

Analysis of Advanced Dynamic Environmental Based Ontology Modeling In Context-Aware System

K.S. Sagaya Priya^{1*}, Y.Kalpana²

¹Department of Computing Science, Vels University, India.

²Department of Computing Science, Vels University, India.

*Corresponding Author K.S. Sagaya Priya, Department of Computing Science, Vels University, India

E-mail: sagayapriyaphd@gmail.com

Received date: September 17, 2021; Accepted date: December 15, 2021; Published date: December 24, 2021

Citation: Priya KSS (2021) Analysis of Advanced Dynamic Environmental Based Ontology Modeling In Context-Aware System. Am J Comput Sci Eng Surv. Vol.9, No.5.

Abstract

The context-aware workflow structure can adjust its behavior, mainly based on changes in the real world, which are generally collected by a large number of sensors. Therefore, context-aware workflow modeling is an important topic in the context-aware workflow structure. Actual context data is expected to be often diverse and complex. To better describe contextual data in business processes, this article provides a completely OWL (Web Ontology Language) -based method for modeling context-aware workflows. In this article we describe the design, shape, and implementation of a general, flexible, and extensible contextual ontology, called Advanced Dynamic Environment-Based Ontology Modeling (ADEOntoM), used for generalized computing systems and environments Context and Situation Modeling. The proposed device allows the discovery, composition and delivery of automated contextual insertion services. The expected ontology is implemented using JAVA, and finally SPARQL language (simple protocol and RDF query language) is used for evaluation. Finally, using context statistics of a specific size and comparing it with different context modeling tests, the proposed version is demonstrated. The final test results show that the proposed modeling method should provide extraordinary overall performance for the build version and the problem in the context device.

Keywords: Context-aware system; Environmental Based Ontology Modeling; RDF Query Language; Web Ontology Language; Pervasive Computing

Introduction

The context model enters the basic framework for modeling and representing context in popular computing environments and context-conscious structures. To support context and context awareness and entity modeling in general computing environments, it is very important to model and specify context and context in an appropriate way so that context statistics can be successfully exchanged, shared and reused. Context Toolkit [1] HP's Cooltown [2] At MIT, a format of a context-conscious

system was developed to demonstrate the usefulness of this new generation of AIRE spaces [3]. Of the context fabric [4] CoBrA [5] and GAIA [6]. However, context-sensitive products are not very convenient for customers on a daily basis. Building a contextual structure is a complex and time-consuming task, though, as appropriate infrastructure or middleware-level support is lost. A good infrastructure for situational awareness structures maximizes the associated responsibilities to handle situations that capture situations from several multiple assets, consisting of body sensors, databases, and agents. Need to do it. A description of the behavioral context; propagating the context to related events in a timely and assigned manner; providing a programming model for developing services in context. To resource these tasks, the context model wants to be properly mounted. In this article, we will version the ADEOntoM context model in the form of the Web Ontology Language (OWL) [7]. Approved through the W3C (World Wide Web Consortium) means, OWL is a group of relatives of a professional illustration language for the creation of ontology. Languages are characterized through formal semantics and RDF / XML means. Based entirely on serialization, primarily for the Semantic Web, [8]. Contexts written in the form of OWL can share effects. Whether or not they are within the same domain, one of the different products and differentiators can understand the context. Another important reason we choose OWL is that it executes reasoning. Some rule languages are supported for evaluating contextual facts in context models, and evaluation is another important step in context-aware computing. The main contributions of the work are:

- Propose and investigate an ontology based primarily on fully semantically rich, reusable and extensible context-controlled versions that enable the joint evaluation of multi-domain common context-aware software.
- Mainly focused on providing a guide of ontology fully programmatically in the smart space. The program can build the ontology and set the ratio through the API. The API provides items to the ontology engine. Aside from the main create/delete functions, this engine checks for changes in the ontology and provides additional guides to support maintenance.

- Context statistics stored in Ontology (with location) can be used by more than one program hanging on the same device, and there is no need to duplicate statistics between them.
- An adaptive selection method for selecting the best organ of a sensor that compromises the precision of energy intake and context popularity. The decision for the sensor agency is largely with the help of the use of the program, in the required context, the available sensors are fully adaptive, depending on the consumer's country and the type of power available. Will be changed to.
- The proposed version consists primarily of self-contained context cost types based on contextual representation. Context-primarily-based charge alignment is a prerequisite for semantic elements. It is based on a huge context that includes computable operations on the intention machine (eg service features, semantic distances of contexts). The task of determining the proximity between contexts.
- Build an ontology to improve forecasting and evaluation for the protection of pervasive computing.
- Characteristic Working Context at a Characteristic Level Framework to help share facts between popular programs: a context stage where a number of programs can percentage guessed context with the modern interest of the consumer. Or the uncooked facts stage where multiple programs can equally percentage uncooked facts extracted from common sensors.
- Devices with low processing energy and memory capacity may question distant devices that can store their expressive expertise base using ontology. Environment..
- Introduces a unified fact ontology that symbolizes a class rather than a special place in that fact unit, and a mapping ontology that associates that class with other identical classes.

