Towards the Future Internet
G. Tselentis et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-60750-007-0-335

Context-Aware Systems and Implications for Future Internet

Nigel Baker[1], Madiha Zafar[1], Boris Moltchanov[2], Michael Knappmeyer[1,3]

[1]University of the West of England, UK
[2]Telecom Italia Lab, Italy
[3]University of Applied Sciences Osnabrück, Germany

{nigel.baker, madiha.zafar, michael.knappmeyer}@uwe.ac.uk,
boris.moltchanov@telecomitalia.it

Abstract. The ubiquity of mobile devices and proliferation of wireless networks will allow everyone permanent access to the Internet at all times and all places. The increased computational power of these devices has the potential to empower people to generate their own applications for innovative social and cognitive activities in any situation and anywhere. This wireless connection is not limited to user devices, almost any artefact from clothing to buildings can be connected and collaborate. Furthermore new sensor technologies and wireless sensor networks provides environmental intelligence and the capability to sense, reason and actuate. This leads to the exciting vision of the interconnection of artefacts embedded in our real environment, forming a society of “intelligent things” and “smart spaces”. This paper discusses the main concepts and role that context-awareness and context aware systems will play in this vision and the significance for future networks and future Internet.

Keywords: context, context-awareness, sensor networks, future internet, sense, reason, actuate

1 Introduction

In the real world being aware of context and communicating context is a key part of human interaction. Context is a much richer and more powerful concept particularly for mobile users and can make network services more personalised and useful. Location and presence are examples of context based services widely deployed today. Harvesting of context to reason and learn about user behaviour will enhance the “internet of services” or “cloud computing” vision allowing services to be composed and customised according to user context. The concept of awareness and context aware applications and systems is a much more difficult proposition. Context awareness refers to the capability of an application, service or even an artefact being aware of its physical environment or situation and responding proactively and intelligently based on such awareness. Context-aware applications, context-aware artefacts or context aware systems are aware of their environment and circumstances and can respond intelligently. The ubiquity of mobile devices and proliferation of wireless networks will allow everyone permanent access to the Internet at all times and all places. The increased computational power of these devices has the potential to empower people to generate their own applications for innovative social and cognitive activities in any situation and anywhere. This wireless connection is not limited to user devices, almost any artefact from clothing to buildings can be connected and collaborate. Furthermore new sensor technologies and wireless sensor networks provides environmental intelligence and the capability to sense, reason and actuate. This leads to the exciting vision of the interconnection of artefacts embedded in our real environment, forming a society of "intelligent things" and "smart spaces". This will enable all sorts of innovative interactive pervasive applications. The key denominator in all these applications and systems is that awareness manifests itself from the self property of being able to sense, reason and actuate. A future internet capable of embracing this concept and delivering context aware services to users and artefacts elevates this to a pervasive sensing and acting knowledge network. This would be a network able to make decisions, actuate environmental objects and assist users.

2 Context and Situations

It is a challenging task to define the word 'context' and many researchers grapple with the task of creating definitions. Ryan et al. [1] referred to context as the user's location, environment, identity and time. Dey [2] defines context as the user's emotional state, focus of attention, location and orientation, date and time, as well as objects and people in the user's environment. Another common way of defining context was the use of synonyms. Hull et al. [3] describe context as the aspects of the current situation. These kinds of definitions are often too broad. Brown [4] defines context to be the elements of the user's environment which the computer knows about. Perhaps the most often used definition is given by Dey and Abowd [5]. These authors refer to context as "any information that can be used to characterize the situation of entities (i.e., a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves." Another common way to classify context instances is to distinguish different context dimensions. Prekop and Burnett [6] and Gustavsen [7] refer to these dimensions as external and internal, and Hofer et al. [8] refer to it as physical and logical context. External (physical) dimension refers to context that can be measured by hardware sensors, for example location, light, sound, movement, touch, temperature or air pressure, whereas the internal (logical) dimension is mostly specified by the user or captured by monitoring user interactions, for example the user's goals, tasks, work context, business processes, the user's emotional state or social relationships.

