Wireless World Research Forum
(WWRF)

Context-aware Communication in Ambient Networks

Raffaele Giaffreda, British Telecom, UK
Ahmed Karmouch, University of Ottawa, Canada,
Annika Jonsson, Ericsson, Sweden,
Anders M. Karlsson, TeliaSonera, Sweden,
Mikhail I. Smirnov[1] Fraunhofer FOKUS, Germany,
Roch Glitho, Concordia University, Canada
Alex Galis, University College London, UK

Subject Area: SIG3 - Self-organisation in wireless world systems

The paper defines research agenda for context-sensitive communication in wireless-world systems as discussed in WWI Ambient Networks[2] project of FP6

I. Introduction

Ambient Networks project aims at composition of heterogeneous networks based on self-organisation, i.e. performed co-operatively, on demand, transparently to the potential users, and without the need of pre-configuration or offline negotiation between network operators. This composition will ultimately enable users and operators to jointly exploit available radio and network resources for a broad range of services. In addition to networks composition, the AN vision examines the possibility for instant network composition to allow rapid adaptation of network topology as required for mobile networks. Such composed networks can be used for both service provision and the support of handover operations.

One of the main topics in ANs is to seamlessly incorporate context information into service provisioning, network composition, and service adaptation. Context information is *any* information that can be used to characterize the situation of an [networked] entity, where an entity can be a person, place, physical or computational object. Thus, context information is an essential enabler of cross-layer optimisation of a network-level service.

Network-related context information is the one about changes in the network; at the same time it can be used to cause desired changes, it can be used to make networks more receptive to users needs and enhance the users experience by making the communication easier and richer. For instance, making use of context information such as the capabilities of the available end-devices and the network connectivity of the user can optimize a streaming service for a mobile user. Exploiting network-related context information can make ambient networks simpler, more efficient, and more powerful thus simplifying the management of the networking infrastructure for network operators while providing end-users with value-added services and an enhanced communication experience.

In the rest of this paper, we first discuss research challenges and categorisation of attributes of context awareness based on the state of the art in this area; then we introduce Ambient Networks functional areas and briefly touch associated context information; we conclude the paper with presentation of our current architectural building blocks that reflect the so far identified concepts.

---

[1] Corresponding author: address: Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany, E-mail: smirnow@fokus.fraunhofer.de
[2] http://www.autonomic-communication.org/

II. Research Challenges

A. Context Sensitive Communications

Mobile computing followed by pervasive computing and ending with ambient computing are opening the door for a new era where human and computing resources are integrated into one world. This hybrid environment introduces challenging issues ranging from the network level up to the application level in conjunction with social and personal matter.

A key for solving most of these challenges lies in knowledge and ambient interaction. Knowledge is achieved through context while ambient interaction is achieved using context sensitive communications. Being the case and considering the characteristics of ambient computing environments (mainly heterogeneity, mobility and dynamics), we need new techniques to achieve reliable, when needed, non-intrusive communication. One way to achieve this is by making the applications context aware and the network context sensitive. This is termed as *context-sensitive communications* [1]. This type of context-sensitive communications supersedes client-server communications [2].

Context-sensitive communication is a type of communication [3], where the communication channel is established between devices based on some specific contexts. ***Context-sensitive inter-object communication*** is a type of context-sensitive communication between distributed context-sensitive objects, such that these objects are compatible to communicate with each other and both objects can be activated in the current context according to their context-sensitive interfaces.

The new challenges that arise in order to provide context-sensitive communications include [1]:

(i) systematic adaptation and customized representation of context,

(ii) context association in a real-time behaviour,

(iii) spontaneous processing and dissemination of acquired context.

Therefore, the basic requirement to provide context-sensitive communications is that the underlying system itself should be context-sensitive. An object-based context aware middleware is proposed for certain systems in order to deal with the above listed challenges and facilitate interoperability independently of device and network type, as in the Reconfigurable Context-Sensitive Middleware (RCSM) [3] and Object Request Broker (ORB) [2] while an agent based middleware is proposed in [1].

Except for proximity preference, there is no clear ongoing research on how entities can communicate with each other using contextual information as basis of communication. This includes initiating communication sessions, modifying sessions and terminating sessions automatically subject to context information.

This deficiency in current research is due to the focus of context on proximity only and thus cripples current context aware systems to have other contextual information as a fact in initiating communication and interaction between entities. This can be formulated as a challenge of

(iv) [direct] context triggered communication.