Examples of pervasive computing environments

- Ontology has been employed for modelling contextual records in pervasive computing environments. Various context aware infrastructures in pervasive computing environments exist that benefit from ontology for representing and dealing with contextual knowledge. In this part, we referred to some present context aware systems to context modeling basing upon OWL (ontology-based totally) such as: CONON, SOCAM, COBRA-ONT, and SOUPA [9].
- SOUPA [10]: This ontology is one of the primary ontologies that attempted to model context in pervasive computing environments. It became advanced with the useful resource of the use of UbiComp Special Interest Group as brand new context ontology for developing ontology-driven pervasive and ubiquitous applications. The ontology developers defined a hard and fast of use times and designed the ontology based mostly on the essential aspect vocabularies driven from the use times. They discovered out that some of the SOUPA vocabularies can be positioned in extraordinary well-defined ontologies, and consequently accompanied part of the SOUPA vocabularies from those ontologies which consist of Friend-Of-A-Friend ontology (FOAF), DAML- Time, Spatial ontologies in OpenCyc, Regional Connection Calculus, COBRA-ONT, MoGATU BDI ontology, and Rei coverage ontology. SOUPA contains awesome ontologies: SOUPA Core which defines the overall phrases in not unusualplace domain names of pervasive computing, and SOUPA extensions outline extra standards for helping unique applications. SOUPA core represents nine distinctive ontologies related to person, agent, belief-desire-intention (BDI), action, policy, time, space, and event.
- COBRA-ONT [11]: This ontology changed into evolved for assisting context-conscious pervasive computing. The ontology consists of a fixed of ontologies for modeling context in clever environments. It is designed for use in broking-centric agent structure (CoBrA) in clever spaces. CoBrA is an agent-primarily based totally structure that allows expertise sharing and context reasoning. The vital a part of CoBrA is an agent known as a context broking which makes use of CORBA-ONT to hold a shared version and offer contextual reasoning. COBA-ONT is expressed via way of means of OWL and includes 4 predominant principles: Place, agent, and events. The region idea describes bodily places and extends to 2 principles of AtomicPlace as room and CompoundPlace as campus. Agent is any other pinnacle stage idea that's divided into precise principles: Person and SoftwareAgent. Each agent may have one of kind roles in an event.
- CONON [12]: CONON is likewise an ontology that tries to version contextual statistics in pervasive environments. This ontology is defined through OWL. CONON gives a proper context version for semantic illustration of contextual statistics. The ontology is split into ontologies: higher ontology and area ontology. Upper ontology describes number one and widely wide-spread entities that are standard in pervasive computing domain names. It has the power of extending into area ontologies. Domain ontologies outline information of standard standards in sub-domain names and describe their domain names with precise statistics. This separation encourages reusability of standard standards. One of the essential barriers in the back of growing context-conscious structures is the dearth of context-conscious infrastructure. The presence of a context version is crucial for context-conscious infrastructures. CONON changed into advanced to cope with this issue. Some context-conscious infrastructures use CONON as their context version. Service-Oriented Context-Aware Middleware (SOCAM) makes use of CONON as a proper context version. SOCAM is an infrastructure for producing context-conscious services. Similarly, CONON as a shared context version is implemented withinside the OSGi Infrastructure. OSGi is a context-conscious infrastructure for rapidly constructing context-conscious programs in clever domestic environments.
- SOCAM: The SOCAM structure offers an infrastructure for growing context-conscious offerings in pervasive computing environments [15]. SOCAM makes use of CONON as a shared context version which permits context reasoning to create context-conscious offerings. SOCAM structure incorporates 3 predominant components: context issuer, context interpreter, and context-conscious services. The context provider issue is liable for obtaining contexts from heterogeneous reassets and changing them into OWL representations. The context interpreter techniques context and infers high-stage context from low-stage context the use of context reasoning. Context-conscious offerings are agents, applications, and services that use contextual records provided via way of means of the

context issuer to react accordingly. Context-conscious offerings can accumulate contexts via way of means of appearing a question to the context issuer or via way of means of listening for occasions despatched via way of means of the context issuer. These offerings may be known as each time a fixed of policies turns into genuine via modifications in the contemporary context. The utility builders can sincerely broaden strategies which are invoked while the predefined policies in a condition are triggered.

