

A Protocol Design for Ubiquitous Object Categorization and Communication through Query Language

Premalata G. Langati
M.Tech. Dept of CSE
KLS Gogte Institute of Technology
Udyambag, Belagavi, Karnataka, India

Vidya R. Kulkarni
Associate Professor, Dept of CSE
KLS Gogte Institute of Technology
Udyambag, Belagavi, Karnataka, India

ABSTRACT

An autonomous and intelligent computing object that can be accessed by anything, anytime, anywhere is known as ubiquitous object. An environment in which these objects are present is known as Ubiquitous Computing environment. Advancement in computing technologies resulting in the growth of computer technologies and computer based applications; there is a demand to access information at anytime, anywhere remotely. This has resulted into the research focus towards ubiquitous computing. In ubiquitous environment a large number of objects communicate with each other. The question is how to provide a unique identification to objects in growth. This requires the design of unique protocols to establish meaningful communication between the ubiquitous objects. And it also provides the uniqueness of object identity and object categorization. In this paper protocols are designed for ubiquitous objects categorization and query language for communication of objects.

Keywords

Protocols, ubiobject, query language, bnf.

1. INTRODUCTION

To make individual objects more intelligent by providing awareness to the objects as this awareness gives objects the ability to understand their position in a given environment and allow them to make use of the given environment knowledge base to operate more effectively. Objects to be more aware of their own existence, what is appropriate in the given environments and how other objects communicate with them. Objects produce huge amount of data about their identity in the environment in which they are present. It is necessary that object should be more intelligent to make interaction among other objects present in that environment. Providing full information of the object is: what is the object around which the object is present, all other objects that are present in environment. Advancements in ubiquitous computing have initiated the need for a systematic method of classifying objects [1, 2].

Important aspects of awareness are

- To possible the interactions between objects.
- To make more intelligent objects.
- To make the ubiquitous computing environment.
- To make the objects have complete information about themselves and also the information about the environment in which they are present.

Language is the medium of communication between two objects. Here we make use of query language to interact various objects among themselves and with other objects

through queries [3]. For this purpose, query language used, which acts as a mode communication between these objects. Any kind of interactions are possible among objects are through queries. Query language handles data in the form of tree structure. Each node in the tree structure is an object.

Nodes having attributes like:

- Manufacturer ID
- Object ID
- Serial Number
- Expiry Date
- Calorie Content of the object

The Query Language provides different queries that can be used to manipulate or retrieve data from the tree data base.

2. LITERATURE SURVEY

The authors propose design and implementation of self-aware protocol for ubiquitous objects communication and querying using Query Language [1], and the future enhancement to this is an improvement can be made with respect to the “Tag Structure”. Many other fields can be included like temperature, number of calories etc. Another extension to the system is the design of other protocols. The tree structure can be extended for other protocols also. Technologies like RFID [4, 5] and Electronic Product Code [EPC] are considered as the most emerging ubiquitous computing technologies that are used for automatic identification of the objects. Generic term for technologies that uses radio waves for automatic identification of the objects or people is RFID. Several methods are there for automatic identification, the most common one is the RFID tag which is associate with an object or person.

In an ubiquitous environment, all the objects present in the universe have to be given unique identity. The existing EPC has some drawbacks [2].

These drawbacks are as follows-

- The number of bits allocated for each field in the tag is limited. As the number of objects grows, we need more and more bits to assign unique identifiers to each of these objects.
- There is no scope to add intelligence to the objects.

3. PROPOSED WORK

3.1 Problem Definition

After knowing a brief history of Ubiquitous computing various challenges faced by this field, we are ready to formally define the Problem that we are addressing in our project. The Problem Definition involves three different parts; the proposed Protocols (modified version [1]) are realized by addressing the following issues:

1. Providing Awareness to the object.
2. Design of Protocols for.
 - Ubiquitous computing Object Categorization.
 - Scalable ID for Ubiquitous computing objects.
 - Setting the standard for Ubiquitous compliant Objects' communication.
3. Query Language for object communication.

3.2 Proposed Concept for Object Categorization and Identity

An autonomous and intelligent computing object that can be accessed by anything, anytime, anywhere which is known as ubiquitous object. Objects are identified by its unique Object Id provided to it. Here unique object Id is RFID Tag. Each ubiquitous objects are responsible for managing their own internal state, their behavior and also manages interaction with other objects in the environment in which they are present. Once the object identifies the other ubiquitous objects near by, as it stores its identity and information in the knowledge base for further use. The necessary actions are initiated by the execution unit [2,3] depending upon the RFID Tag. Advances in the ubiquitous computing have initiated need for the systematic object classification scheme [2]. Every object can become smart by having their RFID tags, which is attached to the object. Objects have electronic identity along with their physical structure. Proposed ID format for RFID tag content is as shown in Figure 1. Table 1 gives the details of the fields.

