontology, weigh the overall value for the snippet and construct the composite rating. The term relationship rule by itself needs application of various rules finding the local and global relationships and if no relationship exists, then the distance vector measurement in a shortest path is used. Thus, a single event can necessitate the call of subsequent nested events. All this is controlled by the event manager. Any ontology update will first be validated by the ontology manager which will check for overlap, inconsistency in the rules, hierarchical conflicts and cross linking of structures. The system allows multiple paths between two terms as this will enhance the expressiveness of the overall ontology, but at the same time check for inconsistencies. The outcome of the event manager is sent to the aggregated output system which will then form the result. The separation of the reasoning system and the output aggregation is to ensure that the overall system is modular in nature and the data and the code aspect are separated clearly.

### 3.3. Operation Of Reasoning System

To show the sample of the operation of the ontology operation, the query expansion stage is explained in detail here. The query suggestions are located in the precedence order of

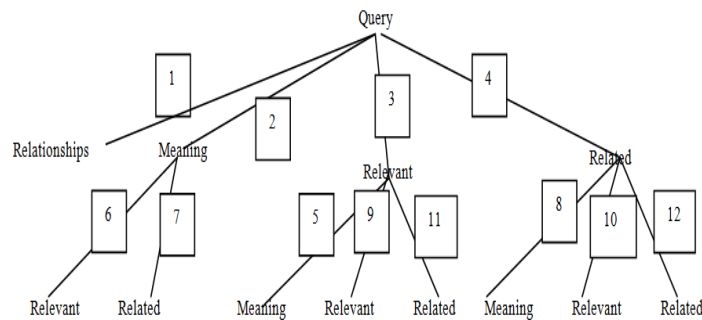
- 1 Meaning
- 2 Relevant
- 3 Related.

For a given query, the corresponding terms are searched one by one in the Ontology in the precedence order, till nine terms are reached. The system is modeled as a dependency tree shown in Figure 3. The dependency tree parsing algorithm is used to model the overall process of retrieval. The query is the overall guideline of the entire process. The query is expanded in terms of: relationships, meaning, relevant and related contents in the Ontology. In general, whenever two words are connected by a dependency relation, one of them is the head and the other is the dependant, and that there is a link connecting them. According to the conventional theory, the dependant is the modifier, object, or complement. The head plays the larger role in determining the behavior of the pair. The dependant presupposes the presence of the head, and conversely, the head may require the presence of the dependant.



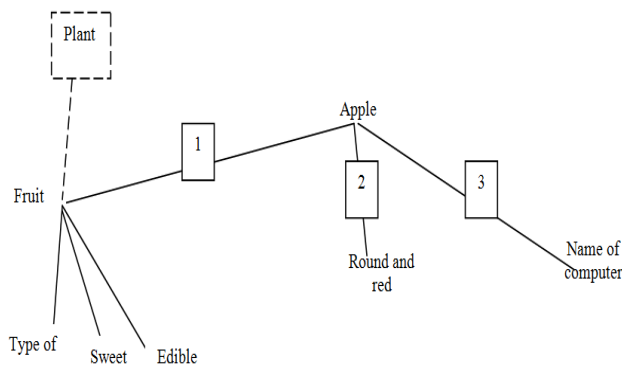
**Fig. 3.** Query Dependency tree – Priority Order

The overall structure of the dependency tree shows the relationships between the constituent terms. The order of parsing is shown in Figure 4. The logic of the precedence order is that the meaning of the word is paramount. Once the terms in the meaning column are exhausted, the contents of the relevant terms are expanded, and then the related terms. The next step is to expand the relevant and related terms of the meaning node. After this, the meaning of the related term is expanded. By this, the precedence order is maintained and the semantic relationships established. Thus, the expansion is based on the contextual dependencies of the content. The example for precedence in the dependency tree is shown in Figure 4.



**Fig. 4.** Order of Parsing in the Dependency tree

The dependency parser follows a single uniform set of steps. There is only one step that follows the current step, and all the next steps are clear and well defined. Each term has only one head, and can have zero or more ‘children’.



**Fig. 5.** Example – Order of Parsing in the Dependency tree

While the overall structure may be a dependency graph, during the query expansion, a clear dependency structure is followed, and the terms populated for the user. As is clear from the above example (Figure 5), a fruit may, in turn, have many nodes entering it. But, during the expansion stage, the contents are a dependency tree alone. The following example shows the query expansion process.