Advanced dynamic environmental based ontology modeling (adeontom)

Generally the context aware computing shape is break up into five layers; Context Sensing Layer, Context Acquisition Layer, Context Modelling Layer, Context Inference Layer and Context Application Layer. Context sensing layer is responsible for collecting context facts and supplying a unified interface for adapting sensors. It moreover devices the initial facts processing cache module. The cause of Context Acquisition Layer is collecting and transmitting the static and dynamic context facts from various reassets and converts them to OWL instance just so contexts can be included and reused thru manner of way of various components of the platform. The acquisition of context is the idea to acquire the potential of context interest for an application. In this paper specially interest on context modeling inference and application layer.

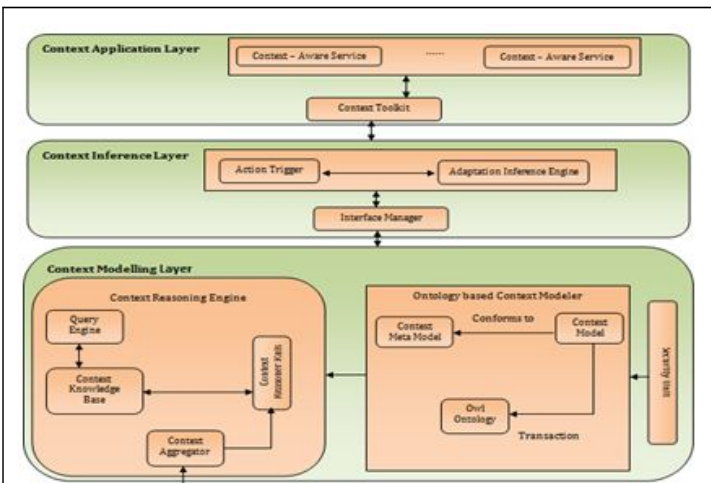


Figure 1: Architecture for Advanced Dynamic Environmental Based Ontology Modelling (ADEOntoM).

Context modeling layer

After obtaining contextual data through sensors, context-aware applications model the context information in order to fuse the various pieces of information together into a unified view.

Ontology based context modeller

Ontologies offer a proper illustration of information in a domain. They are very promising for modelling contextual data due to the fact that they permit context reasoning. As ontology has an excessive degree of formality, it is able to assist common sense reasoning strategies together with first order common

sense. Many context-conscious architectures and frameworks exist that use ontology for context modelling. It defines the seize segment and illustration of context. It consists of:

- A model of context in precise domain (Univ-Bench in our case), conformity with the proposed Meta model. This Meta model is defined in a common and precis way to be used independently from the application area.
- OWL context ontology generated from a model of context using models transformation based mostly on Univ- Bench.
- Context modelling is addressing the resource of the usage of Context Ontology Modeler. This tool is based mostly on paradigms, Model driven engineering in which models are used to strain software program application development existence cycle and ontologies to model semantic context data.
- Context Metamodel: To version context in widespread and summary way and primarily based totally at the paintings evolved in, we purpose to advocate a metamodel that defines the context and its sub context, context property, validity, and specs of every context property see Figure 2.

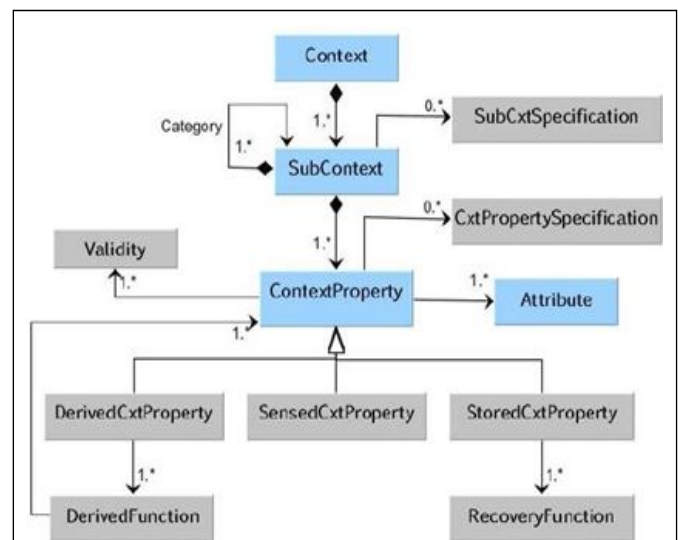


Figure 2: Context metamodel.

