ARCHITECTURAL ANALYSIS OF CONTEXT-AWARE SYSTEMS IN PERVASIVE COMPUTING ENVIRONMENT

Divya Udayan J*, HyungSeok Kim**

Abstract Context aware systems are those systems that are aware about the environment and perform productive functions automatically by reducing human computer interactions(HCI). In this paper, we present common architecture principles of context-aware systems to explain the important aspects of context aware systems. Our study focuses on identifying common concepts in pervasive computing approaches, which allows us to devise common architecture principles that may be shared by many systems. The principles consists of context sensing, context modeling, context reasoning, context processing, communication modelling and resource discovery. Such an architecture style can support high degree of reusability among systems and allows for design flexibility, extensibility and adaptability among components that are independent of each other. We also propose a new architecture based on broker-centric middleware and using ontology reasoning mechanism together with an effective behavior based context agent that would be suitable for the design of context-aware architectures in future systems. We have evaluated the proposed architecture based on the design principles and have done an analyses on the different elements in context aware computing based on the presented system.

Keywords : context-awareness, context architectures, broker-centric middleware, ontology reasoning, behavior based context agent

*Department of Internet & Multimedia Engineering, Konkuk University, e-mail: divyadon.udayan@gmail.com
**Department of Internet & Multimedia Engineering, Konkuk University, e-mail: hyuskim@konkuk.ac.kr
■ 접수일 : 2011년 6월 23일 / 심사일 : 2011년 7월 15일 / 게재확정일 : 2013년 4월 23일

1. Introduction

Context-awareness in pervasive computing aims to provide automatically adapted services for end-users and applications according to a global context. This paper aims to make a survey of relevant architectures [1,2] beginning with localization-aware systems up to present context aware systems. This survey presents a comparison and evaluation of architectures considering design principles as parameters for evaluation which are considered important for pervasive computing such as : sensing, context model, context reasoning, context processing, communication model, and resource discovery. Our objectives are to provide a guide for the developers and architecture designers of context-aware systems in a pervasive computing environment. Also, this survey aims towards the proposal of a new architecture that will best suit the context-aware systems in pervasive computing environments.

The rest of the paper is organized as follows, in Section 2 we review some previous surveys done until now on context-aware architectures. In Section 3 we present the evaluation and comparison criteria based on the design principles. In Section 4 we present the new architecture proposal for context aware systems. Finally, Section 5 draws some concluding remarks.

2. Context-aware architectures

Context aware systems can be implemented depending on various aspects like the location of sensors, the amount of users, the availability of resources and the distribution of components. Also the process of context data acquisition and processing is very important in designing a context aware system[3,4]. The common design principles present in the surveyed architectures are related with location of sensors, context model, context reasoning, context processing and resource discovery. Based on these design principles we classify the context aware architectures as localization-aware systems, agent based systems, single server based systems and object based systems. The detailed description is presented in the following section.

2.1 Localization-aware systems

Localization aware systems helps to locate persons or materials relative to known positions. Context aware systems dealing with location awareness is of greater importance today due to the increase in mobile applications. Some architectures designed based on localization aware systems are discussed in this section. The Active Badge[5] aims to build a system for phone calls delivery according to the called person's localization. The system(Fig 1.) uses badges which continuously emit infra-red signals and each badge contains the carrier identification. The signals emitted by these badges are sensed by some receivers distributed in the office. The perceived signals are then sent to a server. The latter presents to a receptionist the information about badge carriers and their location. This information helps the receptionist to deliver a call to the phone closest to the called person. Another architecture stick-e-notes[6] consists of a personal digital assistant (PDA) connected to a localization sensor (GPS or Active Badge). The PDA communicates with one another depending on the application.

