

International Symposium on Emerging Inter-networks, Communication and Mobility
(EICM-2014)

A Generic Layered Architecture for Context Aware Applications

Beza Mamo^{a*} and Dejene Ejigu^b

^aResearch and Innovation Unit, Kifiya Finacial Technology, bezamamo@yahoo.com, Addis Ababa, Ethiopia

^bDepartment of Computer Science, Addis Ababa University, ejigud@yahoo.com, Addis Ababa, Ethiopia

Abstract

As pervasive computing is at its preliminary stage a number of computing solutions were proposed and more are still under experimentation. In this paper a four layered generic architecture is proposed with a recommended components so as to support context awareness for pervasive applications. Detail of each layer is described sufficiently to insight the process of context generation from low level contextual data up to high level context reporting scheme in a given pervasive environment. The implementation section of this article describes how the context reasoning component employ different level of reasoning techniques such as knowledge acquisition, context based rule execution and the application of ontology in pervasive computing.

© 2014 Elsevier B.V. This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Selection and peer-review under responsibility of Conference Program Chairs

Keywords: Context Aware Applications; Context Architecture; Context Aware; Pervasive Computing; Context Reasoning.

1. Introduction

Ubiquitous/pervasive computing requires the integration of invisible devices, software standards, machine learning approaches, knowledge repository, etc. to produce a usable and distraction free context aware application. As pervasive computing is at its toddler stage, a number of approaches and architecture were proposed and used by many scholars^{1, 2, and 3}. To the best of our knowledge most of the architecture lack constancy and developed for specific pervasive environment assumed to be studied for the precise domain. Furthermore, components and layering of contextual data/information lack ample description, despite of update of the current state of art developments and tools in pervasive computing.

* Corresponding author. Tel.: ++251-0911-75-65-15.

E-mail address: bmamo@kifiya.com

This study proposed how the context aware application write-up process takes place from context acquisition up to context usage through an application layer.

A layered architecture provides tremendous benefits for the development of context-aware applications. A number of researches including^{4, 5 and 6} use a layered approach for the realization of a context-awareness. Some of the advantages that can be facilitated from a layered approach includes; increased life expectancy, mobility; value added features, modularity, innate plasticity, interoperability, greater compatibility, and better flexibility.

2. Proposed Architecture

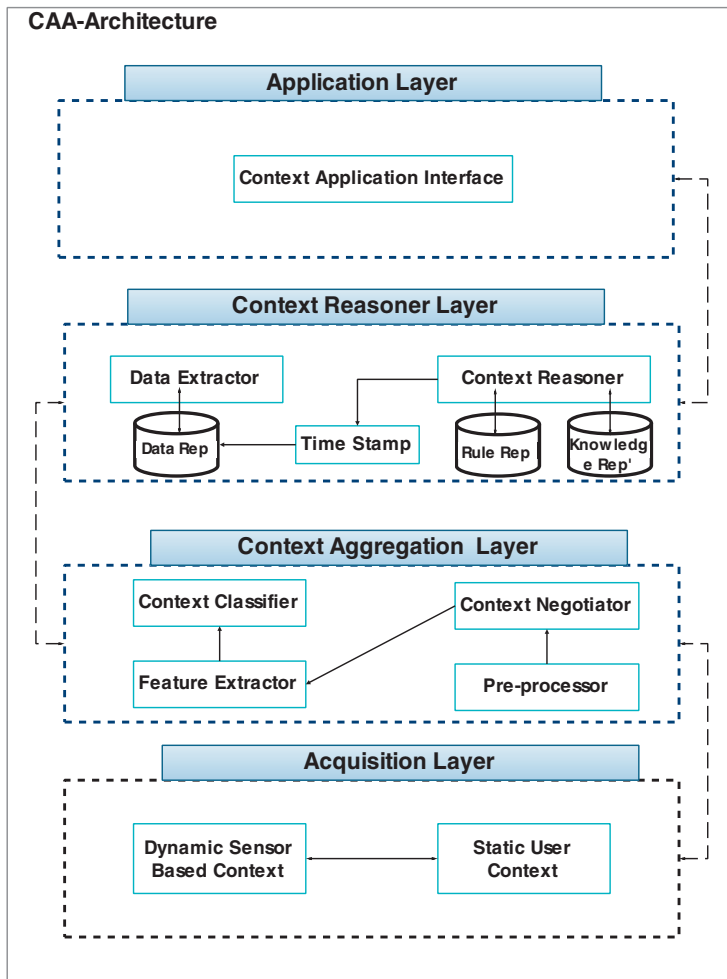


Fig. 1. The Proposed CAA Architecture

2.2 Context aggregation layer

The context aggregation layer comprise of four components described below:

The Preprocessor Component (PRC): This component binds the SUCC data with DSBC data dynamically. As any context is identified via sensors the PRC will make use of a decision tree algorithm provided by Weka (i.e., a machine learning tool) and suggest the possible meaningful context information like location.