This metamodel is based on the following specifications:

- A context dissolve into sub contexts;
- A sub context can be, recursively, dissolve into lessons for its compose;
- A context, a sub context and a class are form of context homes;
- A context property is gathered through sensors: SensedCtxProperty, or derived from unique context homes: DerivedCtxProperty, or saved with inside the database: StoredCtxProperty.
- Each property has context validity, and context specifications.
- A derivedCtxProperty is obtained with the useful resource of the usage of derivation from a difficult and rapid of homes based definitely derivation Rule.
- A StoredCtxProperty is obtained through the healing features of stored homes (e.g. Recovery Rules)

Transformations Context Metamodel/OWL: Utilize the realistic formalism to determine planning correspondences

between our Context Metamodel and OWL Metamodel. The Context metamodel is introduced on the left side, the detail of correspondences (planning) in the middle and a passage of OWL metamodel on the right half of the figure 3.

- Source metamodel: To isolate the seize and representation of setting and semantic outline in our metamodel, we definite a view from the metamodel (see the left angle in parent 3). This view does now presently don't adapt to how setting realities is caught, anyway addresses handiest the setting in reliant and widespread manner. In the way of the offered change this view relates to the stock metamodel. These metamodel claims a Context, a SubContext, ContextProperty and Attributes related with each property.
- Metamodel: The OWL (Web Ontology Language is intended to be utilized through capacities that need to move toward the substance material fabric of records on the other hand of for all intents and purposes providing records to people. The OWL metamodel is gained with the guide of utilizing broadening the RDFS metamodel (see the appropriate aspect in Figure 3) this extract of metamodel demonstrates the class chain of importance found in owl.

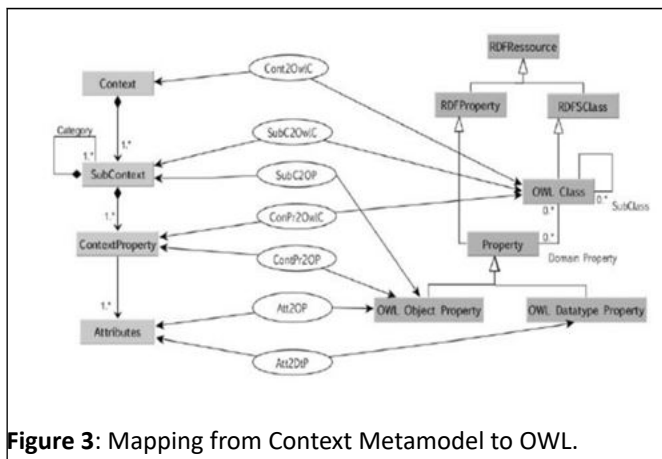


Figure 3: Mapping from Context Metamodel to OWL.

Transformation rules: In this section, we describe transformation rules presented in Figure 4:

- Transforms Context to Owl Class.
- Transforms SubContext to Owl Class.
- Transforms a Composition Relation among SubContext and Context to ObjectProperty and a connection SubClassOf.
- Transforms ContextProperty to Owl Class.
- Transforms a Composition Relation among ContextProperty and SubContext to ObjectProperty and a connection SubClassOf.
- Transforms ID characteristic to ObjectProperty.
- Transforms others credits to DataTypeProperty.

So we got an OWL ontology dependent on our context model, the acquired philosophy comprises of a bunch of classes, people, properties and relations that depict the different setting properties. This OWL ontology can be utilized for thinking by a inference engine.

Context reasoning engine reasoning

Ontologies are made to symbolize actual world know-how the usage of OWL, whilst the complicated trouble conditions want

extra description techniques. Context-conscious structures ought to be capable of carry out context reasoning to facilitate dynamic model to the converting environment, i.e. to be context-conscious. By manner of reasoning, setting irregularity might be checked, certain realities might be gotten, and the proactive lead of exceptional contributions might be provided. A more prominent adaptable thinking component is individual characterized thinking. Through the appearance of individual characterized thinking rules withinside the entailment of first request rationale, an immense scope of higher-stage, calculated setting which incorporate "what the individual is doing" might be concluded from pertinent low-stage setting. On the reason of our setting ontologies, the irregularity is exhibited by means of rule- essentially based thoroughly thinking provided through Hermit.

Contextual Information Aggregator: The contextual facts aggregator reads the primary ontology shape and inputs the incoming records to it. The incoming records has the values from the sensors and meters of the building. It is the premise of context-awareness. The primary ontology shape includes classes, residences, and summary declarations. After it consists of the incoming records, it is transformed right into a primary ontology version and loaded to the memory. The primary ontology version configures the subclasses, times, and records residences with actual incoming records. It has the relation among records and belongings. When an occasion takes place via the sensors and/or meters of the building, the contextual facts aggregator gathers the incoming records and inputs it to the loaded primary ontology version. The accumulated contextual facts is transformed into times or records kind residences of the magnificence described withinside the ontology version. They are used for reasoning and statement of relation. The item belongings defines the relation among times and records, among times and classes, and among times.