The notion of situation is closely related to the concept of context. Zimmermann [9] defines it as "the state of a context at a certain point (or region) in space at a certain point (or interval) in time, identified by a name". Therefore it can be considered as a structured representation of a part of the context with direct comparison to a snapshot taken by a camera. Consequently we can consider that situation can be derived from aggregating and refining types of context information. In other words, as remarked by Loke [10], situation can be viewed as being at a higher level of abstraction than context. Taken to the extreme, situation can be considered as "the complete state of the universe at an instant of time". Therefore a situation may comprise an infinite variety of contextual information. Computational and Artificial Intelligence aspects of situation have been widely explored. Henricksen presents a logical and arithmetical model based on object relations [11], Barwise and Perry have developed a situational calculus [12]. Giunchiglia follows a more philosophical approach and sees context as a "subset of the complete state of an individual that is used for reasoning about a given goal" [13]. The key point is that designers can use sensors to capture and build high level context models of parts of the real world then using these techniques recognise and reason about situations.

3 Context-Awareness and Adaptation

In computing literature the term context-aware first appeared in [14] (1994). The following year one of the authors, Schilit [15], describes context-aware software as adapting according to the location, identities of nearby people, objects and changes to those objects. The primary goal of a context aware application or service is to be able to change its behaviour in response to a context change. Context-aware applications, context-aware artefacts or context aware systems are aware of their environment and circumstances and can respond intelligently. Adaptation therefore is an essential element of a context-aware system [16]. It is important to note that adaptation can be described in terms of adaptive and adaptable properties. An adaptive system is one that adapts to changes in user-related or environmental situations or context with the explicit goal of automatically assisting users. In contrast adaptability empowers end-users to customise or personalise computer systems according to their individual preferences. In other words it is an adaptable system. Adaptive and adaptable systems are complementary to each other [17] and when used together increase the match between user needs and system behaviour. The property of context-awareness can be applied to all types of applications and systems and as such has been identified as an essential feature of pervasive computing. The essential aspect however is that it enables automatic proactive assistance reducing human intervention. Many context aware applications can provide this automatic assistance by using logical context alone that is stored in profiles, databases or social websites. However with the proliferation of wireless sensor-actuator networks there is an increasing interest in context-aware systems that make use of external (physical) context factors such as location, presence, temperature and light information and interact with the environment. This capability to sense, reason and actuate has the potential to imbue the property of awareness to almost any artefact or object. This leads to the vision of the interconnection of artefacts embedded in our real environment, forming a society of "intelligent things" and "smart spaces".

4 Supporting Context Awareness

The development of context aware applications is a complex task because of the need to accommodate for a wide variety of context types and their values, including the ones that cannot be anticipated at the time when the system is designed. In order to handle this complexity many early examples of context aware systems were designed around specific applications and domains. This approach of hard-coding mappings between all possible combinations of context values and corresponding application behaviour is impractical. More importantly it makes context aware systems difficult to extend and almost impossible to introduce new applications without considerable re-engineering to cope with new context types. It is at best extremely demanding to foresee all contexts an application may encounter during its lifetime. Consequently the approach taken is to design a flexible context infrastructure capable of adapting to different applications.

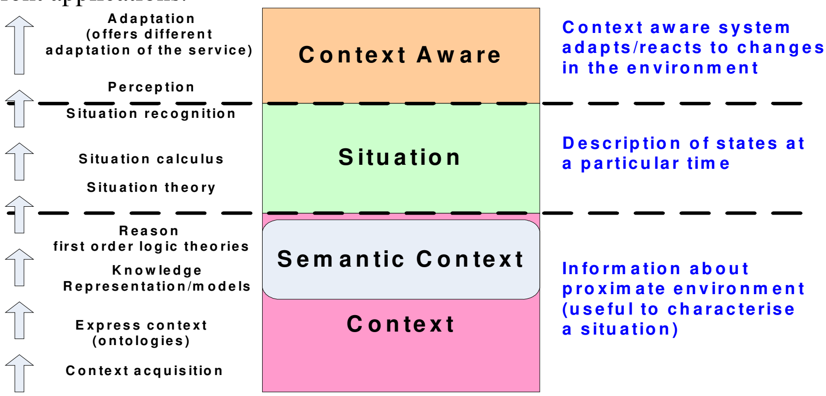

Fig. 1. Context Layers

As illustrated in Fig. 1, there are several layers of abstraction in a context-aware system and any context-aware middleware or architecture must therefore be capable of building representations and models of these abstractions. But these higher level abstractions can only be made from lower level context which requires some form of context management function. The main context management features are context acquisition, context aggregation & fusion, context dissemination, discovery and lookup. Context can be acquired from a diversity of sources from social websites, profiles, databases and physical sensors and filtered and aggregated to form higher level context. Context dissemination is the propagation of context to other entities. Context has a lifetime and must be continuously refreshed. Situation recognition entities and reasoning engines must find or lookup sources of relevant context. Similarly context sources must be able to publish or advertise the context that they have to offer. Distributed context dissemination and discovery requires considerable design effort in larger context aware systems. Establishing context quality is therefore an essential feature of any context management system.