Figure 1 depicts a layered Context Aware Application (CAA) architecture with detailed components. At every level of the recommended architecture, distinct feature of context information will be added so as to produce a usable context data/information.

2.1 Acquisition layer

The acquisition layer consists of two basic components namely; the Static User Context Component (SUCC) and the Dynamic Sensor Based Component (DSBC).

The SUCC: This component extracts user data from the hand held device in off-line mode by means of agenda, reminder, to-do list or any other application installed on the user device. The SUCC data is helpful for context aggregation, prediction, association and reasoning to be performed at higher level of the CAA.

The DSBC: This component extract context information dynamically from heterogeneous sensors scattered in the changing pervasive environment. The DSBC data can be segmented to location, activity, event, or any other user and environment based context information.

Details of the two components are explained in our previous publication⁷.

The Context Negotiator Component (CNC): This component is an essential component in context aggregation. Before dealing with feature extraction and other related activities in context management. It deals with context level agreement at low level so as to increase the validity and correctness of raw context data.

The Feature Extractor Component (FEC): This component finds the most relevant attribute of the preprocessed raw context. Precisely, the FEC deals with the characterization of raw context information in to activity, event, and resource capability in the user environment; and other pertinent features.

The Context Classifier Component (CCC): This component classifies the features extracted by FEC. At this stage medium level context data is generated by combining activity, location, and other context features supplied from the FEC. A new context class will be created as unique features are identified by FEC, or existing context classes are used to support context reasoning.

2.3 Context reasoner layer

This layer contains three components and three storage facilities. The components are: the Context Reasoner Component (CRC), the Time Stamp Component (TSC), and the Data Extractor Component (DEC).

The three storage facilities are: the Rule Repository (RR) the Knowledge Repository (KR), and the Data Repository (DR).

The CRC: This component takes the data from CCC-component discussed in section 2.2 above and produces a high-level context data in consultation with the RR and KR information. The Jena reasoner API was used to predict the exact activity of the user on the classified contextual data. The prediction or inference ability of the CRC, is highly facilitated via the ontology and rule generated for the purpose of this study to be discussed in later sections.

The TSC: This component stamps date/time information before the context data is stored permanently in the DR.

The DEC: This component interfaces the DR and the application. Security, privacy, encryption, and other computational implementation issue on the stored data can be facilitated by the means of this component.

The RR: Among the existing reasoning methods, this study uses a rule based reasoning approach. Rules and policies are viewed in the form of event-conditions-actions and dictate the behavior of the services in reacting to service invocation⁸. The rules that we use for our study will be discussed under the implementation part.

The KR: The knowledge repository stores the smart environment information using an ontology based representation. Ontology is about the exact description of things and their relationships and it is also a widely used tool for modeling context information⁹.

The DR: The Data repository contains important information about the user's activity, location and other contextual data annotated with a short description. Data/information in the DR is vitally important by low level context extractor components so as to enhance the context extraction processes.

2.4 Application layer

This layer contains a single component named as Application Interface Component (AIC) that runs the specific/desirable pervasive application on the user hand held device.

The AIC: this component allows user/developer to:

- provide/define the required user interface; and
- Identify the type of protocol/communication standard to be used such as HTTP, Bluetooth, WiFi, etc.

For this specific study we have used HTTP as a basic means of communication. Detail of CAA layer communication and context data flow is provided by the context state diagram (Figure 2).

3. CAA Implementation Detail of CAA Reasoner

Reasoning in CAA mainly implemented at the third layer as already discussed in Section 2. The capability of Jena ontoModel includes a simplified serialization of Resource Description Framework (RDF) together with rule based reasoning, which provides greater extent for the determination of user activity in our study. Thus, we describe how context ontology is developed to realize CAA; using TopBriad and combined with rules that we proposed in Section 3.2 by using the capability of the Jena Model Factory class. The subsequent sub-sections describe the CAA ontology and CAA rule implementation.