Furthermore, there is no obvious method on how context information can be successfully disseminated between entities (users, services) or between system components that would benefit from this context information in performing spontaneous tasks that are ***not explicitly specified*** by the entities but ***inferred from current situation***. This can be formulated as a challenge of

(v) [indirect] context-based inference triggered communication.

We envision a new method of communication that is capable of providing non-intrusive automated connectivity and information exchange in context aware environments. The communications in this ambient environment are sensitive to the changes and are capable of creating, modifying, adapting sessions and maintaining persistency of the contextual preferences of the sessions.

B. Characteristics of context sensitive communications

In previous work some co-authors have identified four characteristics that a general context sensitive communication protocol, CSCP [1], must handle. These characteristics leverage context from being an ambiguous concept to an essential active component in ambient computing.

1. **Mobility**: CSCP must support inter-domain and intra-domain mobility. It should also support distributed, peer-to-peer, and event-based computing. The mobility should be achieved while keeping persistent connectivity and seamless accessibility to resources.
2. **Management**: CSCP should support session management for real time and non real time applications. Management includes soft handover between different domains and network abstraction to the mobile users and applications. This requires state-full and stateless binding mechanisms and presence notification methods.
3. **Technology transparency**: CSCP should be independent of used technology and devices as much as possible. This will ease extensibility and interoperability between different ambient networks and administration domains.
4. **Association**: CSCP must associate services to users or project users' presence to other users, based on captured context from the environment or from user profiles stored in the system knowledge base. This will facilitate spontaneous invocation of services and presence-based communication.
5. **Personalization**: CSCP must provides seamless service provisioning accustomed to user requirements, supply familiar interfaces to users in the environment and storage capability taking advantage of the common understanding of context.

C. Categorizing Context Awareness

What is context? In Dey's review [4], the following definition was provided: context is *any information* that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. This definition lacks one essential component; *i.e.*, it does not cover all aspects of context state. In other words, the definition is more towards the external behaviour of context revealed in 'characterising the situation of an entity'. However, context itself has an internal state that needs to be added to the definition. **Context internal state defines the structure of context, its domain, range, and functionality**.

Following this definition the main challenge regarding context-aware networking, computing or applications is not the general question of 'what is context?', because context can be seen as everything around (and possibly within) an entity. In fact, if a piece of information can be used to **characterise the situation of an entity in an interaction** it can be identified as context. This however poses a requirement to an entity to be **open** to context-based interactions.

Context-aware computing is a computing paradigm in which applications can discover and take advantage of contextual information such as user location, time of day, nearby people and devices, and user activity. Many researchers since then have studied and explored context-aware computing and built several context-aware applications to demonstrate the usefulness of this new technology. Context-aware applications and system infrastructures to support them are extremely difficult to build and maintain. This is due to **lack of standard methods** to represent context and building context aware applications. This is also manifested by the diverse nature of context and devices that capture context. We now discuss the theory and background of context paradigm followed by related work on context awareness.

A definition of context-awareness is given in [5] as: a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.

As a categorisation of features for context-aware applications are listed in [4]: Presentation of information and services to a user, automatic execution of a service, and tagging of context to information for later retrieval. The exploitation of local resources, or resource discovery is not explicitly mentioned as a context-aware feature in [4], because it is considered as a part of the first two categories. In short, a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.

The next definitions of context-aware computing fall into two categories: **using context** and **adapting to context**. AN project is interested in these approaches because we define the notion of Quality of Context.

The first category corresponds to the general case of *using context*. Pascoe *et al.* [6, 7] define context-aware computing to be the ability of computing devices to detect and sense, interpret and respond to aspects of a user's local environment and the computing devices themselves. Dey *et al* [4] define context-awareness to be work leading to the automation of a software system based on knowledge of the user's context. Salber *et al* [8, 9] define context-aware to be the ability to provide maximum flexibility of a computational service based on real-time sensing of context.

The following definitions are in the more specific '*adapting to context*' category. The following references [10, 11, 12] define context-aware applications to be applications that dynamically change or adapt their behaviour based on the context of the application and the user. Brown [5] defines context-aware applications as applications that automatically provide information and/or take actions according to the user's present context as detected by sensors. Environment-directed applications are applications that monitor changes in the environment and adapt their operation according to predefined or user-defined guidelines.