Context Reasoning Engine: permits battle choice of context interpretations, and reasoning and deducing new situations from OWL representations of context and inference policies. It permits inferring knowledge from a given ontology. Additionally, the OWL specification gives us with staying electricity assist whilst you maintain in thoughts that OWL permits storing each information accessible at modeling time and information on hand at runtime. Reasoning about Context consists on studying the repository of Context so you can derive low-degree context to High-degree one in step with the context model and the inference policies.

Context Knowledge Base: It stores the ontology and area policies. When a context accumulating unit converts contextual records into an OWL description, it shops them with inside the context ontology. The context ontology includes contextual know-how withinside the shape of descriptive logic policies and axioms. For example, "person A has desire B" is declared with inside the context ontology. Moreover, area inference policies with inside the shape of first order logic may be saved with inside the Context Knowledge Base. These policies have an IF-THEN-ELSE layout. For instance, if "person A" is "sitting" and the "TV" is "on" then "person A" is "looking TV" ble from: is a is "sitting" and "tool status" is "on" are

asserted with inside the ontology know-how base and area policies which include Rule1 underneath are saved with inside the area rule repository. `hasActivity (userA, SITTING) ^ DeviceStatus (TV, on) -> hasActivity (userA, WATCHING-TV)`

Context Query Engine: The context query engine is utilized to manage expressive queries from context conscious applications and to answer to context conscious packages bundles with refreshed inquiry impacts. The crucial enter is from the setting cognizant supplier for sending questions to the expertise base, while the imperative yield furthermore utilizes the unique situation – cognizant supplier to accomplish impacts from the ability base. Other than the essential capacity, it very well may be broadly used to conjure the setting induction motor. Two key abilities are characterized under: Handling inquiries from the context –conscious programming is the guideline adventure of the setting question motor. As noted before, our context –conscious middleware utilizes the skill base to keep setting in OWL design: its inquiry format is similarly stand-out from that of the customary SQL. We utilize an inquiry language, SPARQL, to recover and oversee records saved in the setting expertise base in OWL format. The Context Query Engine furthermore has some other capacity: conjuring the context –conscious engine. At the point when the product needs high-stage information base in OWL design. The Context Query Engine likewise has another capacity: invoke the context deduction motor. At the point when the application needs undeniable level setting, it's anything but a technique to invoke the context deduction motor to create the derived context.

Security unit: This module is vital to guarantee the security and protection of clients, yet many existing setting mindful structures don't consider this, and a few systems just give essential security components. To offer more customized types of assistance in setting delicate applications, gather and utilize more setting touchy data identified with client security. The worry behind the utilization is that once this data is spilled, it might make peril or danger related staff. For instance, the medical care climate includes a lot of private data, like clinical data, actual data, and individual inclinations, to help specialists and attendants settle on better choices. Along these lines, the connected security dangers ought to be limited. Steps ought to be taken to ensure security and protection, like appointing logical information to its particular proprietors. Client admittance to relevant information should be painstakingly controlled. Just approved clients can get to logical information that meets subordinate necessities, for example, time requirements and area limitations.

Context inference layer

Inference layer obtains the context information of object-oriented and ontology model and then infer and recommend related services by the inference engine, creates the application development packages through the event engine.

Interface Manager: Deals with a UI and interface between the stage and different modules explicit to areas of uses. It likewise has activity setting off measure contingent upon the particular application area in which the stage is utilized.

Adaptation Tool: To assemble a context-aware service, we desire to diagram the component for choosing a fantastic offer and the adequacy of its conduct dependent on the cutting edge context. These components will work with free coupling between the focal citation and it's anything but an even issue. These progressions are at last molded by the presence of conditions relevant to the context aware sevice. Along these lines, for our situation, the offer ought to have the option to progressively change its conduct dependent on the interesting conditions produced by the derivation motor by utilizing just applicable context oriented data. We propose a transformative variation instrument. Contingent upon the circumstance of the thinking engine and the age of the context aware, the instrument permits choosing the assistance to call and changing it. It additionally permits information logging.

