Collaborative Context Management and Selection in Context Aware Computing

B. Vanathi and V. Rhymend Uthariaraj

Ramanujan Computing Center, Anna University Chennai,
Chennai, Tamil Nadu, India
mbvanathi@yahoo.co.in, rhymend@annauniv.edu

Abstract. Computing and computing applications are merged into surroundings instead of having computers as discrete objects are the objective of pervasive computing. Applications must adjust their behavior to every changing surroundings. Adjustment involves proper capture, management and reasoning of context. This paper proposes representation of context in a hierarchical form, storing of context data in an object relational database rather than an ordinary database and selecting the context using heuristic pruning method. Semantic of the context is managed by Ontology and context data is handled by Object relational database. These two modeling elements are associated to each other by semantics relations build in the ontology. The separation of modeling elements loads only relevant context data into the reasoner. This influences the only limited amount of context data in the reasoning space which further improves the performance of the reasoning process.

Keywords: Context Aware Computing, Pervasive Computing, Ontology, Object Relational DataBase.

1 Introduction

Advancement in computing application is due to the evolutional growth of distributed middleware. The integration of mobile clients into a distributed environments and the ad-hoc networking of dynamic components are becoming important in all areas of applications. The continuing technical progress in computing and communication lead to an all-encompassing use of networks and computing power called ubiquitous or pervasive computing [1]. Pervasive computing system targets at constantly adapting their behavior in order to meet the needs of users within every changing physical, social, computing and communication context. Pervasive devices make ad-hoc connections among them and may be connected to different types of sensors to capture changes in the environment. In the evolution chain from centralized computing to pervasive computing as presented by [2] [3], Context awareness is at the heart of pervasive computing problems. Context can be defined as an operational term whose definition depends on the intension for which it is collected and the interpretation of the operations involved on an entity at a particular time and space rather than the inherent characteristics of the entities and the operations themselves according to Dey

N. Meghanathan et al. (Eds.): CCSIT 2011, Part III, CCIS 133, pp. 348–357, 2011.

& Winogards [4, 5]. The complexity of such problems increases in multiplicative fashion rather than additive with the addition of new components into the chain.

2 Existing Works

Data required for modeling are obtained from the applications using sensors. Sensors can be physical, virtual or logical sensors [6].Physical data are captured using physical sensors. Virtual sensors are software processes which refer to data context. Logical sensors are the hybrid of the physical and virtual sensors and are used to solve complex tasks. After collecting the data from the application, it has to be represented in a suitable way for processing. Various modeling approaches are introduced to support standardization of techniques to present context for productive reasoning in different application area. The major classifications of context management modeling approaches are *key-Value-Pair modeling, Graphical modeling, object oriented modeling, logic based modeling, Markup scheme modeling and Ontology modeling [3].* Key-Value –Pair modeling is the simplest category of the models. They are not very efficient for sophisticated and structuring purposes. It supports only exact matching and no inheritance. Graphical modeling is particularly useful for structuring. It is not used on instance level. Object oriented modeling has a strong encapsulation and reusability feature. Logic based modeling uses logic expressions to define conditions on which a concluding expression or fact may be derived from a set of other expressions or facts. Context is defined as facts, expressions and rules and has a high degree of formality. Markup schema modeling uses standard markup languages or their extensions to represent context data. Ontology based models use ontology and related tools to represent context data and its semantics. Ubiquitous computing systems make high demands on context modeling approach in terms of the *Distributed composition (dc), Partial validation(pv), Richness and quality of information (qua), Incompleteness and ambiguity (inc), Level of formality (for), Applicability to existing environments (app)* [3].Context model and its data varies with notably high dynamics in terms of time, network topology and source (*dc).*Partial validation (*pv*) is highly desirable to be able to partially validate contextual knowledge on structure as well as on instance level against a context model in use even if there is no single place or point in time where the contextual knowledge is available on one node as a result of distributed composition. This is important because of complexity of contextual inter relationships, which make any modeling intention error-prone. The quality of information delivered by a sensor varies over time as well as the richness of information provided by different kinds of sensors. Context model must support quality (qua) and richness indications. Contextual information may be incomplete, if information is gathered from sensor networks. Context model must handle interpolation of incomplete data on the instance level (*inc).*Contextual facts and interrelationships must be defined in a precise and traceable manner (for).A context model must be applicable within the existing infrastructure of ubiquitous computing environments (app). Summary of appropriateness of modeling approaches [3] is given in Table 1. Among all the modeling approaches, ontology based context model is more suitable for context aware computing.