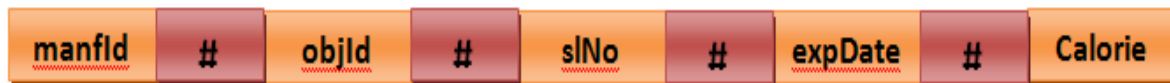


Fig 1: Protocol for Ubiquitous Object Identity & Categorization

Table 1. ID Field Details

RFID Field	Description
manfld	Variable in Length-Unique Identity for Manufacturer.
objId	Variable in Length-Unique Identity for Object Type.
slNo	Serial number of the specific object.
othrP	Objects other programmable details.exp : "Expiry Date" used in protocol versions 1.0 and 1.0.1 And Expiry date as well as "Calorie" is used in protocol versions 1.1 and 1.1.1

Object classification scheme was of five level hierarchies for object classification [1]. Proposed object classification scheme is of six level hierarchies for object classification as shown in the Figure 2 and details are as follows:

- **Logical Aggregation:** Aggregates the generic classes for analysis.
- **Generic Category:** collection of categories.
- **Category:** A Group of generic classes.
- **Generic Class:** A group of interrelated classes commonly recognized.
- **Class:** Group of objects that share common usage.
- **Object:** Common functionalities in a group.

In this project four different versions of protocols are proposed

1)**Protocol Version 1.0 :** In this version each object has RFID Tag attached to them which is of four fields (manfld, ObjId, slNo and expDate) and also Objects are categorized in five level hierarchical structure. At fifth level we can get leaf node, tag is attached to the leaf node object. Modifications are only possible at 5th level.

2)**Protocol Version 1.1 :** In this version each object has RFID Tag attached to them which is of five fields (manfld, ObjId, slNo, expDate and calorie) and also Objects are categorized in five level hierarchical structure. At fifth level we can get leaf node, tag is attached to the leaf node object. Modifications are only possible at 5th level.

3)**Protocol Version 1.0.1 :** In this version each object has RFID Tag attached to them which is of four fields (manfld,ObjId,slNo and expDate) and also Objects are categorized in six level hierarchical structure. At sixth level we can get leaf node, tag is attached to the leaf node object. Modifications are only possible at 6th level.

4)**Protocol Version 1.1.1 :** In this version each object has RFID Tag attached to them which is of five fields (manfld, ObjId, slNo, expDate and calorie) and also Objects are categorized in six level hierarchical structure. At sixth level we can get leaf node, tag is attached to the leaf node object. Modifications are only possible at 6th level.