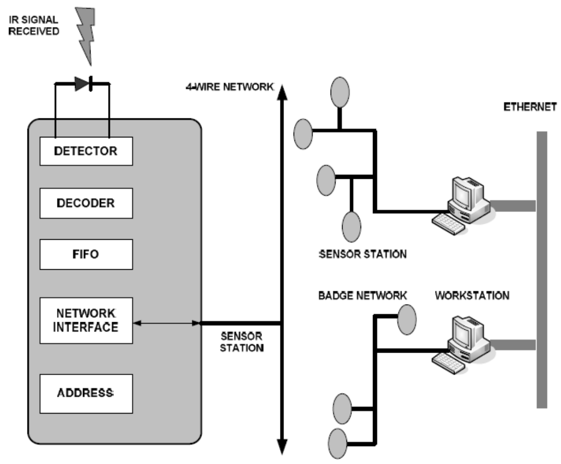

Fig 1. Architecture of Active Badge

2.2 Single-server systems

In this type of architecture, the discovery mechanism is based on single server client design. Context toolkit and JCAF are examples of this type of architectures. The context toolkit [7,8,9] has a layered architecture that permits the separation of context acquisition, representation and adaptation process. It is based on context widgets which operate similarly to GUI widgets. These widgets offer a good abstraction of context and provide reusable blocks for context sensing. This architecture offers a distributed communication among system devices and reusable widgets but the discovery mechanism is centralized which does not make it a perfect P2P communication model. JCAF (java context-awareness framework)[10] is based on java programming language to support the development of context-aware applications. The

JCAF architecture(Fig 2.) consists of a set of components called "context service" which communicates in a P2P mode. These components

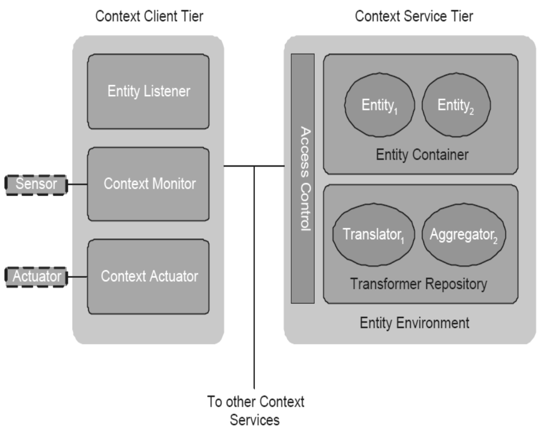

Fig 2. Architecture of JCAF

are responsible for gathering context information in a specific environment. The JCAF also controls the contextual information (trust on the information sensed by a particular sensor, error probability of information perceived by a sensor, etc.). The remote communication between the architecture components is done using java RMI (remote method invocation). The context service does not have an automatic discovery mechanism but can use a configuration file containing all others active context services.

2.3 Agent based systems

Agent based architectures provide a software agent to control communication with applications. The ParcTab[11,12], uses infra-red communication with a transmitter in a room of an office which communicates with a LAN via an RS-232 connection. For each ParcTab there is a corresponding software agent that controls its communication with applications.

Another architecture SOCAM [13,14] is an architecture of a service oriented context-aware middleware. The architecture (Fig 3.) uses the client/server model where the context interpreter collects contextual information from context providers (internal or external) and context database and provides them to the context-aware mobile services and the service locating service. The main strength of the SOCAM architecture is its context reasoner which uses ontology for context description and allows a robust reasoning on context.

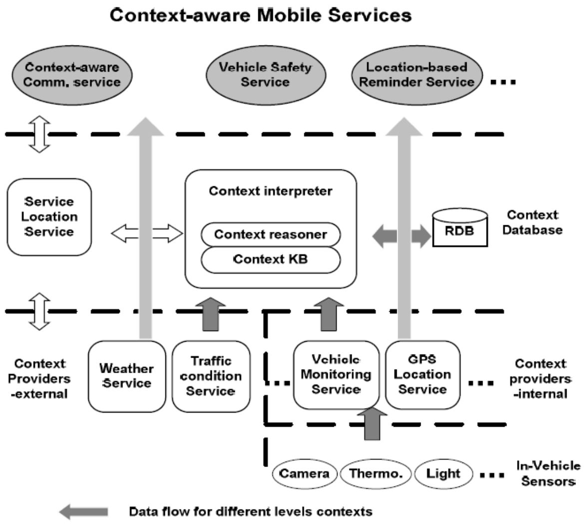

Fig 3. Architecture of SOCAM

2.4 Object based systems

Object oriented architectures can be considered as an extension of capability architecture systems and have the same ability to provide a basis for protection and security. They work based on a object oriented framework. Hydrogen [15,16] is an object oriented architecture. Hydrogen(Fig 4.) is a three layered architecture that responds to particular requirements of mobile devices. The architecture has the following layers: adaptation, management and application. The Hydrogen approach considers context as any pertinent information on an application environment and describes it using an object oriented model. It takes into account the limited resources of mobile devices such as battery, memory, processing, etc. and uses a P2P communication model.

