COMPARATIVE STUDY ON ONTOLOGY BASED MODELS IN CONTEXT AWARE COMPUTING

K.S.Sagaya Priya[1], Dr.Y.Kalpana[2]
#1Research Scholar, Dept. of Computing Science, Vels University, Chennai,Tamil Nadu, India.
#2 Associate Professor, Dept. of Computing Science, Vels University, Chennai,Tamilnadu, India.

Abstract: Ubiquitous computing is essential to illustrate a model for broad contextual information in surrounding environment. It also supports a mechanism of contextual information and provides relevant intelligent services. A better context information modeling formalism improves their maintainability and evolvablility and reduces the complexity of context-aware applications. Context modelling is most popular method among context-awareness service. Ontology is most popular modelling compare than the different method of context modelling. In this paper, discuss about ontology based modelling and examine different approaches of existing ontology-based context modeling like: CONON, COBRA-ONT, SOUPA and SOCAM.

IndexTerms- Context aware computing, context modeling, Ontology based models, CONON, COBRA-ONT, SOUPA, SOCAM.

I. INTRODUCTION

Nowadays, people are more attention for the context aware computing systems which may not only proactively adapt their behaviors to the user's current situation, however also protect them from being disturbed with different types of devices and services while on their standard duty. To recognize the context-aware computing systems, it has very important that various kinds of information from heterogeneous and diverse of sources must be pulled collectively to form a representation model which need to be agree on shared by all contributing devices to maintain interoperability. Additionally, context-aware computing systems can also perform reasoning over contexts which may guarantee to provide the quality of the context, deduce pass decisions and implicit information about the actions to be triggered.

The Context was described by Dey as "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."[2]. subsequently, the context has been complete to describe, the information processed and collected in different ways to determine whether the recall where and where to exceed. At this end, the context model is necessary technology to supply a high level of abstraction of the context. Context modeling is the requirement of all entities and relations between these entities which are desired to illustrate the context as a whole part, for example, any information on location, time, the user and current or planned activity, and computational entities. A particular problem for context modeling in distributed, heterogeneous environments is the proprietary use of representation ideas which hinder the interoperability of the various computational entities. The common use of the context ontologies can solve this problem in easier manner. In what manner to model the context information is a markup schema modeling, object-based modeling, graphical modeling, logic based modeling, key-value modeling, ontology based modeling, and it is categorized by data structure schema that would be better to exchange the context in each system. The ontology-based model have compared to these above modeling that sustain to semantic reasoning, by standard support, for allowing various expression of the context information. An ontology model is helpful to represent the daily life as a type of data structure utilizing computers. In recent times, the context-recognition frameworks have commonly applied ontology-based models and represent declarative expressions by using some standard ontology languages like Ontology Web Language (OWL), resource description framework (RDF) and others using tools that support ontology description [1]. In this paper, we discuss about the various approaches to context modeling depending upon OWL (ontology-based) like: CONON, COBRA-ONT, SOUPA and SOCAM**.**

II. ONTOLOGY BASED MODELING

To building pervasive context-aware systems, ontologies are key requirements independently developed sensors, devices, and agents that are expected to share contextual information, Knowledge and to provide appropriate services to users based on their requirements. Ontologies provide metadata schemas and controlled vocabulary of concepts, each with clearly defined and machine-process able semantics. By describing shared and general domain theories, ontologies help people and machines to communicate concisely, supporting semantic exchange, not just syntax.

The word ontology was derived from Philosophy, where it represents a systematic explanation. The description of ontology as follows: "Ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary". Ontology comprises not only the terms which are explicitly defined in it, however also the knowledge that may be inferred from it. Classifications of ontologies are considered as various levels of generality:

- **Domain ontologies:** Domain ontologies illustrates concepts for a specific domain (like biology or physics);
- **General ontologies:** General ontologies describe standard concepts, regardless of any task or particular domain (e.g. space, time, etc);
- **Application ontologies:** Application ontologies based on both the domain and the general ontologies; they describe concepts that are required for a specific application.