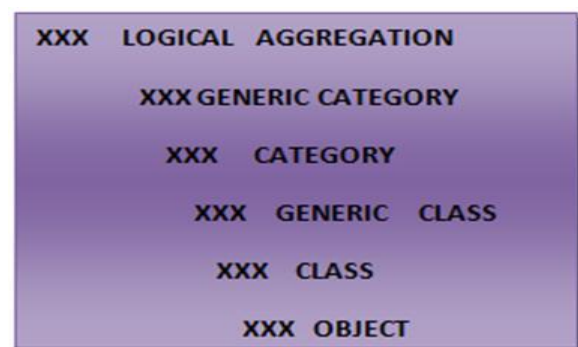


Fig 2: Hierarchical Object Classification

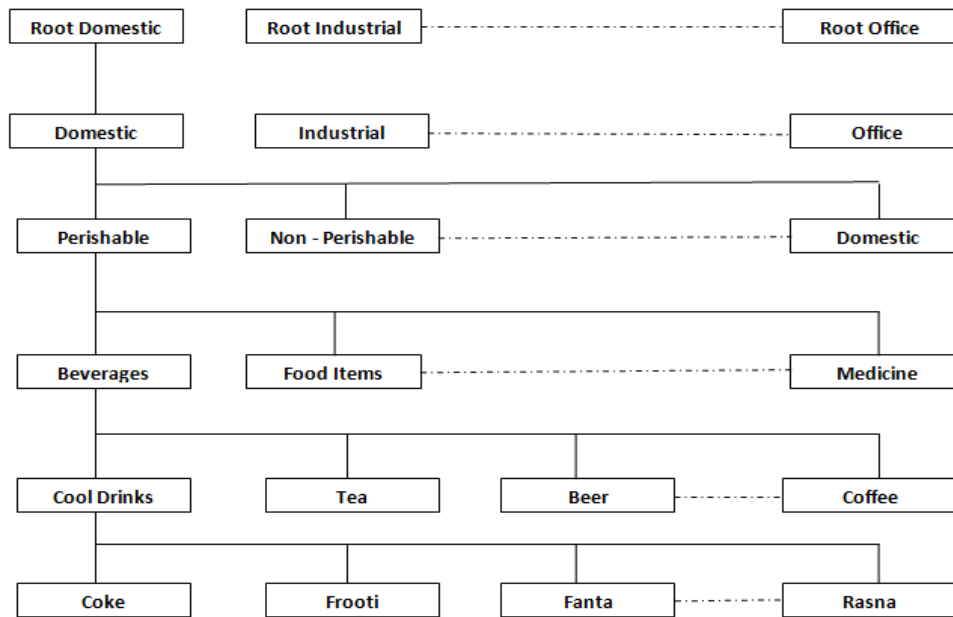


Fig 3: Illustration of Objects Categorization

Consider an example, illustrates the categorization of the objects. As shown in the Figure 3. This is an example for the classification scheme used in the protocol versions 1.0.1 and 1.1.1. The 'expDate' field in the RFID tag determines the expiry dates of ubiquitous computing objects. Here for illustration purpose we have considered three Ubiquitous Computing objects, which are pen-drive, coke and aspirin with RFIDs are as:

- Pen-drive#1.1.1.0.0.1#abc111#12-12-2018#
- Coke#0.0.0.0.0.0#efg111#14-07-2015#22
- Aspirin#0.0.0.3.0.1#mno123#18-10-2017

All these items are placed in the refrigerator, which is known as a smart intelligent component with the processing power, an 'active' element. The items such as pen-drive, coke and aspirin (medicine) are termed as 'passive' elements with no processing power. Refrigerator identifies the type of the objects after reading the RFID tags attached to them and it

may able to send messages to the Owner Object that pen-drive: it is an invalid object to be kept in the Refrigerator. And it may also able to send the messages that contains other details of the objects like items expiry date, its calorie content etc. Hence the user become more aware that which items are valid to be place in the Refrigerator and which are not to be placed.

Language is the medium of communication between two objects. Here we make use of query language to interact various objects among themselves and with other objects through queries. Query language provides various queries. Using different queries manipulation of data or retrieval of information can be done on the tree structured database. Queries are parsed using lexer (lex Program) and parser (yacc Program) [1, 3, 6]. The grammar for yacc program is implementation of the BNF grammar. Figure 4 shows the Query Language as an interface between Refrigerator Object and Owner Object.

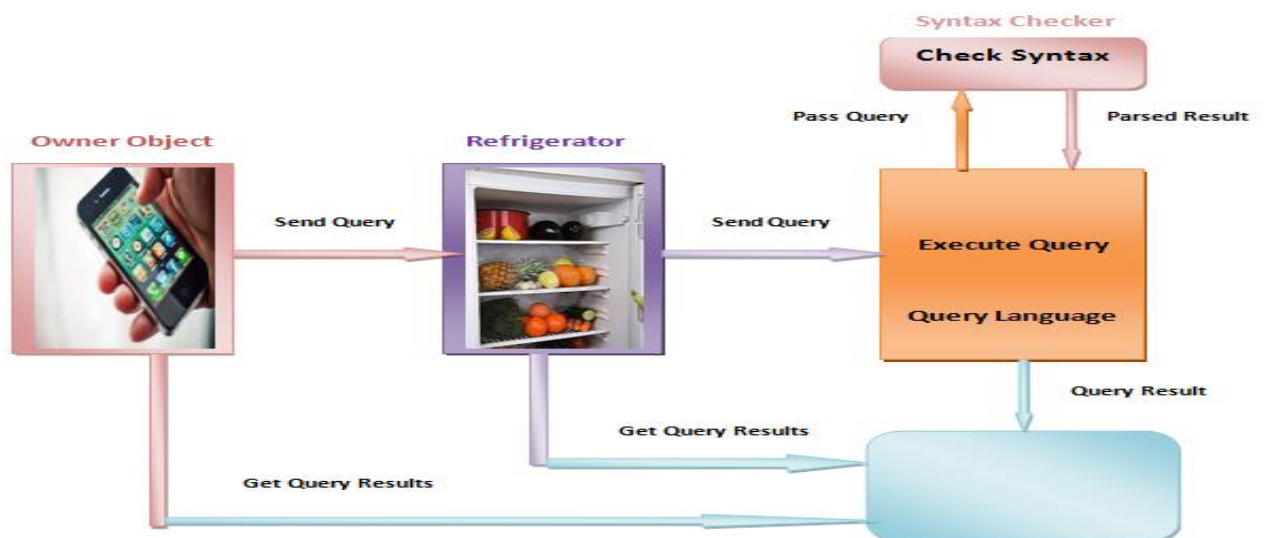


Fig 4: Query Language as an interface between Refrigerator Object and Owner Object

As we have four different versions of protocol, the queries used in the protocols may varied i.e., protocol v1.1.1 is rich in query command compared to protocol v1.0.1 and protocol v1.0.1 is rich in query commands compared to protocol v1.1 and among the four protocols protocol v1.0 has less query commands compared to all.

