IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 15, Issue 3 (Nov. - Dec. 2013), PP 26-29
www.iosrjournals.org

A survey on context aware system & intelligent Middleware's

Sudeep K. Hase[1], Prof. M. B. Vaidya[2]
[1,2](Department of Computer Engineering, AVCOE, Ahmednagar, India)

Abstract: *Context aware system or Sentient system is the most profound concept in the ubiquitous computing. In the cloud system or in distributed computing building a context aware system is difficult task and programmer should use more generic programming framework. On the basis of layered conceptual design, we introduce Context aware systems with Context aware middleware's. On the basis of presented system we will analyze different approaches of context aware computing. There are many components in the distributed system and these components should interact with each other because it is the need of many applications. Plenty Context middleware's have been made but they are giving partial solutions. In this paper we are giving analysis of different middleware's and comprehensive application of it in context caching.*

Keywords: *Context aware system, Context aware Middleware's, Context Cache*

I. INTRODUCTION

The Context aware or Sentient systems are comes under pervasive computing. Weiser (1991) introduced pervasive which refers to the integration of mobile devices such as notebook, PDAs, smartphones etc into the humans everyday life. It is expected that the services should react specifically to their time, places and current locations. Context can be refer as location, identities of nearby people and object, and changes to those objects. The situation of entity can be characterized by context. Users profile, device status, networking information or sensors are the ways to retrieve context information. The first context aware application is developed by Want et. al. (1992) which is based on Active badge Location System. According to the users current location, location-aware tour guide applications developed in middle 1990s. (Abowd et. al., 1997; Sumi et. al., 1998; Cheverest et. al., 2000)

Person's emotional state, focus of attention, place, time and date, people in the user's environment can be defined as the sentient defined by Dey (1998). Hull et al. (1997) describe context as the aspects of the current situation. Brown defines context to be the elements of the user's environment which the computer knows about. Most context aware systems make use of external sentient factors as they provide useful data, such as place information. Furthermore, external attributes are easy to sense by using off-the-shelf sensing technologies. *Watson Project* used logical data (Budzik and Hammond, 2000) and the *IntelliZap Project* (Finkelstein et al., 2001) which support the user by giving useful information due to information read out of web pages, records etc. When dealing with sentient, three entities can be differentiated (Dey and Abowd, 2001): *places* (rooms, buildings etc.), *people* (individuals or groups) and *things* (physical objects, Mobile Component, computer components etc.). These entities may be given by various attributes which can be classified into four categories: *identity* (each entity has a unique identifier), *location* (an entity's position, co-location etc.), *status* (or activity, meaning the properties of an entity, e.g., heat and lightning for a room, processes running currently on a mobile etc.) and *time* (used for timestamps to define events etc.).

II. APPROACHES

The implementation of context-aware system can be done in many ways. The result and system implementation changes when the sensors are at local or remote places. The system users or number of people can change the approach to implement context aware system. These different implementation approaches of the context aware system is given by Matthias Baldauf, Dustdar, S. & Rosenberg [17] are as follows :
1) Sensor Fetching
2) Middleware Implementation
3) Context Server
4) Networked Services
In this section we can briefly explain the concept of all approaches

2.1 Sensor Fetching-

This approach is not used into the distributed system because devices with sensors are locally built in. The application which resides on to the users device can fetch information directly through the sensors. The sensor drivers are hardwired into the application. It will fetch data directly from the sensors so it can be a hard task for the concurrent servers.

2.2 Middleware Implementation-

This approach is useful because it introduces the layered architecture to context aware system. The technique called Encapsulation is applicable in this case because it can hide the implementation details which can be useful for the business purpose. It can be used in caching the contextual data.

2.3 Context Server-

In this approach multiple clients can access the remote data source. So the concept of middleware approach not only act as a broker but also act as a federated broker architecture. Gathering of data from the sensors is now moved to this so called context server. The reuse of these sensors can be neglected by using context servers. This can leads to the increase in power consumption and disk space of devices rather than attached to the sensors.

2.3 Networked Services-

This approach is not so useful as the widget architecture because of complex network based components. Without using the global widget manager this approach using discovery technique.