In order to manipulate context information it must be represented in some form that is compatible with the models that will be used in the reasoning and situation recognition processes. These models could be object orientated, ontological, rule based, logic based, based on semantic graphs or fuzzy logic sets. Expressing context using just one representation is almost impossible since the range is from the most specific, for example a temperature reading, to the most abstract, the state of happiness. Furthermore the representation must lend itself to the reasoning and inference techniques to be used such as classification, taxonomies, data mining, clustering and pattern recognition. Reasoning mechanisms allow higher level context to be deduced or situations to be recognised. Reasoning is also used to check the consistency of context and context models. Finally it must be possible to query context models and context repositories in order match the spontaneous needs of context-aware applications.

5 Context-Aware Systems

Engineering a large scale context aware system capable of scaling to the size of an urban cellular network and supporting smart urban type services and applications is a demanding challenge. There are essentially two main parts to any such context-aware system:

- Context Management subsystem concerned with context acquisition and dissemination
- Context Modelling concerned with manipulation, representation, recognising and reasoning about context and situations

Taking into consideration the full sense-decide-actuate cycle of context-awareness then two other subsystems can be identified:

- Sensor and sensor network subsystem: this will include logical as well as physical sensors
- Actuation and Service Composition: Once the system/application becomes aware it adapts by actuating some device/display and or some service is automatically triggered and delivered. This could well be considered as two subsystems.

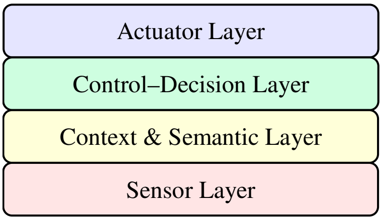

Fig. 2. Context Aware System Layers

There is a natural hierarchy and layering of these subsystems as illustrated in Fig. 2, which captures the complete sense-actuate cycle. This is very much a knowledge layered architecture. Although there are examples that are not, quite often context-aware systems are special examples of knowledge based systems. The processing techniques in such a system range from the fairly simple such as filtering, aggregation, feature extraction to machine learning, knowledge manipulation, rule based processing, situational recognition. Further, different types of context and situations require different types of processing techniques. Most often the context output of one process can be used as an input to another. Designing generic context-aware infrastructure that will support many different types of distributed context-aware services therefore requires a flexible model. This model must support a collaborative component based approach capable of connecting and distributing context between many context processing entities. A generic model that satisfies some of these constraints is a producer-consumer, publish-subscribe, broker model. This is approach is used in project C-CAST and other projects. All context processing entities can either be Context Consumers (CC), Context Providers (CP) or a combination of both. A Context Broker (CB) is required to discover and connect providers with consumers and is responsible for distributing relevant context amongst them. This may be done by query or by publish-subscribe notification methods. Many context processing entities will be both consumers as well as providers of context. Context sources such as WSN gateways, sensor platforms and web sites offering context can be wrapped to provide the common context provider interface.

All context-aware systems must be capable of adapting to context. One of the consequences of context adaptation particularly in a Service Orientated Architecture (SOA) is that adaptation can be partly satisfied by the ability to compose and orchestrate services on the fly, based on user and or environmental context. Web services, semantic web and match making algorithms can play a role in achieving this goal. Service composition is a dynamic and flexible process, which allows for reconfiguration as the context changes. It is not only services that will adapt, displays may change; doors open and traffic lights turn red. This is the actuation part of adaptation indispensable in a future internet of things and artefacts. This raises the issue of how sensor subsystems are best connected to such an infrastructure. For the internet of content then media will be recomposed and advertisements changed. In a distributed system to achieve these outcomes is a formidable challenge. It requires coordination, control and strategic decision making entities.