Different languages are used to describe ontologies, e.g. OWL Web Ontology Language [3, 4], Ontolingua [5] and LOOM [6]. OWL is depending upon the Resource Description Framework (RDF) and RDF Schema (RDF-S) [19], the Extensible Markup Language (XML), XML Schema [18]. It is divided into three expressive sub-languages OWL-Lite, OWL-DL and OWL-Full. OWL-DL provides frequently maximum expressiveness (in direct opposition to OWL-Lite) without losing computational decidability and completeness (in opposite to OWL-Full). The OWL Web Ontology Language is a requirement by the World Wide Web Consortium (W3C) and provides as a fundamental component of the Semantic Web initiative [11].

The majority of them have an extremely complex domain infrastructure; therefore we desire to explain one possible scenario about a smart fridge system in a smart home environment for recognizing a smart healthcare system in our routine life. There are many ontology based technique to recognize a context-aware framework for smart home environments [7], [8]. Given below briefly explain ontologies based modeling by using example (smart fridge system).

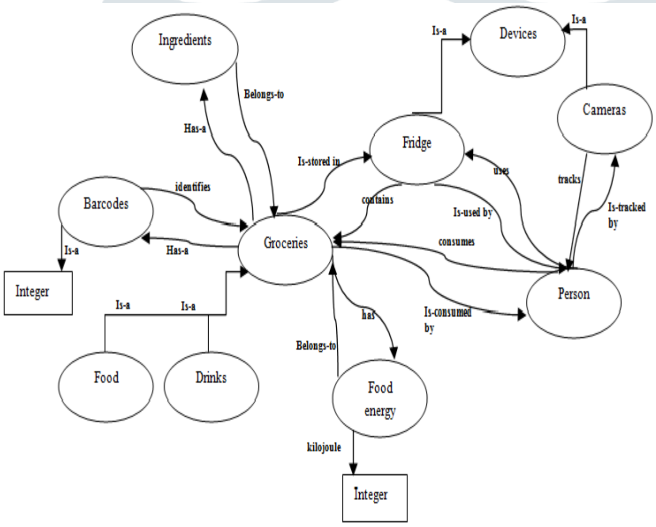

Figure 1 Principles of a fridge system in a smart home environment

The ellipses classify the classes or individuals and each arrow identifies a relation between these entities. The rectangles are inserted to show data-type values for completion process. In figure 1 describe that groceries are stored in the fridge system and the inversive relation "contains" provides the ontology some logic. In this application scenario the total system acquires data from various providers. Whole groceries can be recognized with a barcode, has ingredients and a food energy value. An additional important data source is the use of cameras or fridge that can analyze and track Bob when he is eating some groceries. The system need to set up a content data repository to store the usage of groceries from Bob. After a while the smart fridge system may analyze the habits of Bob and gives Bob some advices to eat something else, like if Bob is eating a lot of frozen-meal then the system could give him some tip's to eat a fresh salad or to cook something instead of eating frozen-meals.

The ontology based context models afford some advantages of interoperability and expressiveness. For example the ontology model acts as a middleware to serve a good interoperability between the different data information (barcodes, cameras, sensors within the fridge). Also this approach allows us to realize simpler representations of the system and it supports reasoning tasks.

It is possible to derive new knowledge about the current context and to detect possible inconsistencies in the context information [9]. For example, if the system knows that Bob consumes the groceries, the groceries are stored in the fridge, Bob uses the fridge then it is possible to know that Bob uses the fridge to consume his groceries. The realization of such a system, if expect this intelligent fridge system as a web connected system, would be realized with OWL based on the graph in figure 1. With OWL we define our classes, there properties and necessary ontologies. If we assume figure 1 as our starting point then the implementation of the classes, there properties and relationships are realized with OWL. Figure 2 shows such a possible OWL implementation of two classes, "Person" and "Device". With the tag "owl:Class" we define the classes which are necessary for our smart fridge. There are four possible tags to define properties of Classes and their relationships. In this sample OWL code we set up two properties for the class Person with "owl: DatatypeProperty" and it is possible to define relationships between these classes (figure 2, owl: ObjectProperty).

```
<owl:Class rdf:ID="person">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>

<owl:Datatypeproperty rdf:ID="age">
<rdfs:domain rdf:resource="#person"/>
<rdfs:range rdf:resource="htpp://www.w3/org/2001/XMLSchema#nonNegativeInteger"/>
</owl:DatatypeProperty>

<owl:Datatypeproperty rdf:ID="gender">
<rdfs:domain rdf:resource="#person"/>
<rdfs:range rdf:resource="htpp://www.w3/org/2001/XMLSchema#String"/>
</owl:DatatypeProperty>

<owl:Class rdf:ID="groceries">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>

<owl:ObjectProperty rdf:ID="consumes">
<rdfs:domain rdf:resource="#person"/>
<rdfs:range rdf:resource="groceries"/>
</owl:ObjectProperty>
```