Table 1. Summary of appropriateness of modeling approaches

Requirement	Approaches				
	Markup scheme	Graphical Models	OO models	Logic based	Ontology based
Distributed composition	+	-	++	++	++
Partial validation	++	-	+	-	++
Quality of information	-	+	+	-	+
Incompleteness/ ambiguity	-	-	+	-	+
Level of formality	+	+	+	++	++
Applicability	++	+	+	-	+
(Key: ++ Comprehensive + Partial - Limited or none)					

2.1 Ontology Based Context Modeling

Ontology is defined as explicit specification of a shared conceptualization [5].Context is modeled as concepts and facts using ontology. Some context aware systems that use these approaches are discussed below.

2.1.1 Context Ontology (CONtext Ontology)

CONON [7] is based on treatment of high-level implicit contexts that are derived from low-level explicit context. It supports interoperability of different devices. CONON defined generic concepts regarding context and provides extensibility for adding domain specific concepts. Logic reasoning is used to perform consistency checks and to calculate high-level context knowledge from explicitly given low-level context information. CONON consists of an upper ontology .It is extended by several domain specific ontology for intelligent environments such as home, office or vehicle. The upper ontology holds general concepts which are common to the sub domains and can therefore be extended. CONON is implemented using OWL. Context reasoning in pervasive environment is time-consuming but is feasible for non-time-critical applications. For time-critical applications like navigation systems or security systems, the data size and rule complexity must be reduced. Context reasoning is dissociated from context usage. A server does the reasoning and tiny devices like mobile phones get the pre-calculated high-level context from the server for direct use. This is an infrastructure based environment.

2.1.2 CoBrA-ONT

CoBrA-ONT [8] is a context management model that enables distributed agents to control the access to their personal information in context-aware environments. CoBrA-ONT is a collection of OWL ontology for context-aware systems. CoBrA-ONT is designed to be used as a common vocabulary in order to overcome the obstacle of proprietary context models that block the interoperability of different devices. Semantics of OWL are used for context reasoning. CoBrA-ONT is central part of CoBrA, broker-centric agent architecture in smart spaces where it supports context reasoning and interoperability. The center of this architecture is context broker agent, which is a server that runs on a resources rich stationary computer. It receives and manages context knowledge for a set of agents and devices in its vicinity, which is the smart space. Agents and devices can contact the context broker and exchange information by the FIPA Agent Communication Language. The architecture of CoBrA-ONT is based on the upper ontology and domain specific ontology that extends the upper ontology .CoBrA-ONT is defined using the Web Ontology Language (OWL) to model the counts of people, agents, places and presentation events. It also describes the properties and relationships between these concepts .CoBrA-ONT depends on the assumption that there always exists a context-broker server that is known by all the participants. CoBrA is infrastructure-centric and is not for pervasive computing whereas the platform proposed in this work is mobile device-centric, where no additional equipment for a mobile device itself is required for system operation.

2.1.3 SOUPA

SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications) [9] is designed to model and support pervasive computing applications. The SOUPA ontology is expressed using the Web Ontology Language OWL and includes modular component vocabularies to represent intelligent agents with associated beliefs, desires and intension, time, space, events, user profiles, actions and policies for security and privacy. SOUPA is more comprehensive than CoBrA-ONT because it deals with more areas of pervasive computing. It also addresses CoBrA-ONT because it deals with more areas of pervasive computing and also addresses problems regarding ontology reuse. The SOUPA sub-ontology maps many of its concepts using owl: equivalentClass to concepts of existing common ontology.

2.1.4 GAS

GAS ontology [10] is ontology designed for collaboration among ubiquitous computing devices. The basic goal of this ontology is to provide a common language to communication and collaboration among the heterogeneous devices that constitute these environments. The GAS Ontology also supports the service discovery mechanism that a ubiquitous computing environment requires.

2.1.5 Limitations

Context aware systems are based on ad-hoc models of context, which causes lack of the desired formality and expressiveness. Existing models do not separate processing of context semantics from processing and representation of context data and structure. Ontology representation tools are suitable for statically representing the knowledge in a domain. They are not designed for capturing and processing constantly changing information in dynamic environment in a scalable manner. Existing ontology languages and serialization formats are test based (xml/rdf/owl) and not designed for efficient query optimization, processing and retrieval of large context data. The main drawbacks of pure ontological approaches are low performance and data throughput.