Contextual Reasoning Engine: The Contextual Reasoning Engine is a significant piece of the setting mindful framework. It's anything but a savvy segment in setting touchy middleware, intended to get undeniable level settings from low-level settings straightforwardly recovered from setting assets, and to check setting consistency when new settings show up. The capacity of the surmising motor relies upon the circumstance and is called by various instruments (is this the importance here?). The induced outcome will be added to the information base and utilized by different segments. The logical thinking motor is comprised of two modules: the fundamental thinking module and the forecast module. These modules are liable for the two principle undertakings of the logical thinking motor. The fundamental thinking module is comprised of OWL Reasoner. Your fundamental undertaking is to check the consistency of the specific circumstance. The consistency check of the setting model is a significant activity, which is enacted by changing the data of the specific circumstance or the designer adjusting the fundamental model of information on the unique situation. OWL Reasoner is a reasoner incorporated into the OWL language, which can fill this need. In view of the philosophy model, it's anything but a bunch of legitimate portrayal rules to achieve this assignment. The shrewd module is comprised of three reasoners: rule reasoner, choice tree reasoner (DT), and covered up Markov model reasoner (HMM). The standard reasoner performs fundamental action thinking dependent on first-request sensible guidelines. The HMM reasoner is intended to take care of the issue of dynamic examination exercises. The DT reasoner gives a functional answer for future movement suggestions dependent on the choice tree calculation. These three reasoners, individually, give answers for the three parts of the work to recognize setting. In the derivation motor, the diverse deduction motors have distinctive call systems with the goal that they work at the right time. When the reasoner is called, the data base should be refreshed in like manner. This permit setting touchy applications to offer facilitated types of assistance dependent on deduced results

Context toolkit: The context toolkit is a toolkit supporting advancement of context-aware applications. It's anything but a stage to acquire context-sensitive data and send context data to context aware applications. Context Toolkit engineering is displayed beneath. It comprises of three fundamental segments: gadget, aggregator and translator. Setting widgets are accused of

recovering context from sensors. Context aggregators join numerous gadgets to gather the whole setting concerning given substance. Translators give undeniable level setting to applications dependent on low-level context from widgets.

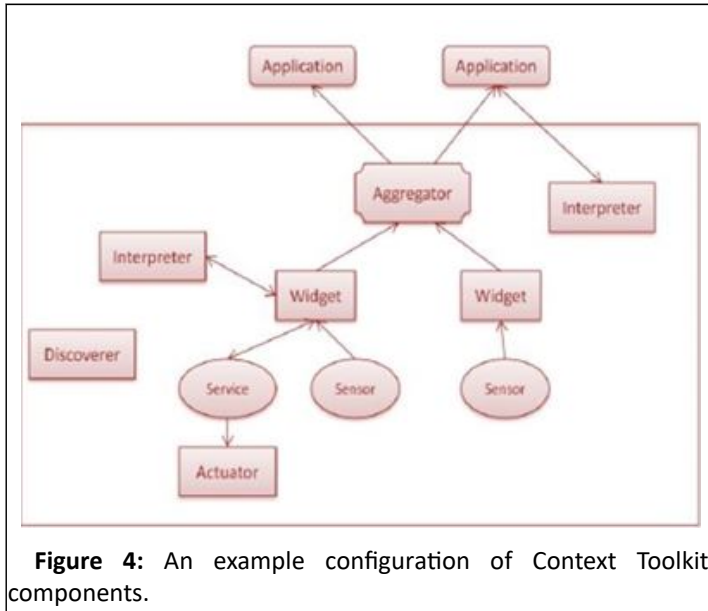


Figure 4: An example configuration of Context Toolkit components.

- **Context Widgets.** Generally talking, a context gadget is a product part that makes it feasible for applications to admittance to setting data. Clients are equipped for interfacing with the applications.
- **Interpreters.** This segment is intended to communicate setting in a higher unique level. It is extremely critical for the entire structure in light of the fact that most applications are anxious to get undeniable level setting data rather than crude data straightforwardly recognized from sending gadgets.
- **Aggregators.** Similarly as the name suggests, aggregators work for coordinating important setting data for improving on the huge number of setting data. Aggregators give a ton of helps to engineering to convey collected bits of setting to the particular application appropriately.
- **Context services context administrations** are liable for executing activities subsequent to utilizing the context data. They can be either coordinated or offbeat.
- **Discovers.** As the last part of this system, the module of finds is responsible for enlisting every one of the elements which are existing in the structure. They get the consciousness of arrangement of segments continuously. For instance, they understand what gadgets are created and accessible in the current circumstance. Also, finds can help track down a comparative segment to fill in for the unused one.

Context application layer

The Context-Aware Persuasive Application Layer (CAPAL) constitutes a set of persuasive purposes and agents that execute on the pervasive computing devices. They supply persuading offerings to the quit consumer and adapt their behaviour to the contemporary context. They provide a persuading surroundings that enhances bodily activity. For example, they can be a software module controlling a haptic actuator that produces comments involving the exercise intensity or progress, or a

media player that plays songs to encourage customers to exercising more. To deliver a persuasive service to the user, persuasive applications may additionally use offerings supplied by way of a persuasive carrier provider.