Figure: 2. Owl code sample for a smart fridge system

III. Examples of Pervasive Computing Environments

The context modeling is based upon OWL (ontology-based) like: CONON, SOCAM, COBRA-ONT, and SOUPA.

A. CONON:

The CONON context modelling approach depends on the ASC/CoOL approach to improve a context model based on ontologies for the reason that its knowledge reuse capabilities, knowledge sharing and logic inference. In context aware computing, totally formalizing whole context information is likely to be an in-surmountable task. Most fundamental context for capturing the information we found that location, user, activity and computational entity to proceed the execute process. In pervasive computing environments, services and applications are generally grouped into a collection of sub-domains for utilizing at different intelligent environments (e.g., office, home or vehicle).

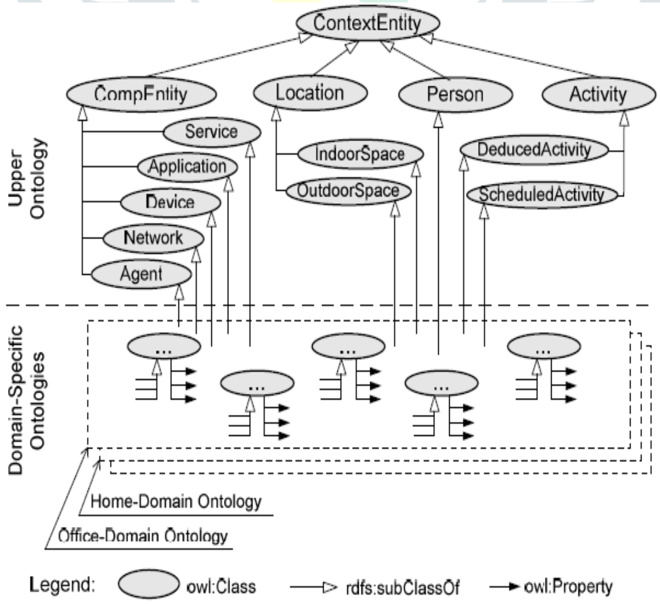

Figure 3. Partial Definition of CONON upper ontology

Context in each domain shares common concepts that may be modelled by using a general context model, even as differs significantly in detailed features. As a result, the separation of application domains encourages the reuse of general ideas, and provides a flexible interface for defining application-specific knowledge. We divide our context model into specific ontology and upper ontology. Specific ontology is a collection of ontology set which define the details of general features and ideas in each sub-domain. The upper ontology is a high-level ontology that captures contextual entities for utilizing common features.