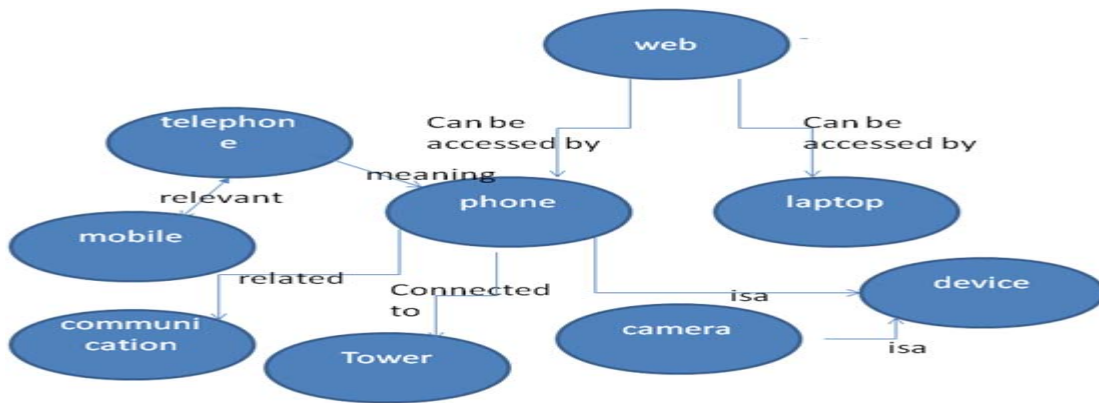


Fig. 6. Ontology – Sample Snapshot 1

In this case the query is the ‘telephone’. (Figure 6). The expansion order based on the ontology shown in Figure 4, is shown in Figure 7.

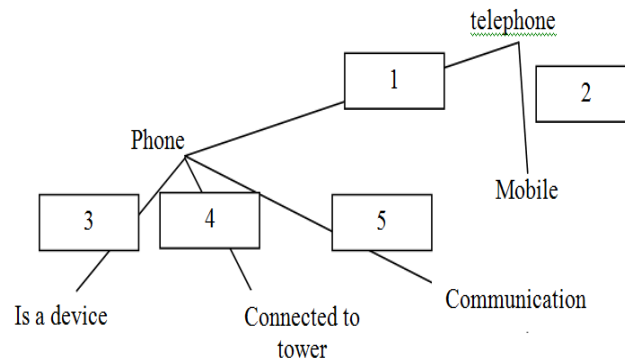


Fig. 7. Expansion order for Fig 4

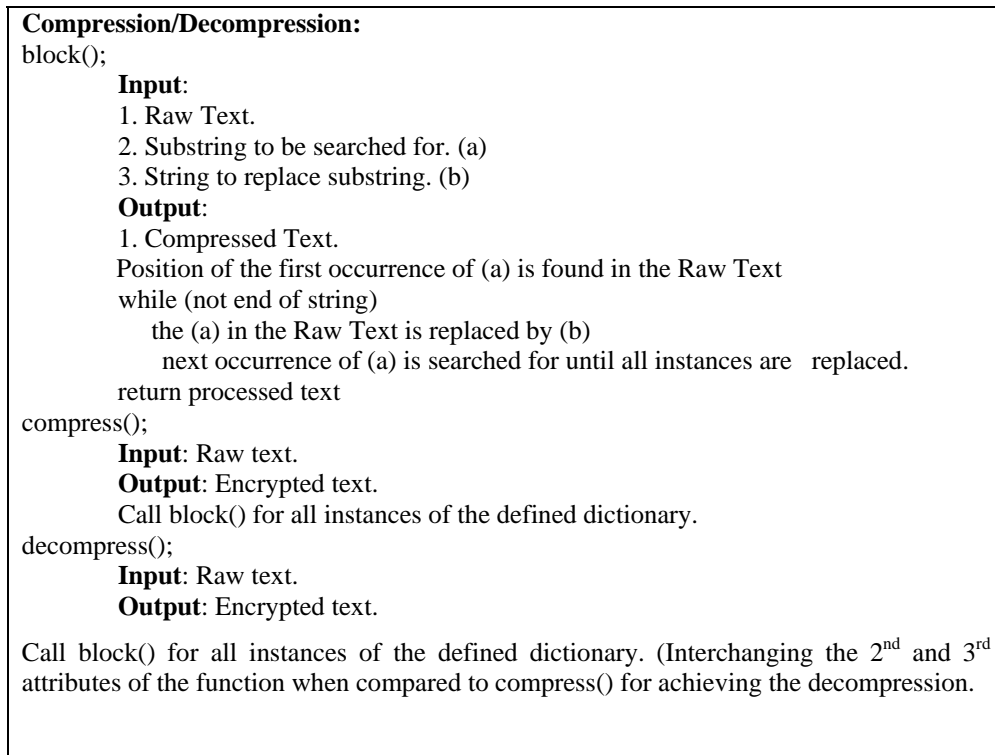
Thus, the meaning term is expanded first, followed by the relevant term. In the case of the connector rules, the order is L-R followed by the corresponding nested precedence order. In the case of a connector relation, (can be accessed by) unless there is a directional relation (can access), the term is not expanded. The above example illustrates the overall process of query expansion. The query expansion proceeds until 9 suggestions are generated. The number 9 represents the display capacity of the conventional mobile device screens. In case there are more than 9 suggestions, the system shows a ‘...’ option at the end with a number 0. The indication is that if the user wants more, then the user presses the 0 option.