3 Advantages of rdbms and ordbms

Relational models provide standard interfaces and query optimization tools for managing large and distributed context database or receive and send notification on context changes. Relational models are not designed for semantic interpretation of data. Relational database alone cannot be used to represent context in a pervasive environment. For semantic interpretations, ontology is used along with relational database. The table 2 below summarizes the appropriateness of both approaches in relation to the necessary features. Both approaches have strong and weak sides with respect to features for context management modeling. Best of two worlds are combined to form a hybrid context management model.From the Table 2 both relational approach and object relational approach are in the same level. Object Relational Approach is more suitable than Relational approach because of the following advantages: Object relational database supports several storage units like collection list, arrays, types and UDTs (User defined data types) and most of them are represented as objects arrays. Object relational approach ensures large storage capacity, which is an important part in web based development. The access speed is fairly quick. Object relational database have a massive scalability compared to relational approach. Object relational database boast excellent manipulation power of object databases. It supports rich data types by adding a new object-oriented layer. The systems are initially implemented by storing the inactive context to a relational database and active context to an ontology. Then the response time to get the relevant time is noted. Further system is implemented by replacing the storing of inactive context to relational database by object relational database. Then appropriate service can be provided to the user using service discovery [12].

Table 2. Advantages of rdbms and ordbms

Necessary Feature	Relational Approach	Ontology Approach	Object Relational Approach
Semantic Support	No	Yes	No
Ease of transaction (large data)	Yes	No	Yes
Query optimization	Yes	No	Yes
Reasoning support	No	Yes	No
Formality	Yes	Yes	Yes
Scalability	Yes	Yes	Yes

4 Proposed Work

The block diagram of the proposed context aware system has three layers is shown in Fig. 1. They are layers are context acquisition layer, context middleware and application layer.

Context acquisition layer gathers the context from the environment using sensors, active badges, camera, Wi-Fi devices etc. Context middleware has three components.

They are representation layer, context management layer and decision layer. Context representation layer represents context as entity relation hierarchy form. In the context management layer context is stored in the object relational database and rules are either defined or learned.

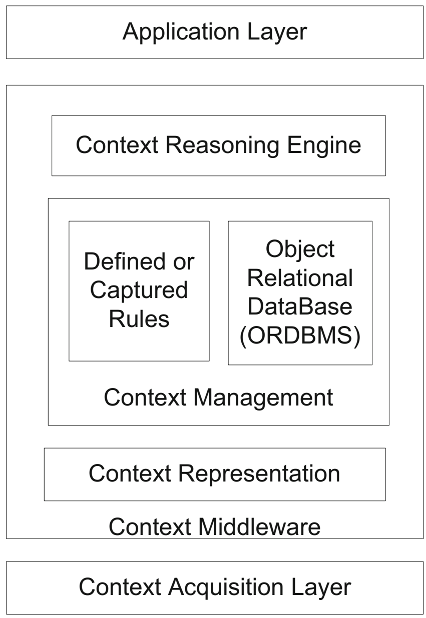

Fig. 1. Block Diagram

The pseudo code for filtering the context is shown below:

```
While (System-is_Running)
{
        newContext=null
        ContextBlock=new ContextClass()
        repeat until newContext=null
        {           newContext=ContextBlock.getNewContext()         }
        reliabilityFactor=0
        if (ContextBlock.hasRelaibleSource() { reliabilityFactor=1} else
         { reliabilityFactor=ContextBlock.estimateSourceRelaibility() }
if (reliabilityFactor>ContextBlock.reliabilityThreshold())
{ if (ContextBlock.hasStaticCategory()){ContextBlock.addContextToORDBMS()}
Else { ContextBlock.inValidContextError()}}
```

Context selector uses historic and current user information, devices availability, institutional policies etc. to select and load only part of context to reasoning space. Rules come from three different sources. They are rules defined by user, rules derived from organizational policies and rules derived from history data of past decisions using rule mining.

4.1 Context Representation

Context can be represented as Entity, Hierarchy, Relation, Axiom and Metadata [11]. Set of entities for which context is to be captured is called as Entity. Set of binary relations that form an inversely directed acyclic graph (inverted DAG) on entities is hierarchies. *Node*s of the graph represent entities and *arcs* of the graph represent hierarchical relations. The root entity at the top of the hierarchy graph is a global entity known as *ContextEntity*. Union of the sets of binary relations R_e and R_a stands for Relation. R_e is a set of binary relations having both its domain and range from the set of entity. R_a is set of binary relations defined from set of entities to set of literals representing entity attributes. Domain of the relation R_a is the set of entity while its range is set of literal values. Axiomatic relation (A) is a relation about relation. Relation can be generic or domain based. Relation can be transitive, symmetry or inverse property. Meta data (M) is a data about data. It is important in context modeling to associate quality, precision, source, time stamps and other information to the context data. Such information is important to prepare the context data for reasoning and decisions. In hierarchy representation metadata information is a relation that describes another relation instance. Hierarchy is an important structure to organize and classify context entities and relations into two layers. They are generic layer or domain layer. Layered organization is used to classify and tag context data as generic domain independent or as domain dependent. Context definition and representation is shown in Table 3.