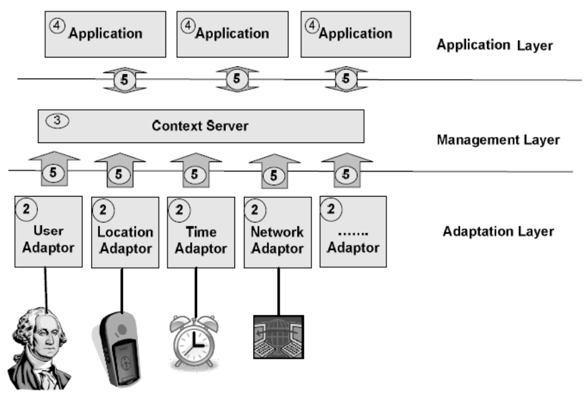

Fig 4. Architecture of Hydrogen

CORTEX[17] is an architecture based on the "sentient object" which has some characteristics like sensitivity, autonomy and proactivity. The sentient object contains two interfaces: Sensor of events perceived by sensors and Event emission to adapt to the current context (actuator or producer).

3. Design Principle Analysis

In this section, we analyse the basic design principles and compare various architectures discussed in the previous section in terms of their design principles. We have considered the first design principle as sensing. The choice of sensors has to be appropriate since the details and shortcomings of the sensors may be carried up to the application level and affect the flexibility and extensibility of the application. Based on the classification of context aware architectures we have analysed that localization aware systems are IR based systems, the single server systems depend on widgets or sensor nodes, agent based systems use blackboard based sensing and object oriented systems use context providers for sensing. Secondly, an efficient model for handling, sharing and storing context data is essential for the working of a context aware system. The context model used by the surveyed systems are either agent based or object based depending on their design. The most relevant context modeling approaches are: Key-Value models, Markup scheme models, Graphical models, Object oriented models, and Logic based models[18]. Information from physical sensors without further interpretation, can be meaningless because context encompasses more than just the user's location. The context may also be environment context or place/time/person/object context. Solving this problem is by deriving information from raw sensor values which is called context reasoning or interpretation. We have analysed that only agent based and object based systems provide some amount of reasoning capability and it is totally lacking in other architectures. Considering the next design principle as context processing which is the processing of raw context data perceived by the sensors, as the end-users are interested in already interpreted and aggregated information rather than raw data. Context aware middleware support a variety of context processing components that manage the flow of context information between the sensors/actuators and applications, since the middleware must address many requirements such as heterogenity, mobility, scalability and tolerance for component failures and disconnections. In addition, it should protect the user's location and preferences together with the privacy policy. Context processing can also occur in the service level. Various architectures use different approaches for context processing as summarized in Table 1.

Table 1. Comparison of Surveyed Architectures

Architecture	Sensing	Context Model	Context Reasoning	Context Processing	Communication Model	Resource Discovery
Active Badge	IR based	-	NO	N.A	C/S	N.A
ParcTab	IR based	Context Agent	NO	N.A	C/S	N.A
Stick-e-notes	IR based	-	NO	N.A	P2P	N.A
Cyber guide	Sensor nodes	-	NO	N.A	HYBRID	N.A
Context toolkit	Widget	Widget	NO	Attribute Value tuples	HYBRID	Context Interpretation & aggregation
JCAF	Sensor nodes	Context Service	NO	N.A	HYBRID	N.A
CASS	Sensor nodes	Object	YES	Relational data model	C/S	Inference Engine & KB
SOCAM	Context providers	Context agent	YES	Ontologies (OWL)	C/S	Context reasoning engine
Hydrogen	Context providers	Object	NO	Object-oriented	P2P	Interpretation & aggregation of raw data only
CMF	Blackboard based	Context Agent	YES	Ontologies (RDF)	C/S	Context recognition service
CORTEX	Context providers	Sentient object	YES	Relational Data model	P2P	Service discovery framework
New Proposal	Application dependent	Context Broker	YES	Ontology (RDF)	P2P	Interpretation & aggregation based on behavior planning