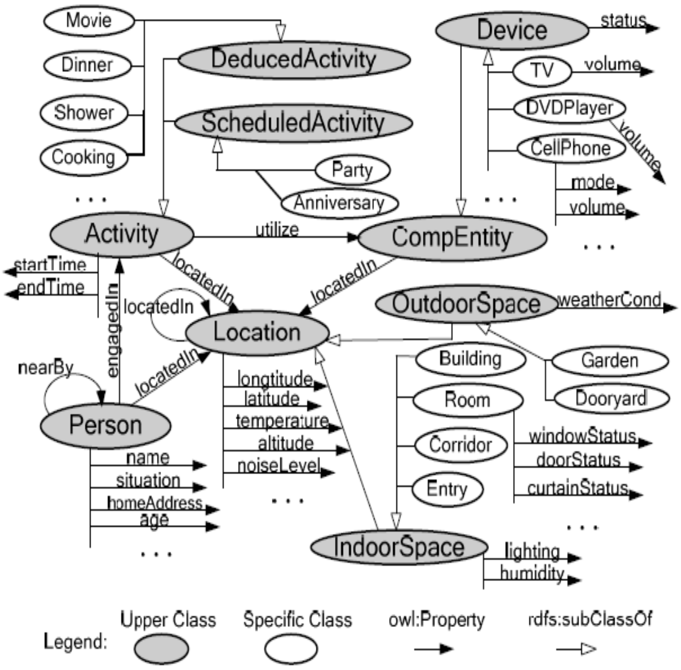

Figure 4. Partial definition of a specific ontology for home domain

Figure 3 demonstrates the upper context ontology (partial view of the OWL serialization show in figure 5). A set of abstract entities is structured by the context model; each entity is defining a conceptual or physical object that comprises *Activity*, *Person*, *Location and* Computational Entity (*CompEntity*), as well as a set of abstract sub-classes. Each entity is associated with its relations with other entities (represented in *owl:ObjectProperty*) and attributes (represented in *owl:DatatypeProperty*). A specific domain is required to add new concepts that allow hierarchically structuring sub-class entities by using OWL property *owl:subClassOf.*

The specific ontology partially defined for utilizing a smart home application domain as clearly shown in Figure 4. General classes defined in CONON upper ontology, a number of concrete sub-classes are defined to model specific context in a given environment (e.g., the abstract class *IndoorSpace* of home domain is classified into four sub-classes *Building*, *Room*, *Corridor* and *Entry*).

```
<owl:Class rdf:ID="ContextEntity"/>
 <owl:Class rdf:ID="Location">
   <rdfs:subClassOf rdf:resource="#ContextEntity"/>
 </owl:Class>
 <owl:ObjectProperty rdf:ID="longtitude">
   <rdf:type rdf:resource="FunctionalProperty">
   <rdfs:domain rdf:resource="Location">
   <rdfs:range rdf:resource="xsd:double">
 </owl:ObjectProperty> ...
<owl:Class rdf:ID="IndoorSpace">
   <rdfs:subClassOf rdf:resource="#Location"/>
   <owl:disjointWith rdf:resource="#OutdoorSpace"/>
 </owl:Class>
 <owl:ObjectProperty rdf:ID="locatedIn">
   <rdf:type="owl:TransitiveProperty"/>
   <rdfs:domain rdf:resource="#Entity"/>
   <rdfs:range rdf:resource="#Location"/>
   <owl:inverseOf rdf:resource="#contains "/>
  </owl:ObjectProperty>  ...
```

Figure 5. Partial OWL serialization of the upper ontology

B. COBRA-ONT:

COBRA-ONT is the central part of the CoBrA, “broker-centric agent” architecture in smart spaces where it assists interoperability and context reasoning [10]. The central part of this architecture is a broker-centric agent that server runs on a resource-rich stationary computer. It manages and receives context knowledge for a set of devices and agents in its proximity which is the "smart space".

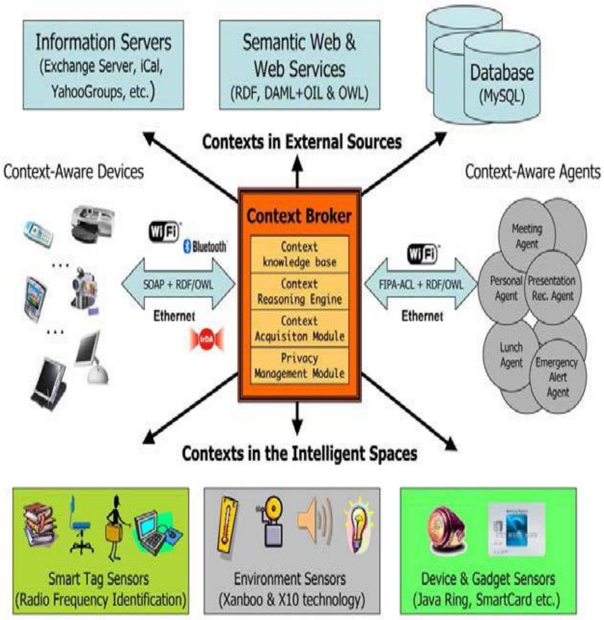

Figure 6: Context broker

It can perform reasoning to deduce high-level context information from inconsistencies detection process and the low-level sensor data analysis in the context information and monitor the users' security policies while passing user information to other agents. Devices and agents can contact the context broker and exchange information through the FIPA Agent Communication Language [10] and [11].