Persuasive functions use two methods to deliver context-aware services. The first technique is that application builders write a set of rules in the PSD unit to specify the conditions for triggering that service. Whenever the stipulations for that context turn out to be true, a context-aware persuasive service ought to be triggered. The PSD either notifies a persuasive application in the CAPAL to invoke the persuasive provider or notifies a PSP that subsequently invokes a context-aware persuasive carrier in the CAPAL. As an example, if a user sits for a lengthy time on a chair, the haptic actuator ought to give haptic feedback. A set of policies that specifies the context of jogging that provider such as "sitting for a lengthy time" is described in the PSD. When the context is met, PSD invokes the haptic service provider, and the issuer sends the command to the software that controls the haptic actuator in the CAPAL. The 2d approach is to have the persuasive applications or agents query the PSD to specify what services are suitable based totally on the current context and hit upon the PSP which affords the service. These functions can supply a set of persuasive offerings to the end user, and that's why they query the PSD to supply the most suitable one for the context. This layer consists of two components: Persuasive carrier Discovery (PSD) and Persuasive provider Providers (PSPs).

Persuasive Service Discovery: The Persuasive service Discovery (PSD) is responsible for choosing or triggering the most appropriate persuasive carrier according to the relevant context, and locating the provider company that provides the persuasive service. Various persuasive services are reachable in pervasive environments. To persuade a man or woman effectively, each persuasive carrier have to be provided based totally on the particularities of a given situation. Therefore, this unit analyses the applicable contexts using the context know-how saved in the Context Knowledge Base and chooses the most splendid persuasive provider via logic reasoning. For instance in a smart home, the PSD analyzes a user's degree of activity. If it determines that the person has been sitting and watching TV for more than three hours thru know-how inference, it concludes that a persuasive service giving focus about the user's sedentary reputation can be furnished to the consumer.

Persuasive Service Providers (PSPs): The Persuasive Service Providers (PSPs) issue is responsible for developing the persuasive services. Each PSP can run one or a set of persuasive services. Information about PSP units and the set of services they can offer is precise in the PSD component. The PSD component is used to identify the persuasive service most suitable for a particular context, and the PSP is used to grant that service. The company in this case can be an agent that creates messages with informative content about the user's sedentary behaviors. It sends the generated messages to the user's clever smartphone application that suggests the user's daily sedentary activity.

Experiment Result

The implementation and the validation of the logical and structural elements of ontology can be routinely realized with specialised tools. The Univ-Bench ontology is implemented in OWL using Protégé together with the Hermit 1.3.8 Reasoner. Protégé is one of the great regarded open supply editors to improve ontologies. Protégé has been allotted firstly for biomedical informatics lookup at the Stanford University School of Medicine. This tool is specifically dedicated to the OWL but it is a highly extensible editor, capable of handling a vast range of formats. Univ-Bench ontology used to be formalized the use of OWL DL due to the fact it is relatively expressive and as a result we can practice all popular automated reasoning techniques. Protégé consists built-in reasoners such as FaCT++, Pellet, Hermit, ELK, jcel, Ontop, Mastro and RACER used for describing logic. Hermit is reasoner for ontologies written the usage of the Web Ontology Language (OWL). Given an OWL file, Hermit can decide whether or not or no longer the ontology is consistent, identify subsumption relationships between classes, and a lot more. Hermit is the first publicly-available OWL reasoner primarily based on a novel "hypertableau" calculus which offers a lot extra environment friendly reasoning than any previously-known algorithm. Hermit uses direct semantics and passes all OWL 2 conformance tests for direct semantics reasoners. To evaluate how well ADEOntoM can scale compare to others, we used 5 LUBM datasets of varying sizes generated using LUBM data generator. The dataset are respectively LUBM 10M, LUBM 20M, LUBM 30M, LUBM 40M, LUBM 50M.

Test environment for evaluate the systems are: 1.80GHz Pentium 4 CPU; 256MB of RAM; 80GB of hard disk, Windows XP Professional OS; Java SDK 1.4.1; 512MB of max heap size.

Result Analysis

In section two, the five delegate five parts of context aware system frameworks are given, investigated, and looked at: structure, context portrayal, context detecting, context stockpiling, and context thinking. We picked the principle highlights for five viewpoints and contrasted our setting mindful middleware ADEOntoM and another two. The outcomes are displayed in Figure 5.

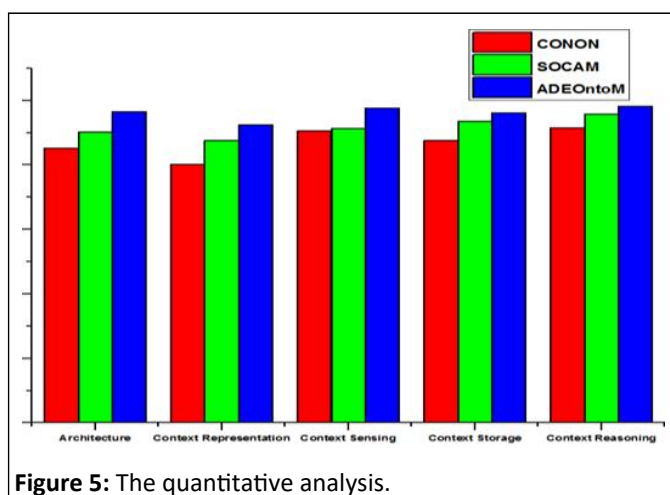


Figure 5: The quantitative analysis.