Next design principle is communication model. With the pervasive deployment of computers, P2P is increasingly receiving attention in research, product development, and investment circles. There is no clear border between a client-server and a P2P model. Both models can be built on a spectrum of levels of characteristics (e.g., manageability, configurability), functionality, organizations (e.g., hierarchy versus mesh), components (e.g., DNS), and protocols. Furthermore, one model can be built on top of the other or parts of the components can be realized in one or the other model. Finally, both models can execute on different types of platforms (Internet, intranet, etc.) and both can serve as an underlying base for traditional and new applications. Also a combination of client-server and P2P model is also possible like in the architectures Cyber-guide, context toolkit, JCAF. Finally, a discovery mechanism to search for and find appropriate sensors at runtime is essential. The discoverer works as registry component which interpreters, aggregators and widgets have to notify about their presence and their contact possibilities. After registration the components are pinged to ensure that they are operating. If a component does not respond to a specified number of consecutive pings, the discoverer determines that the component is unavailable and removes it from its registry list. We analysed that discovery mechanism is completely lacking in localization-aware systems whereas agent-based and object based systems provide more resource discovery capability. A detailed comparison of the surveyed architectures is given in the Table 1.

4. Proposed Architecture

The comparative study of the surveyed architectures shows that some architectures uses local-built in sensors and does not connect to distributed sensors, therefore, no discovery mechanism is involved. Even if discovery mechanism is involved, only few architectures can infer the behavior of the system depending on the context. Thus there arise a need for a new architecture which can support pervasive context-aware systems and use the reasoning mechanism to successfully predict the behavior of the system. The proposed architecture(Fig 5.) is based on Semantic Web languages for sharing context of different services and publishing them for providing reasoning based on context ontology. There is a broker centric middleware [19] that shares the model of the context with other elements in the working space and also maintains the security policies that is very much important in context aware applications.

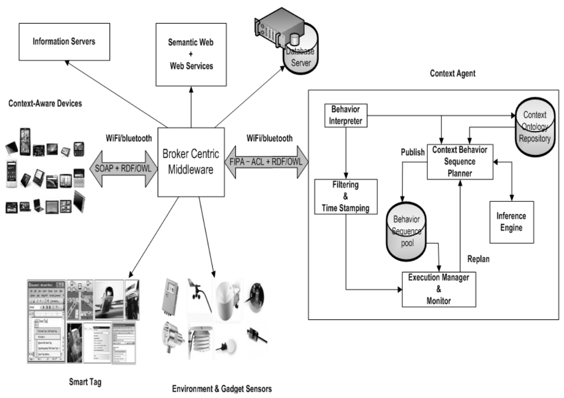

Fig 5. Block diagram of the proposed architecture

The proposed architecture differs from the previous systems(Table 1) in the following ways: This architecture provides a resource-rich agent called the context broker to manage and maintain a shared model of context. The context brokers can infer context knowledge that cannot be easily acquired from the physical sensors. In the previous systems, individual entities are required to manage and maintain their own context knowledge.

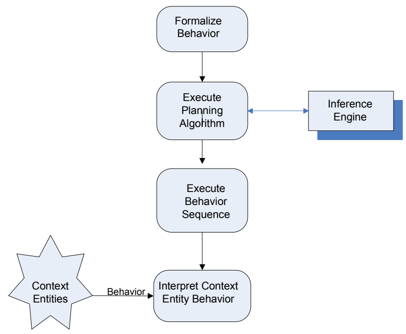

Fig 6. Process of Behavior Planning in Context Agent

Agent takes charge of carrying out the composition by a planning algorithm(Fig 6.) based on context entity behaviors, service behaviors, current context environment, planning requirements and evaluation mechanisms. Recommended contextual behavior sequence can be calculated and relevant services start to execute according to the sequence under the monitoring of the Context Agent. The evaluation will be updated with respect to the detected impacts of contexts during interaction between services and context entities.

5. Conclusion

Context-awareness is an important aspect of the ubiquitous computing vision. Through an understanding of context, computing systems will reduce the human computer interactions(HCI) at every step during their interaction with the systems. This survey aims to study different context-aware architectures to support and ease the development of such system. For each architecture, the strengths and weaknesses are analysed. Context aware systems should be proactive systems that embed a reasoning mechanism to easily adapt to the specific task but it is analysed that it is not present in all the architectures that were considered for study. This architecture differs from other similar systems in using ontologies for context representation and modeling, rule-based logical inference for context reasoning, and declarative policies for privacy protection. The behavior based planning mechanism of the context agent also distinguishes this from the previous existing architectures.

As future work, we would like to evaluate application specific context aware systems in terms of the design principles suggested in our paper and also based on historical context data and security policies.