COBRA-ONT is considered into four sub-ontologies: Agent, Agent's Location, the Agent's Activity and Place whose relations and concepts and attributes are given in tables 1. The important parts of the ontologies will be analyzed in the given following and a more explanation can be found in [10].

Table 1: A complete list of the names of the classes and properties in the CoBrA ontology [10]

CoBrA Ontology Classes		CoBrA Ontology Properties	
"Place" Related	Agents' Location Context	"Place" Related	Agent's Location Context
Place AtomicPlace CompoundPlace Campus Building AtomicPlaceInBuilding AtomicPlaceNotInBuilding Room Hallway Stairway OtherPlaceInBuilding Restroom Gender LadiesRoom MensRoom ParkingLot	ThingInBuilding SoftwareAgentInBuilding PersonInBuilding ThingNotInBuilding SoftwareAgentNotInBuilding PersonNotInBuilding	latitude longitude hasPrettyName isSpatiallySubsumedBy spatiallySubsumes accessRestricted-ToGender lotNumber	locatedIn locatedInAtomicPlace locatedInRoom locatedInRestroom locatedInParkingLot locatedinCompoundPlace locatedInBuilding locatedInCampus
"Agent" Related	Agent's Activity Context	"Agent" Related	Agent's Activity Context
Agent Person SoftwareAgent Role SpeakerRole AudienceRole IntentionalAction ActionFoundInPresntation	PresentationSchedule Event EventHappeningNow PresentationHappeningNow RoomHasPresentationHappeningNow ParticipantOfPresentation-HappeningNow SpeakerOfPresentationHappeningNow AudienceOfPresentationHappeningNow PersonFillsRoleInPresentation PersonFillsSpeakerRole PersonFillsAudienceRole	hasContactInformation hasFullName hasEmail hasHomePage hasAgentAddress fillsRole isFilledBy intendsToPerform desiresSomeone-ToAchieve	participatesIn startTime endTime Location hasEvent hasEventHappeningNow invitedSpeaker expectedAudience presentationTitle presentationAbstract presentation eventDescription eventSchedule

The Place ontology is the concept Place with attributes like latitude and longitude to illustrate its location. It is the unique concepts of CompoundPlace and AtomicPlace. The CompoundPlace has a relation patiallySubsumes whose domain is Place. Besides relation places can be spatially "nested". Its converse relation is spatiallySubsumes. There are particular AtomicPlaces like Stairway, ParkingLot and Room. The CompoundPlaces are Building and Campus. An agent has attributes like an email address (hasEmail) or a name (hasFullName). Agents can be allotted with roles (Role) like the audience (AudienceRole) or speaker (SpeakerRole) during an event handle by fillsRole. The main role is used to provide detail about the intention of an agent because a role can be related to IntentionalAction by the relation intendsToPerform. On the other hand, IntentionalAction is not identified by COBRAONT. The Agent's Location ontology inserts the locatedIn associate to the Agent concept. LocatedIn points to Places. The relation specializes to particular locatedInCompoundPlace and locatedInAtomicPlace for performing AtomicPlace and CompoundPlace, subsequently. Two axioms can be stated here: an agent cannot be access at different AtomicPlaces at the same time, otherwise, an agent is at two different CompoundPlaces if and only if one of the compound places subsumes (spatiallySubsumes). These axioms permit to reason about the consistency of knowledge about the current location of an agent.

Moreover, the agents can be categorized corresponding to their location. In some case PersonInBuilding describes all people who are in a building whereas SoftwareAgentInBuilding describes whole software agents in a building. In these concepts, there are also focuses on their complements (e.g. PersonNotInBuilding and SoftwareAgentNotInBuilding). There are corresponding category concepts for all other specializations of Place. The Agent's Activity ontology represents events (concept Event) that happen at some places (relation hasEvent) and that are attended by agents (relation participatesIn).

The relation eventSchedule whose range is PresentationSchedule for expressing some events also scheduled. It has attributes examples for executing the start time, the title of the presentation and the end time. The relations expectedAudience and invitedSpeaker both with the range Person represent the audience and speaker of a presentation. During this concept PresentationEventHappeningNow includes whole presentations which are currently (now) taking Place according to the start and the end time. This concept can be defined that allow a person's current activity (e.g. SpeakerOfPresentationHappeningNow and AudienceOfPresentationHappeningNow) and also have currently occupied as a presentation place (e.g. RoomHasPresentationHappeningNow) [10].