### 3.4. Compression Method

The ontological system has the potential to spiral an exponential growth in terms of the memory requirements especially for information retrieval applications. To give an example, a snippet contains 15-20 words. In case a single snippet is processed using the ontology, it will need at least 150 to 200 terms in the system. The rate of the exponential growth will depend on the user’s domain of operation and the facilities for dynamic updates of the user’s ontologies. There is a need for compression models for this aspect. The pseudo code for the compression and decompression is given in Figure 8.

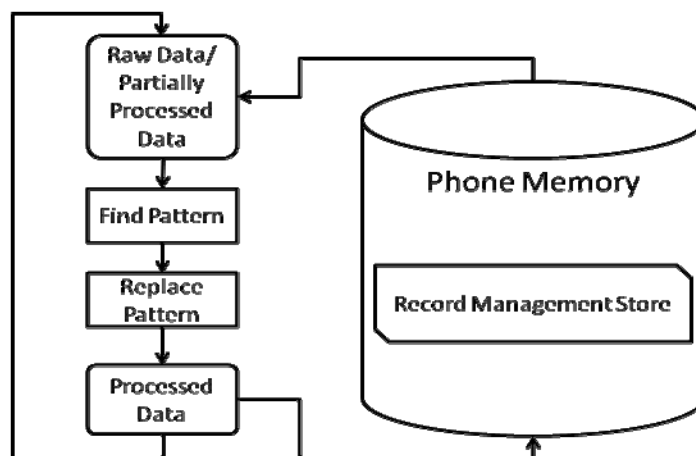
Our proposed compression model relies on a two level model: personalization and parts of speech. The users can input as a part of their profile, a set of words. These set of words will be taken for the personalization aspect pei.. The users can also dynamically add new words to the system - npei. All other terms will be compressed using the parts of speech model. The basis behind the model is as follows. All English terms have sets of parts of speech which are repeated. These parts of speech items can be replaced by a single symbol in the storage system giving a savings of n:1 where n the count of the number of words in the substring. We have done a study of the set of words most frequently used in English and also used in the mobile context too. These words are then replaced by

equivalent symbols. While the ratio of the compression will be more if the parts of speech symbols taken have a higher granularity, it is observed from the studies on the mobile phone usage that most of the terms used in the mobile phones have a low size of the words.



**Fig. 8.** Pseudo-Code for Compression and De-Compression

The most important module in the compression technique is the compression/ decompression module. In this module the raw text is taken as input in the above listed three applications. Two predefined tables exist, one for encoding and the other for decoding the text. The encoding table consists of the frequently used patterns in the English language and an encoding code for each of the patterns. The decoding table consists of the reverse of the encoding table; it consists of the encoding codes and the patterns associated with each of the code. The encoding process takes place by comparing the patterns in the input text with those of the encoding table, and if the pattern matches, the corresponding code for the pattern is substituted instead of the pattern. In the decoding module, a similar form of action takes place where the text is again searched and instead of the codes, the patterns are substituted. Thus, the original text is displayed to the user.