ACKNOWLEDGEMENT

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No. 2012M3C4A7032182)

References

[1] Baldauf, M., Dustdar, S. and Rosenberg, F. A Survey On Context-Aware Systems. International Journal of Ad Hoc and Ubiquitous Computing. Vol 2. No. 4. pp. 263-277. 2007.

[2] Winograd, T. Architectures for Context. Human-Computer Interaction Journal, Vol 16. No. 2. pp. 401-419. Special Issue on Context Aware Computing. 2001.

[3] Oh, Y., Hang J. and Woo W. A Context Management Architecture for Large-scale Smart Environments. IEEE Communications Magazine. Vol. 48. No.3. pp.118-126. 2010.

[4] Schmidt, A., Adoo, K. A., Takaluoma, A., Tuomela, U., Van Laerhoven, K., Van de Velde, W. Advanced Interaction in Context. Proc. 1st Int'l Symp. Handheld Ubiquitous Comp. 1999.

[5] Want, R., Hopper, A., Falcao, V. and Gibbons, J. The Active Badge Location System, ACM Transactions on Information Systems. Vol 10. No 1. pp. 42-47. 1992.

[6] Pascoe, J. The Stick-e Note Architecture: Extending the Interface Beyond the User. In Proceedings of International Conference on Intelligent User Interfaces. pp. 261-264. 1997.

[7] Kim, H., Cho, Y. and Oh, S. CAMUS: A Middleware Supporting Context-aware Services for Network-based Robots. IEEE Workshop on Advanced Robotics and Social Impacts. 2005.

[8] Singh, A. and Conway, M. Survey of Context aware Frameworks - Analysis and Criticism. UNC-Chapel Hill ITS. Version 1. 2006.

[9] Henricksen, K., Indulska, J., McFadden, T. and Balasubramaniam, S. Middleware for Distributed Context-Aware Systems. OTM Conferences. pp. 846-863. 2005.

[10] Bardram, J. E. The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications. Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive 2005). Lecture Notes in Computer Science. Volume 3468. pp. 98-115. 2005.

[11] Want R., Schilit, B. N., Adams, N. I., Gold, R., Petersen, K., Goldberg, D., Ellis, J. R. and Weiser, M. An overview of the ParcTab ubiquitous computing experiment. IEEE Personal Communications. Vol. 2. No. 6. pp. 28-43, 1995.

[12] Biegel, G. and Cahill, V. A framework for developing mobile, context-aware applications. Proc. of the 2nd IEEE Conference on Pervasive Computing and Communication. pp.361- 365. 2004.

[13] Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H. and E-J. Malm, Managing context information in mobile devices. IEEE Pervasive Computing. Vol. 2. No. 3. pp. 42-51. 2003.

[14] Gu, T., Wang, X. H., Pung, H. K. and Zhang, D. Q. A Middleware for Context-Aware Mobile Services. IEEE Vehicular Technology Conference. 2004.

[15] Kjær, K. E. A Survey of Context-Aware Middleware. Proceedings of the IASTED software engineering conference. 2007.

[16] Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G. and Altmann, J. Context-awareness on mobile devices - the hydrogen approach. Proceedings of the 36th Annual Hawaii International Conference on System Sciences. pp. 292-302. 2002.

[17] Verissimo, P., Cahill, V. et al. CORTEX: Towards Supporting Autonomous and Cooperating Sentient Entities. In Proc. of European Wireless. 2002.

[18] Strang, T. and Linnhoff-Popien, C. A context modelling survey. In Workshop on Advanced context modelling, reasoning and management. UBICOMP. 2004.

[19] Chen, H., Finin, T and Joshi, A. An Intelligent Broker for Context-Aware Systems. Adjunct Proc. of Ubicomp. pp. 183-184. 2003.

Divya Udayan J

Divya Udayan J received her BTech degree in Information Technology at Kerala University in 2005 and MTech degree in Embedded Systems from Amrita University, India in 2010. She currently is a PhD student in Internet and Multimedia Eng. at VRLab, Konkuk University, Korea. Her research interests include Augmented reality, Ubiquitous computing and HCI.

HyungSeok Kim

HyungSeok Kim is an Associate Professor at the Department of Internet & Multimedia Eng., Konkuk University, Korea from the year 2006. Before joining Konkuk University, he was a senior researcher at MIRALab, University of Geneva, being involved in research activities on virtual reality. He received his PhD in Computer Science in February 2003 at VRLab, KAIST. His current research activities are focused on topics of shaped modeling for real-time rendering and evoking believable experiences in the virtual environment.