3.1. CAA deployment architecture

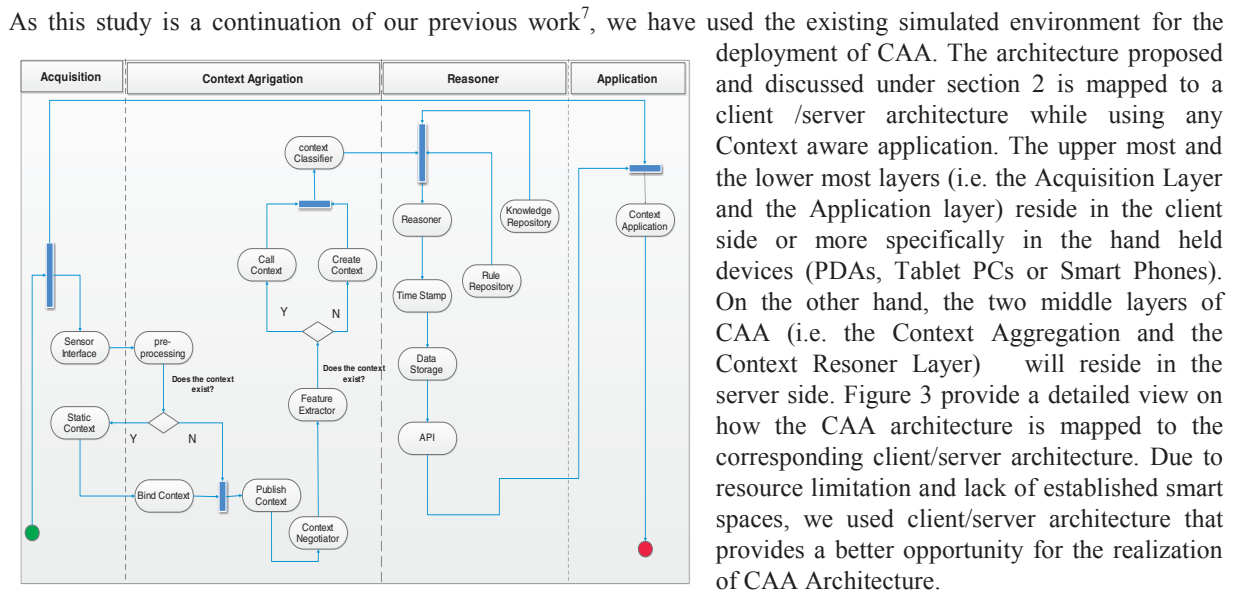


Fig. 2. Context state diagram of CAA from Acquisition to Application layer

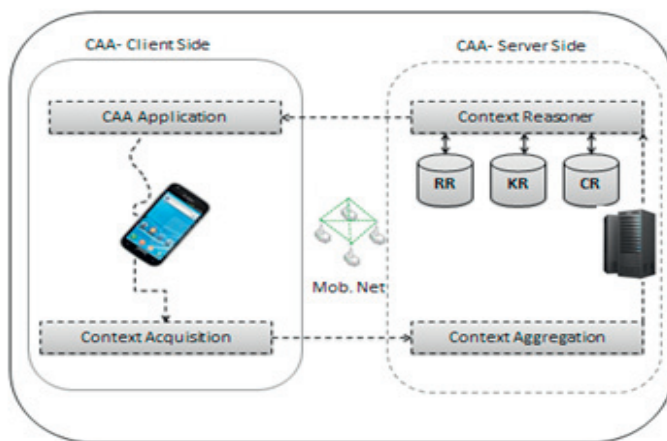


Fig. 3. Client Server Architecture for CAA

3.3. CAA Ontology Implementation

This section explains and considers a segment code of a class description of our ontology while deploying CAA together with its instances. Figure 5 contains two RDF OWL-ontology segment codes for CAA. Line 5 up to 9 contains a description of PDA456, which is an instance of PDA subclass that belongs to the Device class. The status and label of PDA456 descriptions are indicated at line 6 (on) and line 7 (PDA456), respectively.

On the other hand, line 10 onward describes Digital Library which is an instance of Indoor Location subclass that belongs to the Location class of CAA application. The label, default activity, and location name of the Digital Library resource description are indicated at line 12(Digital Library), Line 13(Study in Library) and line 14 (D-Library), respectively.

3.2. CAA knowledge representation

The generic context knowledge representation of CAA consists of Location, Device, Person, and Activity entities. The relationship Owns, LocatedIn, EngagedIn, and Undertaken are used for the interaction representation of the identified entities. Every generic entity is mapped to the domains and instances provided in figure 4. We used UML representation of EHRAM: Entity, Hierarchies, Relation, Axiom and Metadata model proposed and used by Ejigu et al,¹⁰ to elaborate the generic, domain and instances.