Protocol v1.1.1>Protocol v1.0.1>Protocol v1.1>Protocol v1.0

4. MODELLING AND DESIGN OF PROPOSED ARCHITECTURE

Several components of the system and its coexistence with their behavior are shown in Figure 5.

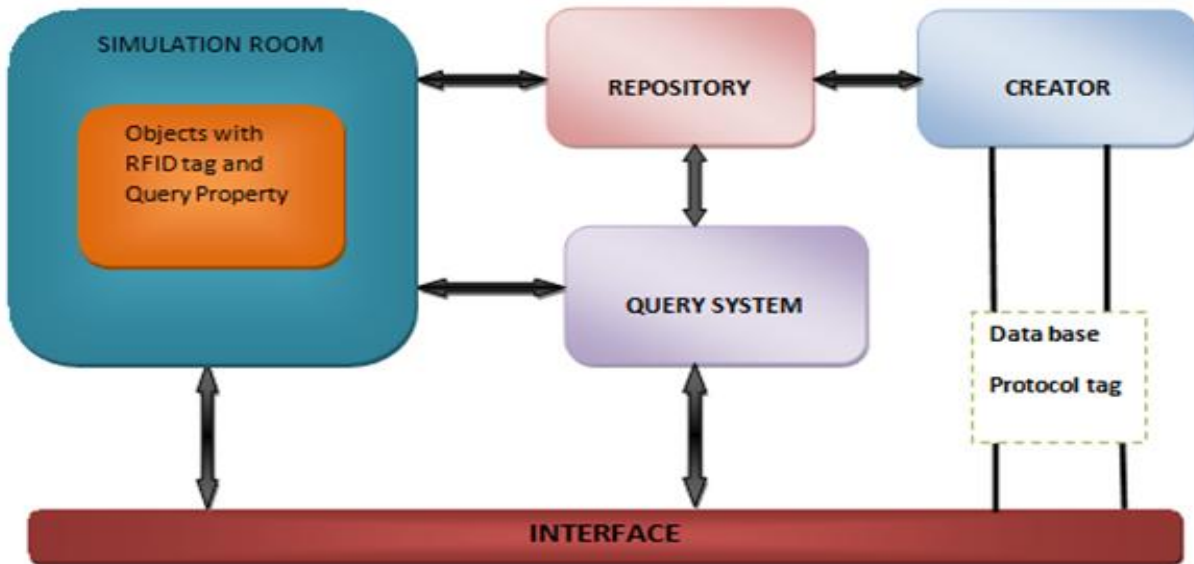


Fig 5: Architectural Design

Creator: creates various objects and also assigns property.

Data Base Protocol Tag: Standardized hierarchical storage and operation in storage, tag repository and usage in specific protocol versions.

Simulation Room: Meeting place for all objects and perform the functions of protocol under simulated environment.

Interface: Main program which encapsulate all system capabilities.

Query system: Subsystem to simulate query and retrieval functions.

Figure 6, Figure 7 and Figure 8 show the use cases of User, Refrigerator and Owner Objects. Figure 9 illustrates the different classes and their association used for implementing Protocols.