III. ARCHITECTURE

3.1 Context Broker Architecture-

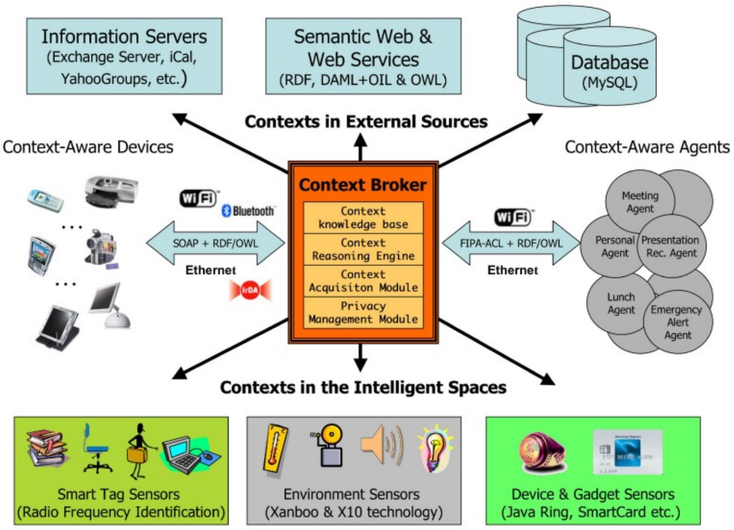


Fig.1 Context Broker Architecture

Context Broker Architecture (*CoBrA*) (Chen et al., 2003) [17], [18] is used to support context aware computing which is an agent based architecture called physical spaces. Physical spaces (e.g., Halls, Bikes, offices and meeting rooms) that are grouped with intelligent systems that provide pervasive computing services to system users. With *CoBrA* there is the presence of an CxB (Context Broker) that maintains and manages a shared contextual framework on the behalf of the agents. These agent can be the applications held by mobile devices , services that are provided by devices in a Hall (e.g., Hall temperature controller) and web services that gives a web presence for individuals, places and things in the physical world (e.g., People tracker service). [10][11] The CxB consists of four functional main components: Context KB, Context Inference Engine, Context Acquisition Module and the Privacy Management Module[14]. *CoBrA* offers the possibility of creating broker federations called as broker synchronization.

3.2 The SOCAM Architecture-

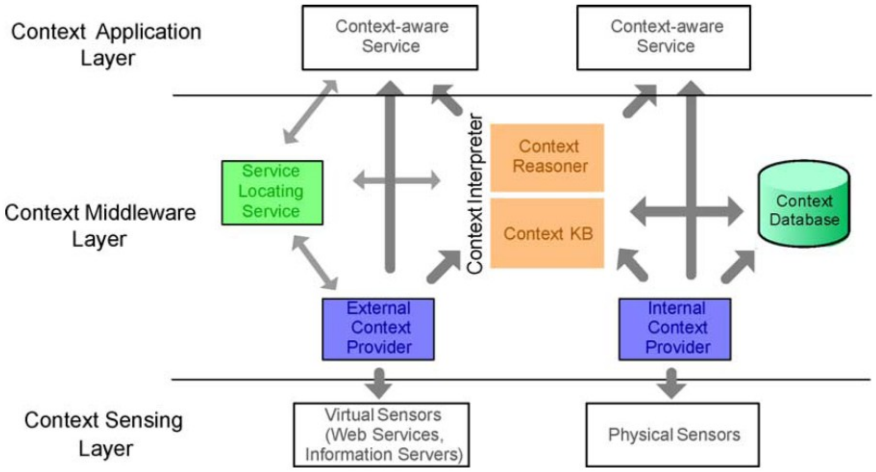
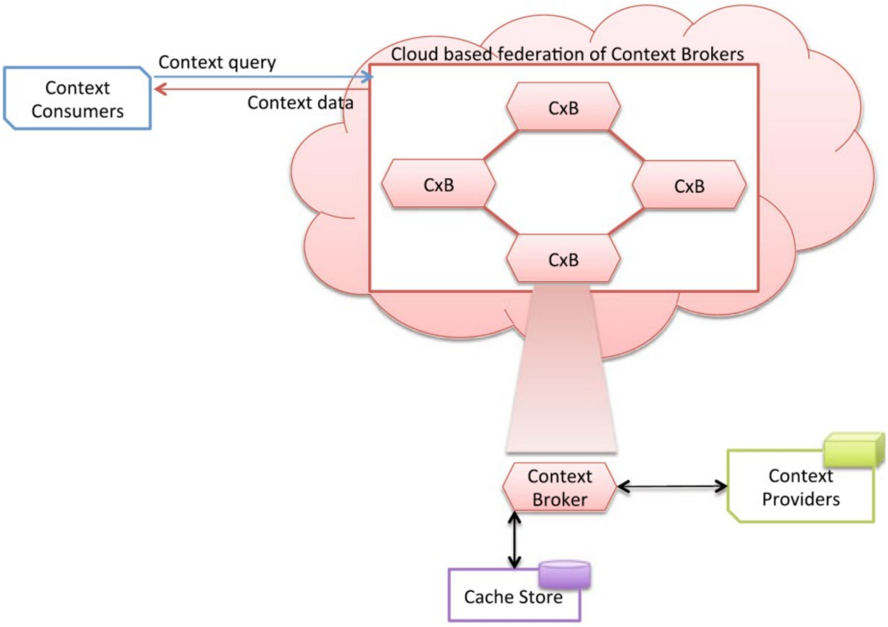


Fig.2 The SOCAM Architecture

In Pervasive computing environment the SOCAM (Service Oriented Context Aware Middleware) [19] architecture provides efficient way to implement context aware system. It consists of the following components which act as independent service components. CxP, Context interpreter, Context database, Context-aware services, locating service[7][11]