6 State of Practice

Building upon the context acquisition and context management models discussed in literature, numerous context management architectures have been proposed. Chen [25] presents a Context Broker Architecture (CoBrA), which is an agent-based architecture, for supporting context-aware systems in smart spaces (e.g., intelligent meeting rooms, smart homes, and smart vehicles). Other well-known examples for broker-based approaches are the SOUPA, and GAIA [26]. Many EU projects have explored context-awareness aspects, for example, SPICE [20], OPUCE [21] and MobiLife [19]. However the proposed approaches fail to completely offer a generic, scalable and flexible architecture supporting both evolving context models and evolving services and applications. Moreover, an efficient context diffusion and coherent integration into mobile communication services continues to be a challenging research area. Industrial research already addresses operations on context-aware information such as context capturing and context reasoning in [22]. In [23] Ti-Lab proposed a context management architecture, based on the producer-consumer-broker model. An application called TeamLife was evaluated on this platform during the Venice Carnival 2008. End users shared contextualised photos on a portal and in real time they were available at several locations of the city on mega-screens [24]. When a picture was taken the application automatically collected related context data, by inquiring the Context Broker [Fig. 3], and proceeds to a seamless machine tagging of the image.

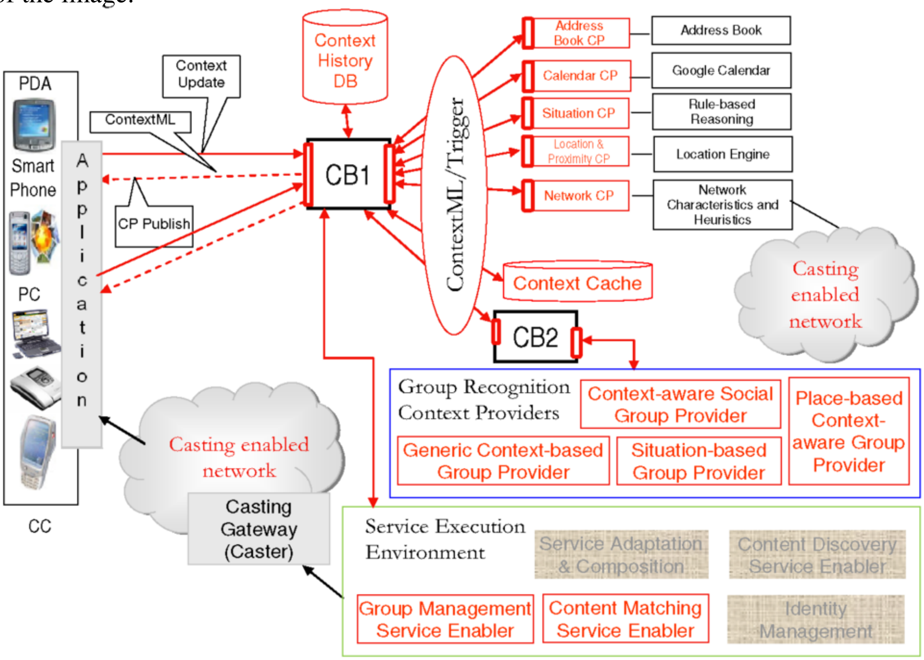

Fig. 3. C-CAST Context-aware platform

The context aware platform proposed by C-CAST is based on a distributed context broker concept and is an enhancement of TiLab's context platform presented in [24]. As illustrated in Fig. 3, the architecture currently encompasses various Context Providers: (1) Address Book CP offering access to address data and relations stored in profiles, (2) Calendar CP providing entries of Google calendars, (3) Location CP providing physical, logical location and proximity to other contextualised entities, (4) Situation CP inferring the users' situation (e.g. formal meeting) based on primitive context provided by other CP, (5) Network CP providing QoS measurements and other information related to the access network. A Context History database allows for storing obsolete context information. This can be used for autonomously learning from previous contexts and actions as well as for estimating user goals. Since C-CAST focuses on efficient group-aware content provisioning, the framework comprises several CP for dynamic ad-hoc group recognition. An event based triggering mechanism adopting the subscribe/publish paradigm is planned. In this mode the CC subscribes to context changes of interest and is asynchronously notified by the CP or by the mediating CB in case of event occurrence.