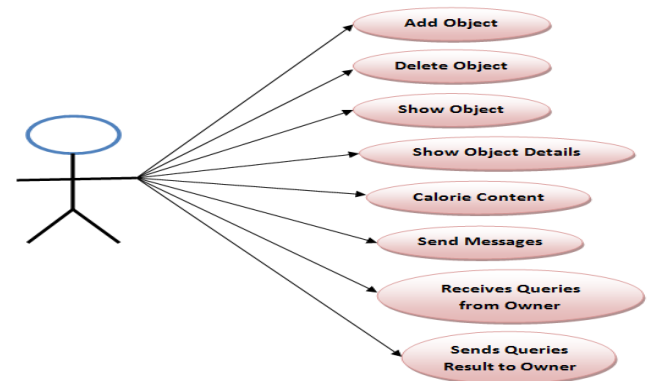


Fig 7 : Use-Case diagram for the Refrigerator Object

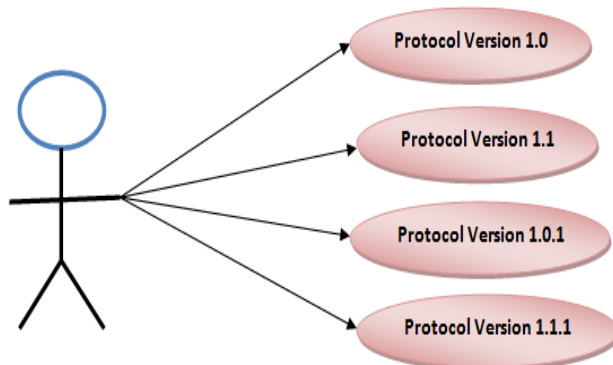


Fig 6 : Use-Case diagram for the User

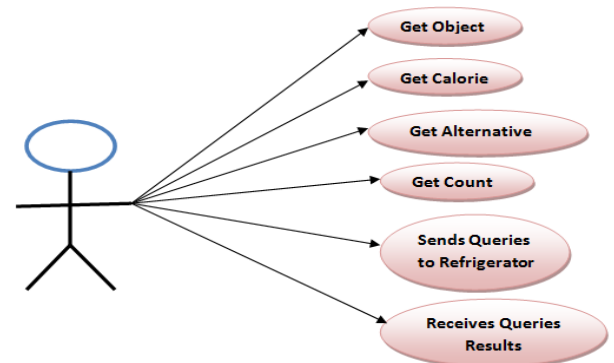


Fig 8 : Use-Case diagram for the Owner Object

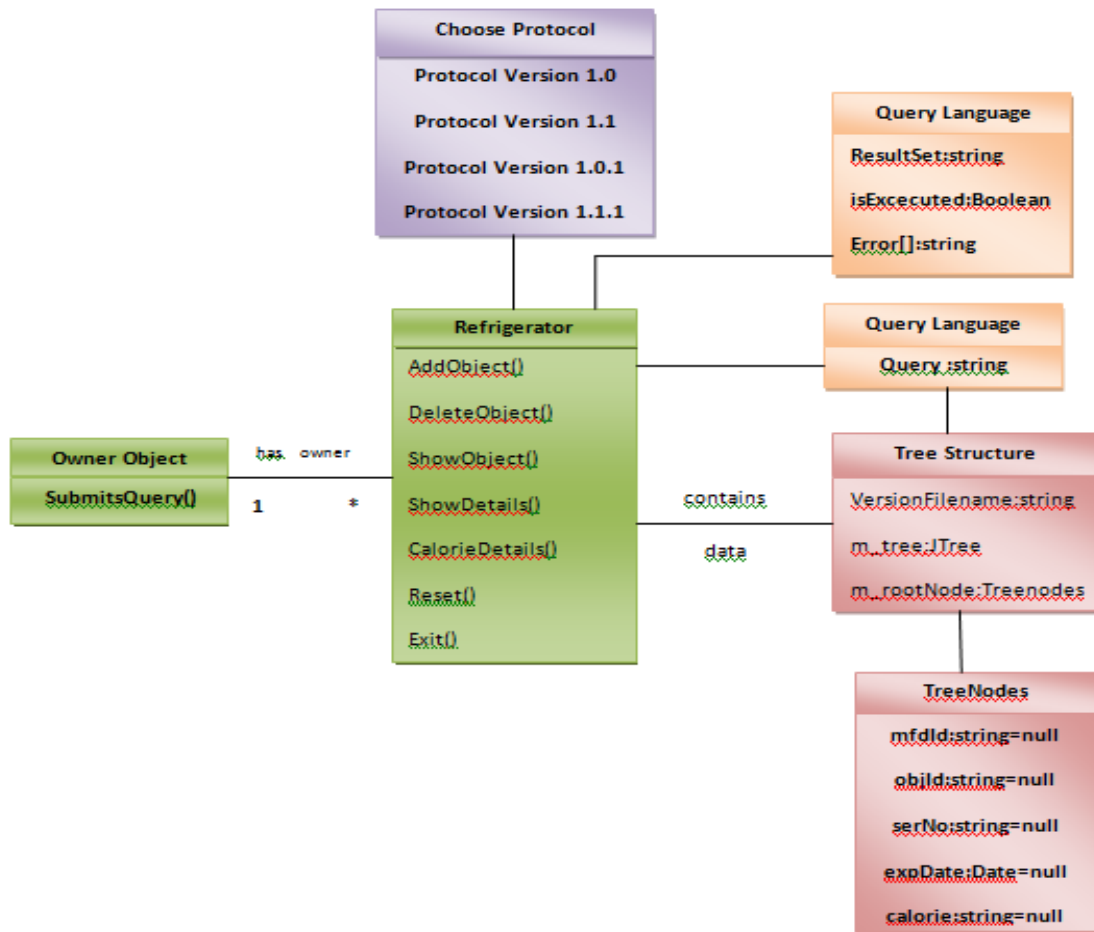


Fig 9 : Class Diagram of Protocols

5. IMPLEMENTATION AND EXPERIMENTAL STUDY

Simulation for the proposed system is implemented using Java Software Development Kit, Eclipse and Linux Platform (Fedora version 13). The proposed Protocols are validated using Refrigerator, Owner Object and Passive Objects.

User: This allows the user to choose a protocol version among four different versions of protocols. Once chose GUI will be disabled. Figure 10 shows the GUI Design for the user.