**Fig. 9** Data Compression – Process Flow

The compression/decompression module (Figure 9) takes the input as raw text and compares the patterns present in the text with the predefined dictionary. If there is a match, then the pattern is replaced with the code. This process is carried out for the entire text, and the resulting output is stored in the Record Management Store. This is the compression process; in the case of decompression, the reverse process is followed. The data is taken from the record store, compared with the predefined dictionary, and the codes are replaced by the patterns. The decompressed data is now displayed to the user.

### 3.5. Event Condition Action System

The Ontology has been implemented using the Event Condition Action (ECA) system. In general, the system consists of a protocol for handling the rules and conditions. The rule element in the schema model has a set of attributes. Each element in the schema is made up of a sequence of a combination of attributes and other elements. The schema model allows the ECA rules to be specified as part of a set or a part of a protocol. The events are said to be an interaction between the user and the information retrieval, causing a change in the state of the contents of the system. Here, the contents are a combination of the external context and contents; in short, the 'personal data'. The events-condition-action rule (ECA rule) is the most common method for constructing the personal data. Its structure is as follows.

```

onevent
ifcondition
doactions
    
```

Each rule is composed of these three parts. When an event occurs (on event), in accordance with the pre-set conditions (if condition), the event manager responds by performing actions according to the reactions that are implemented by the system manager (do actions). Events are caused by user operations, the system, regular action, or responses to derivative action. When an event satisfies the conditions in the set rules, it is known as the trigger. When an event occurs, the following steps are taken:

- a. Finding a relevant trigger,
- b. Evaluating the condition and
- c. Executing the action if the condition is true.

The overall structure of implementation for the storage process is shown in Table 2.

**Table 2** Sample Event Condition Action System

Category	Event	Condition	Action
Storage	Add a word – meaning	If existing	Allow the user to edit the word
		If new	Check the Right Hand Side (RHS) of the word
	Check RHS of word	If existing	Add the word to the meaning of the RHS term; Create a new term with the word and the RHS.
		If new	Create a new term with the word and RHS term; Create a new term with the RHS term and the word.
	Add a word-relevant	If existing	Allow the user to edit the word
		If new	Add the relevant term to the RHS of the word with an appropriate operator
	Add a word-related	If existing	Allow the user to edit the word
		If new	Add the relevant term to the RHS of the word with an appropriate operator
	Add a word-subclass	If existing	Allow the user to edit the word
		If new	Add the relevant term to the RHS of the word with an appropriate operator structure



### 3.6. Ontology Management

Overall, the mobile ontology system has been described in this section. The mobile ontology system can be implemented in all models of mobile phones. In general, mobile phones can support database structures and/or tree based structures. The above ontology can support both types of systems. While the efficiency will be more if a connected and linked structure is supported by the system, the database structure is not an impediment for the operation.

Overall, the ontology based system for the mobile environment has been discussed in detail in this section. The ontology is expressive enough to handle the range of data it can encounter, flexible to be applied in heterogeneous mobile environments, has reasoning system to make sense out of the data and is personal to the needs of the user. These features along with the distraction free computing aspects (modes) of the mobile ontology have been tested and the results are explained in the next section.

## 4.0 RESULTS AND DISCUSSION

The mobile ontology was implemented using J2ME for mobile phones and tested experimentally for the search query retrieval re-ranking. The overall system used, accepted an input from the users, searched the ontology for the related terms and refined the query, then, the query was searched in the web. The snippet results of the web search were refined further using the ontology and the re-ranked results shown to the users. The key aspect of this work was the user of ontologies in the web search and the incorporation of the compression system in the overall process. The experimental work consisted of testing the system using the mobile phones on the Search applications developed by our team. In this, twenty users used the prototype model for web search. Their queries, interaction were noted and analysed both after collecting the log files and understanding the data modalities. A simulator was developed in Java to analyse the interaction models and find the key metrics. The data was pruned to take the aspects which were coherent. The interaction of the user produced terms which were then populated into the ontology. These terms and the outcome of the search process were modelled in the system based setting in parallel. Thus a parallel simulator based model (one where the users used the mobile phones for web search) and a system based model where the growth of the ontology was modelled was run in parallel. For this reason, metrics which could test the speed of the system, user satisfaction and other user subjective measures were not used in the testing process. The users were pre-final students of Engineering with similar academic and social backgrounds. They were well versed in using the system for their regular work and on an average used the system for 1.5 hours a week using the mobile phones. The parameters were validated empirically. In a system like this, there are several parameters that can be tested.