7 Conclusion

Context-aware applications, context-aware artefacts or context aware systems are aware of their environment and circumstances and can respond intelligently. This is the vision that those working in mobile, ubiquitous and pervasive computing are working towards. A future internet capable of embracing this concept and delivering context aware services to users and artefacts elevates this to a pervasive sensing and acting knowledge network. This would be a network able to make decisions, actuate environmental objects and assist users. There are some immense and fundamental challenges for the supporting infrastructure for scenarios of this type:

- This is a major change in the computing paradigm. These are pervasive adaptive systems that respond to things around them with no centralised authority.
- Services provided are more dynamic requiring discovery, ad-hoc composition and orchestration.
- It is a more knowledge based infrastructure that attempts to recognise situations and reason about the environment.

Setting aside the artificial intelligence aspects and concentrating on the network issues then our work in this area reinforces other researchers' experiences that:

- There are many context dissemination and discovery models and architectures. A standard architecture or middleware with well known interfaces is required. We have adopted a broker architecture style augmented with an event based publish subscribe mechanism. But most of this is middleware and perhaps not the concern of the network. Nonetheless discovery, lookup and event distributors do fall in the network domain.
- Representing, learning and reasoning about situations and circumstances in the real world is very hard. Many types of inference engines are required how do we plug them all together. Is this an integral part of the future Internet vision?
- A context aware system by definition is able to sense, adapt and respond. This may require designing in support for communication sense-decide-act loops into a new network. Furthermore if parts of the future network are context aware subsystems themselves designed with intelligent behaviour then there will be conflict. This will arise when components or subsystems make competing and conflicting decisions.
- Context leads to privacy problems. One of the most contentious issues is that gleaning of context is tantamount to monitoring people’s daily activities and situations. There must be simple mechanisms for users to withdraw from such services and applications.

Context-awareness is just one feature and for complete analysis we need to understand its position in relation to the holistic concept of a network, what it does and what users expect from it. The Internet network concept was well thought out. At the network layer IP packets are routed through to the end terminal with best effort delivery; packets can arrive in the wrong order, duplicated, missing or in error. It is the transport protocol that runs only in the end user terminal that provides end to end reliability. So there is minimalist expectation of the network. Therefore IP can usually work over any type of network. It is a huge global network and works well so why would we wish to change it? The user view of technology and expectation would be a good start. Evidence of social networking sites and blogs support the view that when users are given the capability creative and innovative user generated applications and content abound. The simple abstraction of a Web Page built over a network has been very good at this because users have a good understanding of the concept of a page and the technology is fairly intuitive. In contrast the number of users building applications and content for computers and mobile devices is much lower. There are so many operating systems (Vista/Windows {CE, XP}/Mac/Sybian/Linux), interfaces and releases that it is more difficult for the user. Most of the applications and services are owned by software companies or network operators. For the mass population it is still a fight to work with this technology. So the Clouds or Internet of Services is an interesting proposition. In the cloud computing paradigm people define what they want to run or store in the cloud and the cloud architecture does the rest. This is an ideal match for mobile devices with relatively small storage and computing power. So effectively a large part of computer operating system functionality is performed in the network. If the network can compose, orchestrate and deliver services to the user's device then the analogy is complete. The worry of file management, storage, versions and operating systems is left to the network operators. But this vision stands in complete contrast to the current Internet where all the functionality is in the end terminal by design. There is a way of composing services but this is using the Web Page abstraction and Web Services. Furthermore if when designing the Cloud vision effort was channelled into making it easier for composing user generated services then the move to this type of network is even more compelling. Open interfaces and service enablers with which to build applications available to the user community is the only way to empower user creativity and innovation. Contrast this with the situation of cellular networks today. The next step on from an Internet of Services is an Internet of Context-aware Services that incorporates an Internet of Aware-Things. The network would then contain all the entities that we have discussed in the first sections of this paper. Cloud computing or "The Network is the Computer" viewpoint is essential since large amounts of computing power are required to support machine learning, data mining, reasoning and situation recognition. It is a very exciting and challenging idea but what is the motivation? From a user point of view, if it works, then it places technology into the background and makes it more useful. Perhaps of more significance is its potential to promote sustainable living. The planet is under pressure, resources are finite, there is a drive to joined-up government, joined-up local services, transport and general infrastructure in a concerted effort to manage resources in a more sustainable manner. An intelligent network offering context-aware services can assist in this endeavour. In conclusion therefore having made this argument for the future internet vision and the role of context-awareness; is this not Web X.0 in disguise?