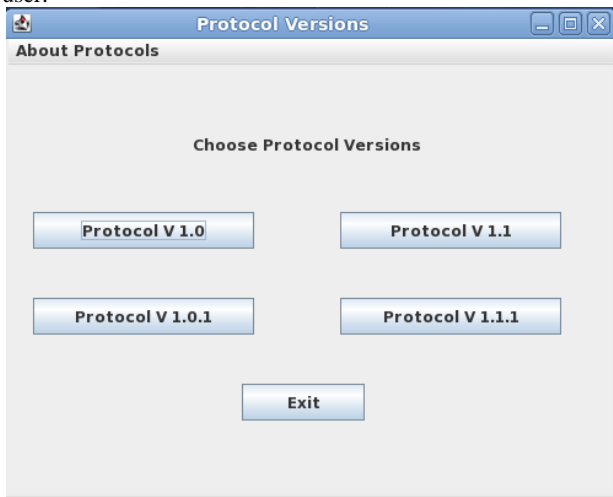


Fig 10 : G.U.I for User

Refrigerator: The Refrigerator Agent acts as Active Object and contains all the information about all the objects present in the Refrigerator in its Knowledge base. An Object emits the RFID signals and these signals are periodically accepted by the Refrigerator object. Accepted signals are processed by the Refrigerator object, depends upon the outcome of the processed data it sends several messages to owner object. Owner object can also sends messages to refrigerator object. The Refrigerator Agent contains dialog box, addition of objects, and deletion of objects, shows the details of objects, also shows information about calorie content of the objects. And it also shows the several activities taking place in the refrigerator. It receives the various queries submitted by the owner object and process the accepted query and sends the result of the submitted query to the owner. Here assumed that user selected a protocol version 1.1.1. And simulated Refrigerator GUI is shown in the figure 11.

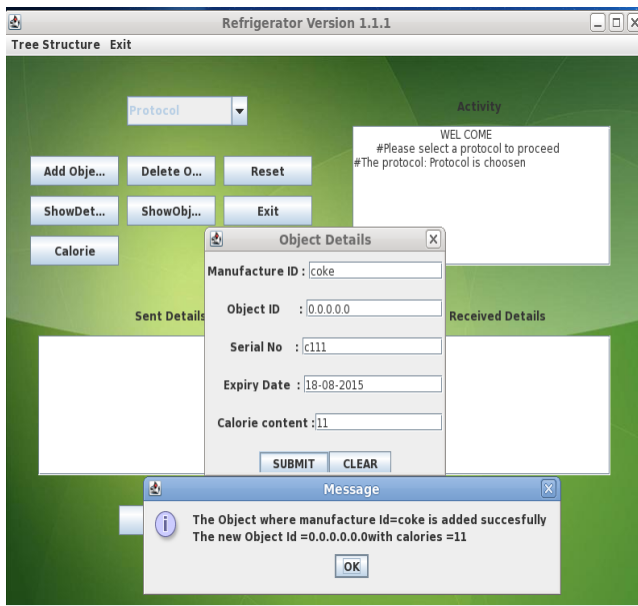


Fig 11 : G.U.I for Refrigerator Object 1.1.1

Owner Object: The Owner object acts as an active object and it has to receive all the messages sent by the Refrigerator object. It contains many other classes to provide GUI for the Owner object. The Owner objects can also send various messages to the Refrigerator such as GetCount-to gives the number of objects of a specific type present in the Refrigerator, GetAlternative-it gives an option for owner to get alternate object of a specific object type, GetObjects- it gives the all objects in the application, GetCalorie-it gives information about the calorie content of the specific type of the object, Submit Query- to submit the query typed. And it also shows the several activities taking place in the owner object. It submits the various queries to the Refrigerator and receives the result of the submitted query from the Refrigerator. Here assumed that user selected a protocol version 1.1.1. And simulated Owner Object GUI is shown in the figure 12.