C. SOUPA:

SOUPA, the Standard Ontology for Ubiquitous and Pervasive Applications [12] was developed by the same authors as COBRA-ONT. SOUPA is more comprehensive compare than COBRA-ONT for the reason that it deals with more areas of pervasive computing. The SOUPA sub-ontologies map many concepts through owl:equivalentClass to existing common ontologies concepts (like OpenCyc Spatial Ontologies [15], DAMLTime [14], FOAF [13], etc.) to enable interoperability with other ontologies. In SOUPA ontologies consists of two use-cases given that: CoBrA and its "EasyMeeting" prototype for smart rooms and MoGATU which is a peer-to-peer data management framework for pervasive environments. The SOUPA ontology consists of two parts: SOUPA extension and SOUPA core. The SOUPA extension holds additional ontologies for specifying domain vocabulary. The SOUPA core contains generic ontologies which afford a general vocabulary for computing different pervasive environments.

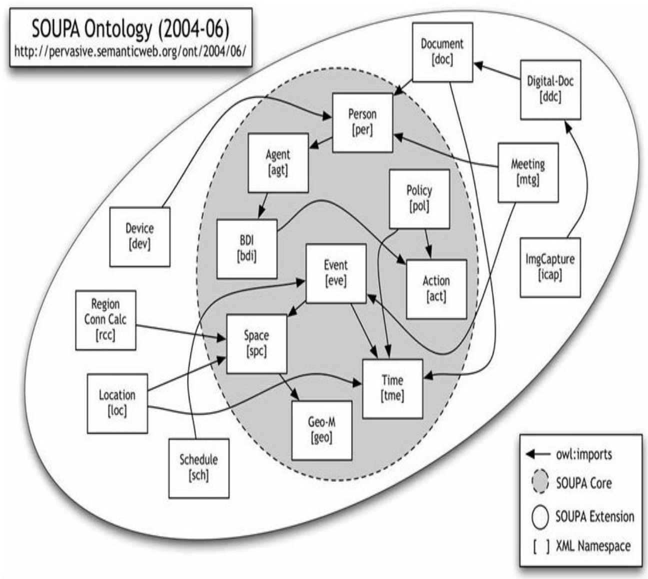

Figure 7: SOUPA Ontology

The SOUPA Person ontology affords typical concepts for providing a person's profile and contact information. The vital concept is Person details which is marked as equivalent to the Person concept in FOAF. It has represent attributes and relations for describing a person's basic profile (e.g. date of birth, name), contact information (e.g. contact number, email address) and professional and social relations.

The SOUPA Action and Policy ontologies afford concepts for privacy issues and security. Central concepts are Policy which enables to allow or forbid actions and Action (e.g. to publish location information).

The BDI ontologies and the SOUPA Agent describe agents (Agent) which are either computational entities or users. Agent-oriented technologies utilize the suggestion of intention, belief and desire. Intentions are the plans of the agent to reach goals in a perfect manner. Beliefs signify facts which are already known to the agent (and which are not necessarily true). Desires are main goals of the agent. Intention, Belief and desire are characterized by the concepts Intention, Fact and Desire of the SOUPA BDI ontology, correspondingly. An Intention consists of a set of Plans, which is a focused Action from the SOUPA Action ontology. A Fact is signified by an arbitrary RDF triple. Desire is unspecified through the BDI ontology. Additionally, an Intention has more preconditions and effects, represented by further not specified BDI ontology concepts Precondition and Effect.

The SOUPA Time ontology is advanced on the basis of the Entry sub-ontology and the DAML Time ontology of Time [16]. It represents concepts to temporal relations and express time, which are describe the temporal properties of events. The important concepts are TemporalThing, which signifies everything is temporal, and its sub concepts to describe TimeInterval and TimeInstant, which represent points and periods of time, correspondingly. For TimeIntervals there relations like startsSoonerThan and startsLaterThan. For TimeInstant there are many relations to represent in the order of two points of time, e.g. before and its inverse counterpart after, sameTimeAs.