IV. APPLICATION – CACHE MANAGEMENT

Cache of the context data is most important part of context aware system. Minimum response time of any query is the primary aim. This cache management application is completed with three main stages. 1. Context Consumer 2. Federated Context Broker 3. Context Consumer. Context consumer send the query by using web application to the federated broker architecture. At broker side cache is maintained. If the answer of that query is available in the cache of the broker then answer directly given to the context consumer. If answer is not presented in the broker's cache then it will ask to the provider. Then provider provides with the answer and it will stored into the brokers cache. For that purpose Saad Kiani (2012) [15][16] implemented three algorithms. A) Remove oldest first B) Remove least used first C) Remove soonest expiring first and after that on the basis of scope and query evaluation time he has implement bipartite context cache algorithm. This bipartite context cache is implemented on the basis of short and long validity. The result can be increased if we have implement the federated context broker architecture in context cache management. These brokers can be federated into the cloud mechanism.



V. CONCLUSION

This survey paper on the context aware system gives the design principles and architecture of context aware system and the intelligent middleware's. Context aware system can be used in day to day life for better performance. Standardized frameworks and protocols can be used to enhance the performance of the context aware system. Different middleware approaches can give the different solution in context aware system. Ubiquitous computing plays vital role into the context aware system. The context management systems are the middleware's between context aware sources and application.

REFERENCES

[1]. Abowd, Atkeson, Long, Hong, Kooper and Pinkerton, M. (1997), Cyber guide - a mobile context aware tour guide, Wireless Networks, Vol-3, No-5, pp. 421-433

[2]. Biegel and Cahill (2004) Framework for developing mobile, context aware applications', Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communication, pp. 361–365.

[3]. [3] G. D. Abowd. Software engineering issues for ubiquitous computing. Proceedings of the 21st international conference on Software engineering, pp 75–84, IEEE Computer Society Press, 1999.

[4]. C. Bisdikian, J. Christensen, J. Davis, M. R. Ebling, G. Hunt, W. Jerome, H. Lei, Enabling location-based applications. Proceedings of the 1st international workshop on Mobile commerce, pp 38–42, ACM Press, 2001.

[5]. L. Capra, W. Emmerich and C. Mascolo. CARISMA – Context aware reflective middleware System for Mobile Applications, IEEE Transactions on Software Engineering, 29, pp 921–945, Oct. 2003.

[6]. A. Harter, A. Hopper, P. Steggles, A. Ward. The anatomy of a context-aware application. Wireless Networks :187–197,

[7]. 2002. Gu T, Pung HK, Zhang DQ (2005) A Service-oriented middleware for building context-aware Services. J Net Comp Appl 28: 1–18

[8]. Weiser M (1991) The computer for the twenty-first century. SciAm 265; 94–104

[9]. Bardram JE (2005) The Java Context Awareness Framework – a service infrastructure and programming framework for context-aware applications. In Pervasive Computing, Vol - 3468, LNCS 98–115. Springer

[10]. Henricksen K, Indulska J, McFadden T, Balasubramaniam S (2005) Middleware for distributed context-aware systems. In On the Move to Meaningful Internet Systems 2005: CoopIS, DOA , and ODBASE, Volume 3760 of Lecture Notes in Computer Science, ed. Meersman R, 846– 863.Heidelberg: Springer- http://dx.doi.org/10.1007/11575771 53

[11]. Floreen P, Przybilski M, Nurmi P, Koolwaaij J, Tarlano A, Wagner M, Luther M, Bataille F, Boussard M, Mrohs B, et al (2005). Towards a context management framework for mobiLife. 14th IST Mobile & Wireless Summit

[12]. Buchholz T, K upper A, Schiffers M (2003) Quality of Context - What It Is and Why We Need It. Workshop of the HP Open View University Association

[13]. Kernchen R, Bonnefoy D, Battestini A, Mrohs B, Klemettinen M (2006) Context-awareness in mobiLife. Proceedings of the 15th IST Mobile Summit, Greece

[14]. Bellavista P, Corradi A, Stefanelli C (2006) A mobile computing middleware for location and context-aware internet data services. ACM Trans Internet Technol (TOIT) 6: 380

[15]. Kiani SL, Knappmeyer M, Reetz E, Baker N (2010) Effect of Caching in a Broker based Context Provisioning System. In Proceedings of The 5th European Conf. on Smart Sensing and Context, Vol- 6446, LNCS 108–121

[16]. Kiani et al.(2012) Context caches in the cloud. Journal of cloud computing : Advances, systems and applications 2012, 1:7

[17]. Baldauf, M., Dustar, S. and Rosenberg, F.(2007) 'A survey on context-aware systems', Int. J. Ad Hoc and ubiquitous computing, Vol.2,No.4, pp. 263-277

[18]. Harry Chen, Tim finin, Anupam Joshi, 'An Intelligent Broker for Context-Aware Systems'.

[19]. Tao Gu, Hung Keng Pung, Da Qing Zhang, A service-oriented middleware for building context-aware service, Journal of network and computer application 28(2005) 1-18