Table 3. Generic and domain based relationship definitions

Generic level definition	Domain level definition
Person isEngagedIn Activity	Physician isEngagedIn Patient treatement
Location isLocatedIn Location	Library isLocatedIn Campus
Person isLocatedIn Location	Student isLocatedIn Library
Network hasDataRate yyy	ConnectionY hasDataRate low
Network hasDataRate yyy	ConnectionY hasDataRate low

4.2 Associating Hierarchical Context Representation to Relational Database

Relational database is a stable model that is used in a wide range of database management applications In relational database, entity relationship (ER) model is used to represent entities, attributes and relationships.A step-by-step mapping algorithm from the hierarchy components to relational schema is given as follows:

Step 1: Collect all context entities in the hierarchy model and create a relational table with attributes context entity, attributes that stores name of the entity one step above in the hierarchy (isa) and layer(generic or domain)
Step 2: Collect all non hierarchical relations (other than isa and isInstanceOf) in the hierarchy model and create a relational table with attributes Relations and Persistence (static/dynamic)
Step 3: Collect all relation instances in the hierarchy model and create a relational table with attributes Relation, Context Entity and Litral Value
Step 4: Collect all context instances in the hierarchy model and create a relational table with attribute entity instances and context entity

Step 5: Collect all relation defined on instances in the hierarchy model and creates a relational table with attribute entity instance, relation, value, timestamp, source and precision
Step 6: Collect all axioms in the hierarchy model and create a relational table with attribute relation, axiom and attribute value

4.3 Associating Hierarchical Conceptual Model to UML

Unified Modeling Language (UML) is used to formalize hierarchical as a conceptual context representation model.UML is a standard specification language for object modeling.UML is a general purpose modeling language that includes a graphical notation used to create an abstract model of a system. Entity in the hierarchical is represented as UML *Class.* The concept of hierarchical relation can be represented as *generalization* relationship in UML. Entity relations are represented using *attributes* in the UML class. Axiomatic relations are represented as *association classes* in the UML. The concept of metaclass can also be used to represent axiomatic properties like symmetric property, inverse property etc. Metadata is represented using association classes in the UML. UML is used as a tool for ordbms designing.

4.4 Purpose for Semantics

Consider the situation of staff members' (Ben ,Dan and Rita) tea break scenario in the following table, a simple query (select Subject from table.context where predicate="isLocatedIn" and Object="Room-305") select "Ben" as an output.

Table 4. Example on need for context semantics

Subject	Predicate	Object	Time
Ben	isLocatedIn	Room-305	2010022310
Dan	isLocatedIn	Room-301	2010022310
Rita	isLocatedIn	Office-305	2010022310
...	...	...	...

By common sense, terms "Office" and "Room" are synonymous in the domain of interest, the output must be "Ben" and "Rita" to the query. This example demonstrates the need for a context model that describes concepts, concepts hierarchies and their relationships. The concepts of *office* and *Room* in Table 3, *owl: sameAs* property that defines as the same concepts using OWL language as below:

```
<rdf:Description rdf:about= “#Office”>
        <owl:sameAs rdf: resource = “ #Room”>
<rdf:Description>
```

4.5 Associating Hierarchical Conceptual Model to Ontology

Ontology provides standardization of the structure of the context representation, semantic description and relationship of entities. E with class, H with subClassOf, superClassOf, A with transitive, inverseof and meta data with reification. Using ontology, deeper knowledge analysis is performed using domain specific rules. For example, to define the similarity axiom between the concepts *ownerOf* and *owns, owl:sameAs property is used.* Similarly, the symmetric axiom on the concept of *coLocatedWith* can be defined using *owl:symmentricProperty* and the inverse relationship property between *ownerOf and OwnedBy* can be defined using *owl:inverseOf Property*. Mapping between hierarchical model and ontology is shown in Table 5.

4.6 Heuristic Selection of Context

A reasoning space is a search space from which the right set of information is extracted to perform reasoning and inferences. A search space is a set of all possible solution to a problem. Uninformed search algorithms use the intuitive method of searching through the search space, whereas informed search algorithms use heuristic functions to apply knowledge about the structure of the search space to try to reduce the amount of time spent on searching. The entire set of context data is virtually organized into the hierarchical graph. The graph consists of hierarchical tree of context entities and their corresponding relations, axioms and metadata. Pruning techniques on the hierarchical tree of context entities in the graph is used to minimize the size of the reasoning space [14].