To be seen effectively, we award a quantitative investigation between the three setting mindful frameworks displayed in Figure 5. This examination depends absolutely on a technique that assesses each trademark as "feeble, medium or solid" addressed by the worth: "1, 2, 3", then, at that point summarizes all upsides of sub components as a definitive worth. SOCAM's setting representation is taken as an occasion to disclose how to process the element's worth. Setting portrayal comprises of three sub highlights: setting model, kind of setting, and chronicled support. In the setting model, it receives the philosophy based model that is considered as "Solid" addressed with the guide of value "3". It upholds five significant sorts of setting, which is the most extreme in six frameworks, considered as "Solid" addressed through value "3". It offers notable setting help saw as "Solid" addressed by means of cost "3". The entire expense of setting outline is "9", summarized through three sub highlights. In quantitative examination, ADEOntoM middleware performs appropriately with appreciate to engineering, helping setting reflection, extensibility and reusability in a top notch way. As for setting portrayal, ADEOntoM middleware embraces the philosophy based model and helps verifiable setting. With appreciate to setting detecting, ADEOntoM middleware performs better compared to others for the explanation that it helps additional sorts of compensation. With respect to setting stockpiling, the prevalent flow innovation (OWL) is utilized in ADEOntoM middleware, so its presentation is no more fragile than others. With appreciate to setting thinking; since we have planned an astute motor to help undertaking setting acknowledgment, it is tons higher than others. To close, our middleware recommends a right exhibition in 5 principal segments of setting mindful frameworks, and specifically in setting thinking.

Conclusion

In this paper delivered a complete context modelling framework for pervasive computing referred to as Advanced Dynamic Environmental Based Ontology Modelling (ADEOntoM). The test used to be carried out in a systematic and popular way by means of using the Lehigh University Benchmark. Compare our proposed ADEOntoM context-conscious machine with every other two consultant context-aware structures described in this paper. We tested those systems with 5 units of benchmark information and 14 extensional queries. To conclude, ADEOntoM shows correct overall performance foremost aspects of context-aware systems, and in particular in context reasoning.

References

1. Dey, A.K., Salber, D. Abowd, G.D., (2001) "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications", Journal of Human- Computer Interaction (HCI) 16:97-166
2. T. Kindberg and J. Barton, (2001) "A Web-based Nomadic Computing System", Journal of Computer Networks 35:443- 456,
3. Jason I. Hong and James A. Landay,(2001) "An Infrastructure Approach to Context-Aware Computing", Journal of Human-Computer Interaction 16:87-419

4. Harry Chen and Tim Finin,(2003) "An Ontology for a Context Aware Pervasive Computing Environment", IJCAI workshop on ontologies and distributed systems, Acapulco MX 18(3):193-291
5. Anand Ranganathan and Roy H. Campbell, (2003) "A Middleware for Context Agents in Ubiquitous Computing Environments", Journal of Proceedings of ACM/IFIP/USENIX International Middleware Conference: 143-161
6. Claudio Bettini, Oliver Brdiczka, Karen Henriksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan and Daniele Riboni, (2010) "A Survey of Context Modeling and Reasoning Techniques," Pervasive and Mobile Computing 6: 161-180.
7. H. Chen, F. Perich, T. Finin and A. Joshi, "Soupa: Standard ontology for ubiquitous and pervasive applications," journal of Mobile and Ubiquitous Systems: Networking and Services, the First Annual International Conference,258- 267.
8. H. Chen, T. Finin and A. Joshi,(2003) "An ontology for context-aware pervasive computing environments," The Knowledge Engineering Review 18:197-207.
9. X. H. Wang, D. Q. Zhang, T. Gu and H. K. Pung(2004) "Ontology based context modeling and reasoning using OWL," in Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference:18-22
10. T. Gu, H. K. Pung and D. Q. Zhang, (2005) "A service-oriented middleware for building context-aware services," Journal of Network and Computer Applications 28:1-18
11. T. Gu, H. K. Pung and D. Q. Zhang, (2004) "Toward an OSGi- based infrastructure for context-aware applications," Journal of Pervasive Computing 3:66-74
12. T. Gu, H. K. Pung and D. Q. Zhang, (2005) "A service-oriented middleware for building context-aware services," Journal of Network and Computer Applications 28:1-18