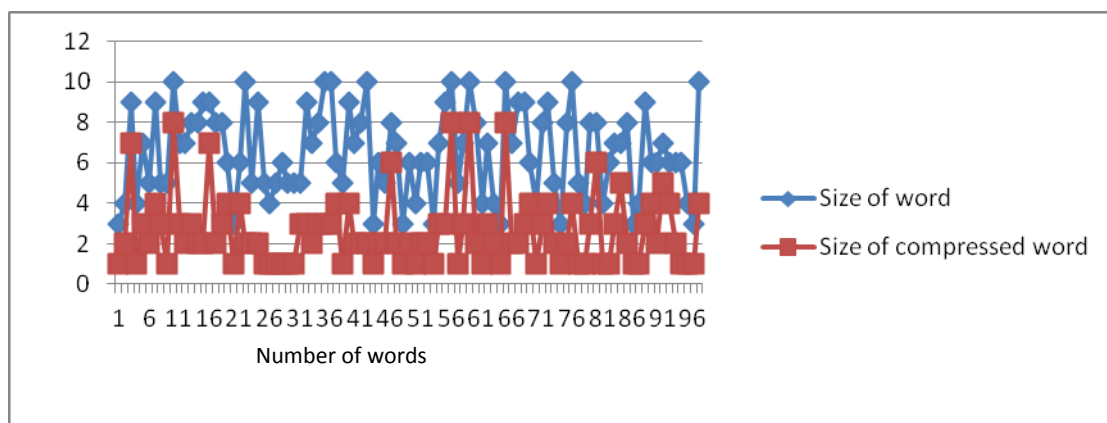
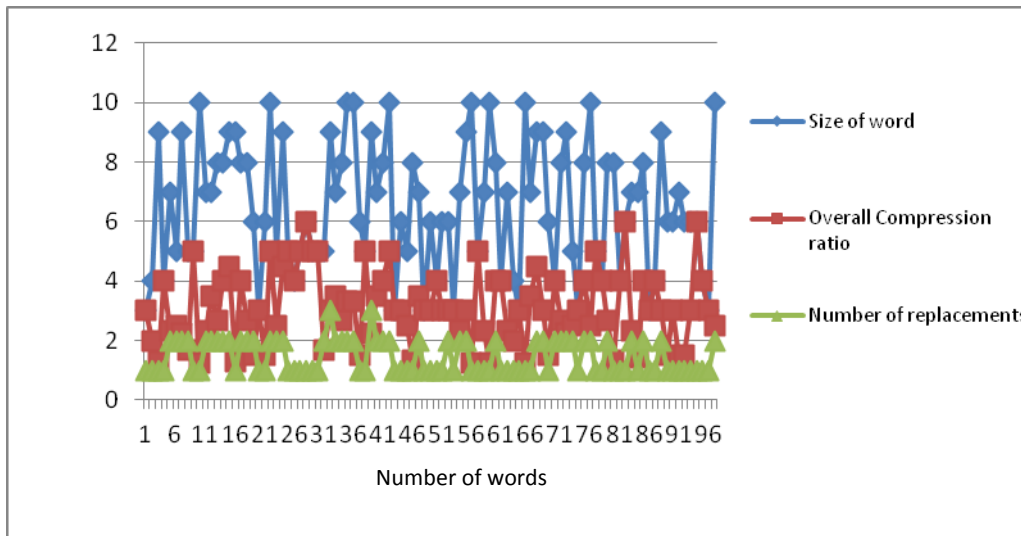


Fig. 10. Compression System efficiency.

In this work, we tested the efficiency of compression models -rate of growth of ontologies and replacement number-performance of system aspects. The compression model is an important aspect of the system. As the number of queries increases, the ontology in the system can grow. This again can impact the compression efficiency of the system. The graph in Figure 10 shows the overall size of the words used and the compressed words used by the system. The y-axis stands for the term values: Size of word and size of compressed word. The x-axis stands for the number of words used by different users.

The trade-off aspect of the compression model is the size of the word, the compression ratio and number of replacements. As the number of replacements of the parts of speech increases, the performance of the system can vary. The objective of the compression is to minimize the data storage in the system. This is especially true of an ontology like this. Hence while we have to aim for maximum replacement efficiency, if the number of replacements for a single word is more, then more CPU cycles will be consumed. Hence the system must aim for minimal replacements and maximum compression. This is aimed at in this system. The diagram (Figure 11) shows that the average number of replacements of the system is 1.25. The y-axis stands for the term values: Size of word, size of compressed word and the number of replacements. The x-axis stands for the number of words used by different users. The objective here is to see the effect of the compression when different words are used at random.