3.4. CAA Rule Implementation

The rules proposed for this study run over the ontology resource description OWL file given in figure 5. We have used user location rules and one activity assertion rule.

Scenario of user's location and activity rule descriptions used by CAA is given here under. The proposed rules for

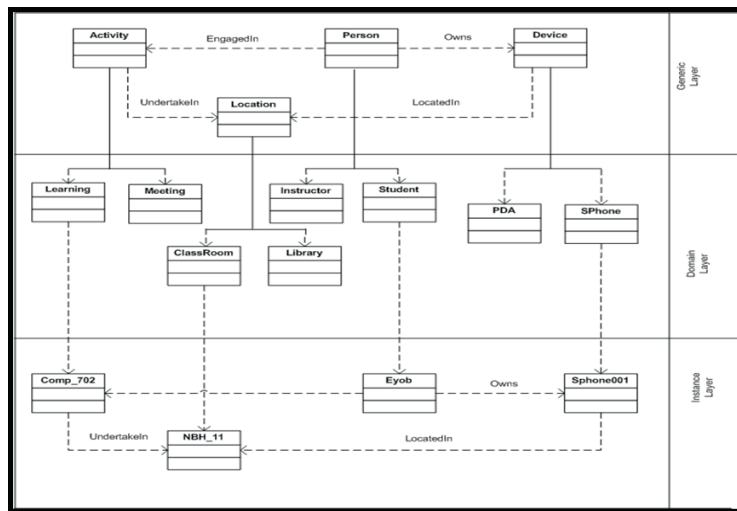


Fig. 4. UML Representation of EHRAM Model for CAA.

```

1.  <?xml version="1.0"?>
2.  <rdf:RDF
3.    xmlns:CAA="http://www.ContextAA.com/MyCAA.owl#"
4.    xml:base="http://www.ContextAA.com/MyCAA.owl" >
5.    <rdf:Description rdf:about="#PDA456">
6.      <CAA:hasStatus rdf:datatype="http://.../2001/XMLSchema#string">on</CAA:hasStatus>
7.      <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">PDA456</rdfs:label>
8.      <rdf:type rdf:resource="#PDA"/>
9.    </rdf:Description>
10.   <rdf:Description rdf:about="#DigitalLibrary">
11.     <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"></rdfs:label>
12.     <rdfs:label rdf:datatype="http://.../2001/XMLSchema#string">Digital library</rdfs:label>
13.     <CAA:DefaultActivity rdf:resource="#StudyInLibrary"/>
14.     <CAA:locationName rdf:datatype="http://.../2001/XMLSchema#string">DLibrary</CAA:locationName>
15.     <rdf:type rdf:resource="#Library"/>
16.   </rdf:Description>
17. </rdf:RDF>

```

Fig. 5. Segment Code of OWL Ontology used for CAA

4. Related Work

The work of ³, assume Model-View-Controller paradigm for the realization of context-aware application. The interaction model developed and used in this work considers the user along with the implicit/explicit interaction towards different devices; the different context as controller; and the context domain.

location and/or activity description then converted in to ontology files for context reasoning, as shown in figure 6.

User Location Rule Scenario:

- If a device D (i.e., PDA456) is held by a person P (Beza);
- If a device D is connected with another device d (i.e., PC-123, which is a static resource);
- If a device D is found in atomic location AL1 (i.e., Reading Room in the library);
- If atomic location AL1 is a subset of Location L1
- Then the rule concludes that P (Beza) is located in L1 (Library).

User Activity Rule Scenario:

- If a person P (i.e., Beza) owns a device D (i.e., PDA456);
- If PDA123 is connected with another device d (Projector-456 static resource);
- If the two devices D & d are found in the same set of atomic location AL2;
- If AL2 is subset of Location L1 (i.e., Room224, which is a class room);
- If L1 default activity is A (i.e., Learning);
- Then the rule concludes that the P (Beza) is engaged in A (Learning).

```

# =====
# Sample_CAA:User_Location_Rule
# =====
[UserLocationRule1:
(?CAA:D CAA:holdBy ?CAA:p)
(?CAA:D CAA:Connected ?CAA:d)
(?CAA:D CAA:foundIn ?CAA:Al)
(?CAA:Al CAA:subsetOf ?CAA:L)
→(?CAA:p CAA:foundIn ?CAA:L) ]
# =====
# Sample_CAA:User_Activity_Rule
# =====
[UserActivityRule1:
(?CAA:p CAA:owns ?CAA:D)
(?CAA:D CAA:Connected ?CAA:d)
(?CAA:D CAA:foundIn ?CAA:Al)
(?CAA:d CAA:foundIn ?CAA:Al)
(?CAA:Al CAA:subsetOf ?CAA:L)
(?CAA:Al CAA:DefaultActivity ?CAA:A)
→(?CAA:p CAA:EngagedIn ?CAA:A) ]