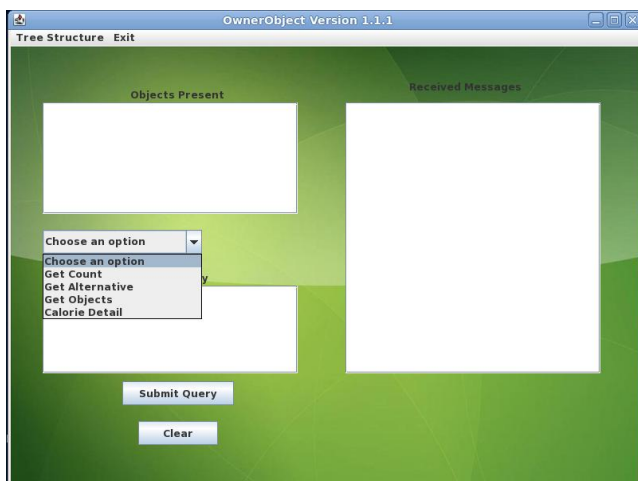


Fig 12 : G.U.I for Owner Object 1.1.1

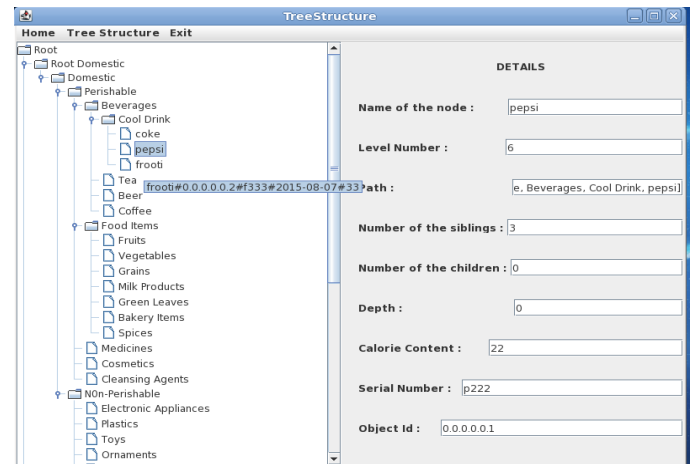


Fig 13 : Tree Structure Data Base Shows the Hierarchical Objects Categorization

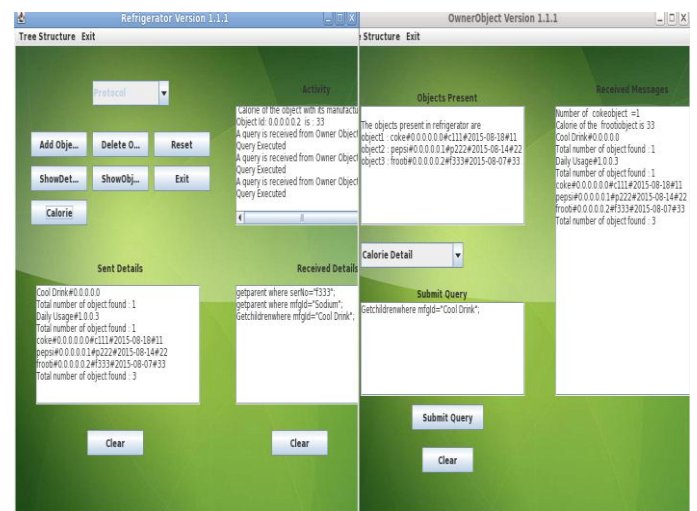


Fig 14 : Illustration of communication between Refrigerator and Owner Object

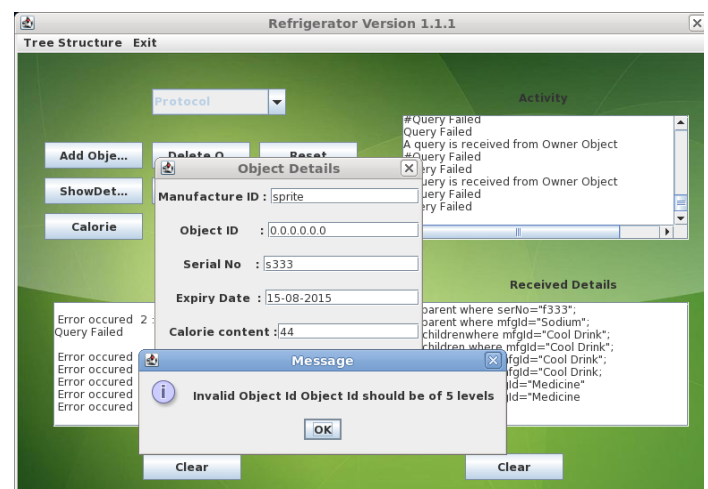


Fig 15 : Illustration of Object Categorization

Table 2. Illustrates Comparison of Different Versions of protocol

Protocol Versions	RFID Tag content	Tree Structure Levels	No. of Query Commands used
Protocol 1.0	4	5	9
Protocol 1.1	5	5	13
Protocol 1.0.1	4	6	22
Protocol 1.1.1	5	6	28