**Fig. 11.** Compression and Number of Replacements.

This shows that though the system delivers high performance the overheads of the performance is very low. Table 3 shows the snapshot of the compression and number of replacements for a sample session.

**Table 3** Compression efficiency

User	Compression %		
	Ave. Size of word	Ave. Size of compressed word	Number of Replacements
1	7	5	1
2	4	2	1
3	6	3	1
4	5	5	0
5	6	3	1
6	8	3	1
7	4	2	1
8	6	3	1
9	3	1	1
10	10	4	2
11	7	3	2
12	8	4	2
13	3	1	1
14	10	3	2
15	3	1	1
16	5	2	1
17	6	3	1
18	9	4	2
19	6	3	1
20	10	4	2
Average	6.25	2.95	1.25

There are a large number of metrics that can be tested. We have selected a few metrics that can influence the performance of the system which are the storage and the operation of the system. In future the effect of the replication on the ontology will be tested and benchmarked.

## 5.0 CONCLUSION

This work has explained the design of the ontology for mobile applications. The structure, operation, reasoning systems and the compression algorithm for ontology systems has been explained in detail. The ontology has been designed keeping in mind the limitations of the domain and the needs of research applications like information retrieval and web usage. A prototype model using the ontology has been developed and some aspects of the ontology tested. The work shows that the ontology shows considerable promise for the usage in the domain. In future the focus will be on semantic methods to address the exponential data increase in the system and to analyze whether the contents of the ontology influence the metrics of the system.

## 6.0 REFERENCES

- [1] Specht, G., & Weithoner, T.: "Context-aware processing of ontologies in mobile environments." *In: IEEE International Conference on Mobile Data Management* (Vol. 2006).

- [2] Villalonga, C., Strohbach, M., Snoeck, N., Sutterer, M., Belaunde, M., Kovacs, E., Zhdanova, A. V., Goix, L. W., & Droegehorn, O.: "Mobile ontology: Towards a standardized semantic model for the mobile domain." *In ICSOC Workshops*, pages 248–257. (2007).
- [3] Zhdanova, A. V., Li N. & Moessne, K.: "Semantic Web in Ubiquitous Mobile Communications," *In: The Semantic Web for Knowledge and Data Management*, Ma, Z., Wang H. (Eds.). 2009.
- [4] Spice Mobile Ontology Project. [http://ontology.ist-spice.org/spice\\_ontologies\\_files.htm](http://ontology.ist-spice.org/spice_ontologies_files.htm)
- [5] Zhu, J., Li, B., Wang, F. & Wang, S.: Mobile Ontology. *JDCTA* 4(5): 46-54 (2010)
- [6] Sadler, C. M. & Martonosi, M.: "Data compression algorithms for energy constrained devices in delay tolerant networks." *In Proceedings of the 4th international conference on Embedded networked sensor systems (SenSys '06)*, pages 265–278, New York, NY, USA, 2006.
- [7] Suresh, D. C. Agrawal, B., Yang, J. & Najjar, W. : "Energy- efficient encoding techniques for off-chip data buses". *ACM Transactions on Embedded Computing Systems (TECS)*, Volume 8 Issue 2, January 2009.

## BIOGRAPHY

**Mrs. W. Aisha Banu** works as Associate Professor in the Department of Computer Science and Engineering at B. S. Abdur Rahman University, Chennai, India. She has published over 5 papers in the domain of Information retrieval, web mining and mobile computing. She is pursuing her Doctoral work in Anna University, Tiruchirappalli, Tamil Nadu, India.

**Dr. P. Sheik Abdul Khader** is the Head of the Computer Applications Department at B. S. Abdur Rahman University, Chennai, India. He has published over 30 peer reviewed journal and conference papers in the domain of soft computing, Networks and Data mining.

**Dr. R. Shriram** is a Professor in the Department of Computer Science and Engineering at B. S. Abdur Rahman University, Chennai, India. He has published over 12 peer reviewed journal and conference papers in the domain of Pervasive computing, mobile computing and information retrieval. He has completed many funded projects in the domains of Pervasive Computing and Mobile Computing.