4.7 Metrics in Selection Process

Context entities are parameters from which the contents of the reasoning space(context data) are defined.

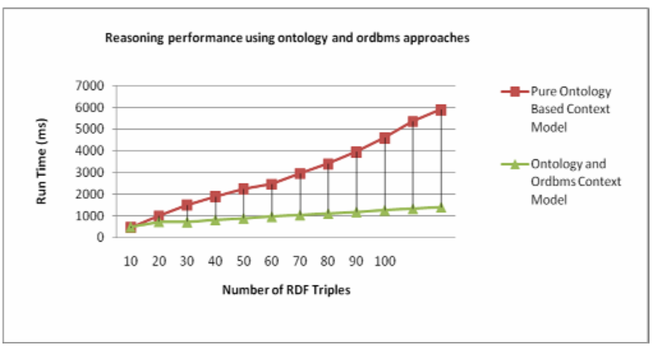

Fig. 2. Reasoning performance using ontology and ordbms approaches

Two measures of performance of the algorithm are accuracy(quality) of reasoning and the response time(speed).Context data are loaded into the reasoning space selectively by pruning the irrelevant part of the data. The context model using pure ontology and using ontology and object relational database is implemented.

5 Conclusion

Context is represented using layered and directed graph. Layered organization helps to classify and tag context data as generic domain independent or as domain dependent. A combination of context model using ontology and object relational database is proposed. This paper focuses on context representation and storage of context. Reasoning ,decision making of the context obtained from the context management are the future work.

References

1. Mattern, F., Sturn, P.: From Distributed Systems to Ubiquitous Computing- State of the Art. In: Trends and Prospects of Future Networked systems, Fachtagung Kommunication in Verteilten Systems (KiVS),Leipzig, Leipzig, pp. 3–25. Springer, Berlin (2003)
2. Satyanarayanan, M.: Pervasive Computing Vision and Challenges. IEEE Personal Communications, 10–17 (2000)
3. Strang, T., LinnhoPopien, C.: A Context Modeling Survey. In: The Proceedings of the First International Workshop on Advanced Context Modeling, Reasoning and Management, 6th International Conference on UbiComp 2004, Nottingham, England (2004)
4. Winograd, T.: Architectures for Context. Human-Computer Interaction 16(2-4), 401–419 (2001)
5. Dey, A.K., Abowd, G.D.: Towards a Better Understanding of Context and Context Awareness. In: Proceedings of the CHI Workshop on the What, Who, Where and How of Context-Awareness, The Hague, the Netherlands (2000)
6. Baldauf, M.: A survey on context aware Systems. Int. J. Ad hoc and Ubiquitous Computing 2(4), 263–277 (2007)
7. Wang, X., Zhang, D., Gu, T., Pung, H.K.: Ontology Based Context Modeling and Reasoning using OWL, Workshop on context modeling and reasoning. In: IEEE International Conference on Pervasive Computing and Communication, Orlando, Florida (2004)
8. Chen, H.: An Intelligent Broker Architecture for Pervasive Context-Aware Systems. PhD Thesis University of Maryland, Baltimore Country, USA (2004)
9. Chen, H., Perich, F., Finin, T., et al.: SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In: International Conference on Mobile and Ubiquitous Systems: Networking And Services, Boston, USA (2004)
10. Christopoulou, E., Kameas, A.: GAS Ontology: ontology for collaboration among ubiquitous Computing devices. International Journal of Human-Computer Studies 62(5), 664–685 (2005)
11. Ejigu, D., Scuturi, M., Brunie, L.: An Ontology Based Approach to Context Modeling and Reasoning in Pervasive Computing. In: 5th IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 14–19 (2007)
12. Vanathi, B., Rhymend Uthariaraj, V.: Ontology based service discovery for context aware computing. In:1st IEEE International Conference on Advanced Computing. IEEE Computer Society, Chennai (2009)
13. Vanathi, B., Rhymend Uthariaraj, V.: Context Representation and Management in a Pervasive Environmen. In: International Conference on Advances in Information and Communication Technologies, pp. 543–548. Springer, Heidelberg (2010)
14. Vanathi, B., Rhymend Uthariaraj, V.: Context Representation Using Hierarchical Method and Heuristic Based Context Selection in Context Aware Computing. In: Proceedings of the ACM-W 1st Conference of Women in Computing in India: Emerging Trends in Computer Proceedings (2010)