The SOUPA Space and Geo Measurement ontologies, to make on the basis of OpenGIS and OpenCyc [17] provide concepts to represent spatial coordinates, geographical regions and relations and corresponding conversions. The vital concept is SpatialThing, which is utilized for everything that has spatial properties. The relation has Coordinates permits to represent SpatialThings with accurate coordinates (e.g. GPS). GeographicalRegions can be spatially nested through the spatiallySubsumed relation which is the inverse process of spatiallySubsumedBy. GeopoliticalEntity (e.g. USA) has the relation manages which points to GeographicalRegion (e.g. Alaska). The Geo Measurement Ontology provides typical geo-vocabulary, e.g. longitude, latitude, altitude etc.

The SOUPA Event ontology elaborates events which have both spatial and temporal properties. The Event concept describes schedules and activities and it's combined with TemporalThing and SpatialThing resulting in the concept of SpatialTemporalEvent. The SOUPA extension ontologies are made on the support of the SOUPA Core to maintain special application scenarios. The Schedule and Meeting ontologies for instance represent typical concepts in relation to their participants. The Digital Document ontologies that allows to describe Meta data for documents like creation date and size.

D.SOCAM:

In pervasive computing environments, the Service-Oriented Context-Aware Middleware (SOCAM) is an architecture that allows the rapid prototyping of context-aware services [18]. The SOCAM architecture utilizes the modelling context from the ontology-based model CONON (CONtext ONtology). The ontology-based context model describes a hierarchical approach for planning context ontologies. It provides ontology for general concepts in domain-specific ontologies and pervasive environments that concerns to totally different sub-domains [15]. The CONON approaches have two different reasoning types: user-defined reasoning and reasoning with description logic which consists of the components as follows.

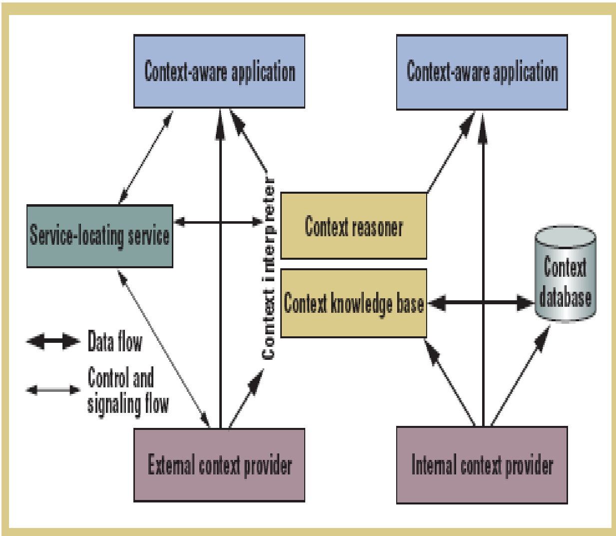

Figure 8. Overview of the SOCAM architecture.

Context Interpreter: Context Interpreter consists of Context KB (Knowledge Base) and Context Reasoning Engines. The context KB affords the service that other components can add, modify, delete or query context knowledge stored in the Context Database. The Context Reasoning Engines represent the context reasoning services comprising resolving context conflicts, inferring deduced contexts and maintaining the consistency of Context KB. Different inference rules can be indicated and input into the reasoning engines.

Context Providers: Context Providers abstract contexts from different basis - Internal or External Context Providers that can be converted into OWL representation so as to contexts were reused and shared through other SOCAM components.

Context-aware Services: The current context is performed according to the way of they adapt and contexts to make use at different levels by utilizing the Context-aware Services.

Service Locating Service: Service Locating Service provides a mechanism where the Context Interpreter and the Context Providers can publicize their presences; applications or users can access and locate these services.

SOCAM components are designed as independent service components. It also can be distributed over heterogeneous networks and interact with each other. All SOCAM components are implemented in Java, so that they are independent of underlying system platforms.