Acknowledgment

This work is supported by the European ICT project "C-CAST" - Context Casting - (Contract-No. ICT-216462) [18].

References

1. Ryan, N., Pascoe, J. and Morse, D. (1997) 'Enhanced reality fieldwork: the context-aware archaeological assistant', Proc. of the 25th Anniv. Computer Applications in Archaeology.
2. Dey, A.K. (1998) 'Context-aware computing: the CyberDesk project', Proceedings of the AAAI, Spring Symposium on Intelligent Environments, Menlo Park, CA, pp.51-54.
3. Hull, R., Neaves, P. and Bedford-Roberts, J. (1997) 'Towards situated computing', Proceedings of the First International Symposium on Wearable Computers (ISWC ‘97).
4. Brown, P.J. (1996) 'The stick-e document: a framework for creating context-aware applications', Proceedings of the Electronic Publishing, Palo Alto, pp.259–272.
5. Dey, A.K. and Abowd, G.D. (2000) 'Towards a better understanding of context and context-awareness', Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness, ACM Press, New York.
6. Prekop, P. and Burnett, M. (2003) 'Activities, context and ubiquitous computing', Special Issue on Ubiquitous Computing Computer Communications, Vol. 26, No. 11, pp.1168-1176.
7. Gustavsen, R.M. (2002) 'Condor - an application framework for mobility-based context-aware applications', Proc. of the Workshop on Concepts and Models for Ubiq. Computing.
8. Hofer, T. et al. (2002) 'Context-awareness on mobile devices – the hydrogen approach’, Proceedings of the 36th Annual Hawaii International Conference on System Sciences.
9. Zimmermann, A. et al. (2005) 'Personalization and Context Management.' User Modeling and User-Adapted Interaction 15, 3-4 (Aug. 2005), pp. 275-302.
10. Loke, S (2007) 'Context-aware Pervasive Systems', Auerbach Publications, New York.
11. Hendricksen, K. et al. (2003) 'Generating context management infrastructure from high-level context models', Proceedings of the 4th International MDM, pp.1–6.
12. Barwise, J. and Perry, J. (1983) 'Situations and Attitudes', Cambridge, MA: MIT-Bradford.
13. Giunchiglia, F., (1992). 'Contextual Reasoning', Trento, Italy.
14. Schilit, B. and Theimer, M. (1994) 'Disseminating active map information to mobile hosts', IEEE Network, Vol. 8, No. 5, pp.22–32.
15. Schilit, B. N. (1995) 'A System Architecture for Context-Aware Mobile Computing', PhD Thesis, Columbia University.
16. Ryan, N., Pascoe, J. and Morse, D. (1997) 'Enhanced reality fieldwork: the context-aware archaeological assistant', Proc. of the 25th Anniv. Computer Applications in Archaeology.
17. Oppermann, R., (2005) 'From User-Adaptive to Context-Adaptive Information Systems', i-com Zeitschrift für Interaktive und Kooperative Medien, 4(3): pp. 4-14.
18. Context Casting (C-CAST), a Specific Targeted Research Project in European Union’s ICT 7th Framework Programme, http://www.ist-ccast.eu
19. MobiLife, an Integrated Project in European Union’s IST 6th Framework Programme, http://www.ist-mobilife.org
20. Service Platform for Innovative Communication Environment (SPICE), An Integrated Project in European Union’s IST 6th Framework Programme, http://www.ist-spice.org
21. Open Platform for User-centric service Creation and Execution (OPUCE), An Integrated Project in European Union’s IST 6th Framework Programme, http://www.opuce.tid.es
22. Lamorte L. et al. (2007) 'A platform for enabling context aware telecommunication services', Third Workshop on Context Awareness for Proactive Systems.
23. Claudio Venezia, Carlo Alberto Licciardi (2007), 'Improve ubiquitous Web applications with Context Awareness, 11th ICIN 2007.
24. Moltchanov, B. et al. (2008) 'Context-Aware Content Sharing and Casting', 12th ICIN 2008.
25. Chen, H., Finin, T. & Joshi, A. (2003) 'An Intelligent Broker for Context-Aware Systems', Adjunct Proceedings of Ubicomp 2003.
26. M. Roman et al. (2002) 'GAIA: A Middleware Infrastructure to Enable Active Spaces', IEEE Pervasive Computing, vol. 1, no. 4, 2002, pp. 74-83.