```

Fig. 6. Sample CADB Rules for user Location and Activity

On the other hand¹¹, propose adaptive context-aware pervasive system architecture, which emphasize entities (such as user, devices, application, or group) as self-managed composites. The architecture proposed by this study also assumes the interaction among entities. Each composite maintains a representation of its relationships with other entities and any context conditions that affect those interactions.

The above researches contribute towards the context-aware application representation and modeling but have some limitations on how context reasoning works end to end from low level context extraction to high level context data representation through different stages of context analysis.

5. Conclusion and Feature work

Context awareness in CAA is aided by distinct functions at different layers in the CAA Architecture. At every layer of the proposed architecture a context information is build up on existing user experience (from user profile) and utilized by other higher components as discussed above.

The architecture proposed in this study is already proven to support daily life interaction, for context aware diary builder¹². We strongly believe CAA is a complete architecture that can fit to any context aware application with a minimal modification of the assumptions and rules proposed in this work.

Context aware applications in a pervasive world require understanding user context in order to provide the necessary services. On the other hand, the more user context is acquired by a system the more user concerns about privacy and security becomes an issue. Thus, as a future work, one has to consider issues such as privacy, security, and performance analysis of the suggested architecture.

6. Reference

1. Patrícia Dockhorn Costa. Architectural Support for Context-Aware Applications: From Context Models to Services Platforms. CTIT Ph.D.-Thesis Series, No. 07-108 ISSN 1381-3617; No. 07-108.
2. Harry Chen. An Intelligent Broker Architecture for Pervasive Context-Aware Systems. PhD Thesis, University of Maryland, Baltimore County, December 2004.
3. Kasim Rehman, Frank Stajano, and George Coulouris. Architecture for Interactive Context-Aware Applications. IEEE Computer Society, Vol. 6, No. 1 January–March 2007.
4. Xu Huang, Mark Barralet, and Dharmendra Sharma. Accuracy of Location Identification with Antenna Polarization on RSSI. Preceding of the International MultiConference of Engineers and Computer Scientists 2009 Vol I, MESCE 2009, Hong knong, March 18-20, 2009.
5. Adenauer Corrêa Yamin, Jorge Luis Victória Barbosa, and Iara Augustin, Cláudio Fernando Resin Geyer. SAM: a Software Architecture for Pervasive Computing”, CLEI Electronic Journal, Volume 8, Number 1, Paper 3, August 2005
6. Moeiz Miraoui, Chakib Tadjl, and Chokriben Amar. Architectural Survey of Context-Aware Systems in Pervasive Computing Environment. Ubiquitous Computing and Communication Journal (UBICC), Vol. 3, No. 3, 2008.
7. Mamo Beza, and Ejigu Djigu. Agent Based Context Acquisition for Context Aware Pervasive Systems. 6th International Multi Conference on Intelligent Systems, Sustainable, New and Renewable Energy Technology and Nanotechnology (IISN-2012), March 16–18, 2012. Institute of Science and Technology Klawad-133105, Haryana, India.
8. Etter, R., Costa, P.D., and Broens, T. A Rule-Based Approach Towards Context-Aware User Notification Services. In: Proceedings of the IEEE International Conference on Pervasive Services 2006, , Lyon, France. pp. 281-284. IEEE Computer Society. ISBN 1424402379, Jun 26-29, 2006.
9. Krummenacher R. and Strang T. Ontology-Based Context Modeling. In Workshop on Context-Aware Proactive Systems Issue: 2007.
10. Ejigu D., Marian Scuturici, and Lionel Brunie, Hybrid Approach to Collaborative Context-Aware Service Platform for Pervasive Computing”, Journal of Computers, Vol. 3, No. 1, January 2008.
11. Alan Colman, Minh Tran, and Jun Han. An Adaptive Architecture for Context-Aware Interaction in Pervasive Applications. Centre for Information Technology Research (CITR); Swinburne University of Technology; Melbourne, Victoria, Australia
12. Mamo Beza. Context Aware Diary Builder. MSC thesis submitted to Addis Ababa University, Addis Ababa, Ethiopia, Jan 2010.