Modules / Properties	CONON	SOCAM	COBRA	SOUPA
Year	2004	2004	2004	2004
System Type	Middleware	Middleware	Framework	Framework
Context Model	Ontology	Ontology	Ontology	Ontology
Architecture style	Distributed	Distributed with Centralized server	Agent based	Component Based
Sensing type	Context Provider	Context Provider	Context Acquisition module	Context Acquisition module
Ontologies Language	OWL	OWL	OWL	OWL
Communication Model	Client/server	Client/server	Client/Server	Client/server
Implementation		Java RMI		
Scope of Environment	Single-domain	Single-domain	Single-domain	Single-domain
Security and Privacy	Available	Available	Available	Available

IV. CONCLUSION

OWL was introduced based upon different existing approaches. Ontologies can be utilized to help ontology reasoning and modelling by applying the common application areas for ontologies (such as logic inferencing, knowledge reuse and knowledge sharing,) to the domain of context in pervasive computing environments. SOUPA was developed utilizing the experiences from COBRA-ONT, and it has the more supports interoperability and comprehensive conceptualization by mapping concepts to other familiar ontologies. CONON illustrates that context reasoning is a calculation intensive job which makes it necessary to reduce the needed number of concepts. This is achieved by SOUPA and CONON by partitioning the conceptualization into different sub ontologies. And also the SOCAM architecture uses the ontology-based model CONON (CONtext ONtology) for modeling context. All approaches build use of the semantics of the ontology language to perform logic reasoning.

V. REFERENCES

[1]. Jonghun Kim & Kyung-Yong Chung," Ontology-based healthcare context information model to implement ubiquitous environment", Multimed Tools Appl (2014)

[2]. Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. "Towards a better understanding of context and contextawareness", In Handheld and ubiquitous computing pp. 304-307, 1999.

[3]. M. K. Smith, C. Welthy, and D. L. McGuinness: "OWL Web Ontology Language Guide", available online: http://www.w3.org/TR/owl-guide

[4]. S. Bechhofer, F. v. Harmelen, J. Hendler, I. Horrocks, et al.: „OWL Web Ontology Language Reference", available online: http://www.w3.org/TR/owl-ref

[5]. T. R. Gruber: "Ontolingua: A Mechanism to Support Portable Ontologies", Stanford University, 1992

[6]. R. M. MacGregor: "Inside the LOOM description classifier", in ACM SIGART Bulletin, 1991

[7]. Y.-G. Cheong, Y.-J. Kim, S. Y. Yoo, H. Lee, S. Lee, S. C. Chae, and H.-J. Choi, "An ontology-based reasoning approach towards energyaware smart homes," in Consumer Communications and Networking Conference (CCNC), 2011 IEEE, jan. 2011, pp. 850 –854.

[8]. J. Xu, Y.-H. Lee, W.-T. Tsai, W. Li, Y.-S. Son, J.-H. Park, and K.-D. Moon, "Ontology-based smart home solution and service composition," in Embedded Software and Systems, 2009. ICESS '09. International Conference on, may 2009, pp. 297 –304.

[9]. C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modeling and reasoning techniques," Pervasive and Mobile Computing, vol. 6, no. 2, pp. 161–180, 2010. [Online]. Available: http://www.sciencedirect.com/science

[10]. H. Chen, and T. Finin: "An Ontology for Context Aware Pervasive Computing Environments"

[11]. Feruzan Ay, "Context Modeling and Reasoning using Ontologies" University of Technology Berlin.

[12]. H. Chen, F. Perich, T. Finin, and A. Joshi: „SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications."

[13]. Hewlett-Packard Development Company: "Jena – A semantic web framework for Java.", http://jena.sourceforge.net

[14]. V. Haarslev and R. Möller: "RACER User's Guide and Reference Manual", http://www.sts.tu-harburg.de/~r.f.moeller/racer/racer-manual- 1-7-7.pdf, 2005

[15]. D. Brickley and L. Miller: FOAF vocabulary specification, in RDFWeb Namespace Document, RDFWeb, xmlns.com, 2003.

[16]. F. Pan and J. R. Hobbs: "Time in OWL-S", in Proceedings of AAAI-04 Spring Symposium on SemanticWeb Services, Stanford University, California, 2004.

[17]. D. B. Lenat and R. V. Guha: "Building Large Knowledge-Based Systems": Representation and Inference in the CycProject. Addison- Wesley, February 1990

[18]. "Resource description framework (rdf): Concepts and abstract syntax." [Online]. Available: http://www.w3.org/TR/rdf-concepts/