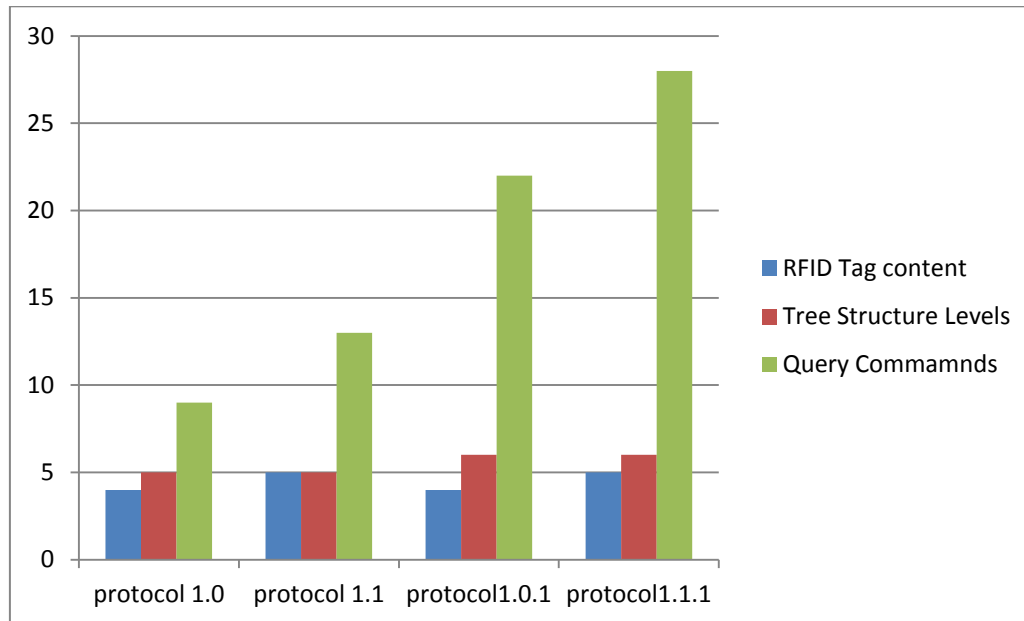


Fig 16 : Illustration of Comparision of Different Versions of Protocol

5.1 Syntax and Examples of Different Query Types

Different queries are used to establish the communication between ubiquitous object are as follows:

Q1: delete where mfgId="coke";
Q2: GETPARENT where mfgId="Medicine";
Q3: getDetails where mfgId="Fruits";
Q4: GetSiblingCount where mfgId="Domestic";
Q5: getsiblingdetails
Where
mfgId=
"Frooti"
;
Q6: getchildren where mfgId="Beverages";
Q7: getlevel where objId="0.0.0.1.0";
Q8: getDepth where objId="0.1.2.1.0";
Q9: getleafnodes where objId="0.0.0.1.0";
Q10: Getlevel where serNo="222";
Q11: GetDetails where serNo="111";
Q12: Getparent where
objId="0.0.1.2";
Q13: delete where serNo="333";
Q14: getlevel where serNo="444";
Q15: Getleafnodes

Where objId="0.0.0.0.1";
Q16: getsiblingdetails where serNo =
"321";
Q17:delete where expDate="11-09-2015";
Q18: getparent where serNo="abc";
Q19: getdetails where serNo =
"abc329";
Q20: delete
Where serNo=
"mno999";

6. CONCLUSION

Based on the above work and concepts developed, we infer that; the protocols designed facilitate interaction of basic details of objects (like id, name sId, expiry Date and calorie content). Protocols can be improved further by adding more programmable attributes like temperature, warranty date of the object, user registration, human health factor, etc. We can also make the system work only for certain temperature ranges. More query commands can be added to the Query Language to enhance the efficiency of interaction among the objects. Hierarchical categorization of the objects has several advantages. Along with unique identification code, but also facilitates the object awareness and their relationship. It also makes the system more scalable and intelligent.

7. FUTURE SCOPE

An improvement can be made with respect to the “Tag Structure”. Many other fields can be included like temperature, warranty date of the object, user registration, human health factor, etc. We can also make the system work only for certain temperature ranges. More query commands can be added to the Query Language to enhance the efficiency of interaction among the objects. Java SDK – “Software Development Kit”, Eclipse and Operating System Linux is used for the implementation of presented work. Communication between Refrigerator object and Owner object window is of file to file communication in Java programming. This implementation can be further improved by implementing the communication between Refrigerator object and Owner object by using Socket programming

8. REFERENCES

- [1] Thyagaraju.GS, Umakanth P. Kulkarni, “SAP: Self Aware Protocol for Ubiquitous Object Communication”, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume 1, Issue-5, November 2011
- [2] U.P.Kulkarni et al “ Ubiquitous Object Categorization and Identity”, International Conference on Computational Intelligence for Modeling Control and Automation, and International conference on Intelligent Agents, Web Technologies and Internet Commerce(CIMCA-IAWTIC“ 06)- IEEE2006
- [3] U.P.Kulkarni,etal “Query Language Interface for Ubiquitous Objects” , International Conference on Computational Intelligence and Multimedia Applications 2007,IEEE , DOI 10.1109/ICCIMA.2007.331
- [4] Anthony D. Joseph, Almudena Díaz, Pedro Merino, F. Javier Rivas, Umakant P. Kulkarni, J.V.Vadavi, G.S. Thyagaraju, S.M. Joshi, and A.R. Yardi, “Mobile and Ubiquitous Objects” ,IEEE pervasive computing journal ,vol5 ,No3, July–September 2006 ,wips ,pg 57-60.
- [5] C.M. Robert, “Radio Frequency Identification (RFID)”, Elsevier Journal: Computer & Security, pp18-26-2006.
- [6] lex & yacc by John Levine,Tony Mason and Doug Brown