Building on top of these definitions, researchers started categorising context as a technique for separating and clarifying the context's boundaries in different situations.

Tuulari [13] divides context awareness into two parts: **self-contained context awareness** and **infrastructure based context awareness**. The former implies context awareness achieved without any outside support and the latter implies context awareness achieved with outside support.

Chen and Kotz [14] extended Schilit's division of context into four categories to achieve a better understanding of the concept.

- **Computing context** includes network connectivity, bandwidth, communication costs, and nearby resources such as printers, displays, and workstations.
- **User context includes** user profiles, user location, and nearby users and people.
- **Physical context** includes lighting, temperature, and humidity.

Time context includes time of the day, week, year, and also the season of the year.

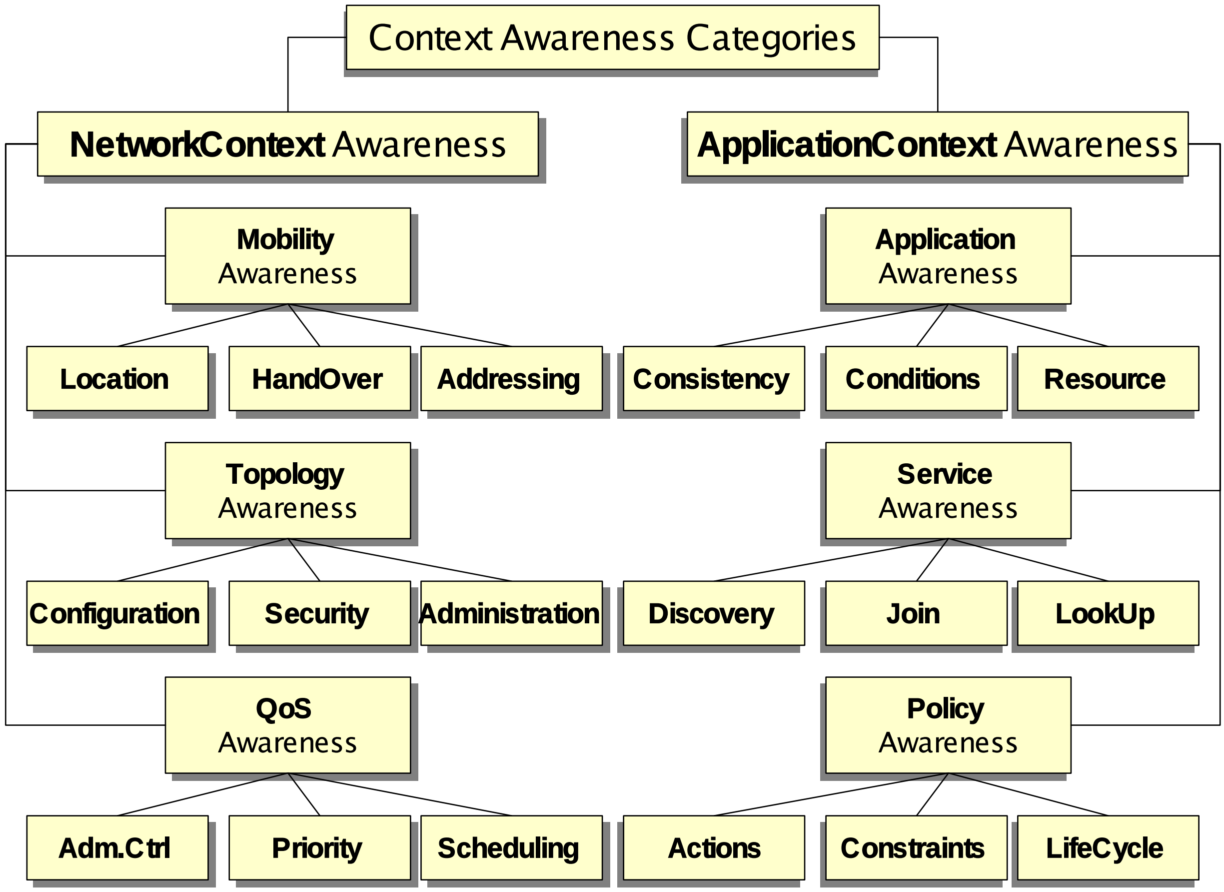

Figure 1: Categorization of ContextAwareness attributes

This context categorisation has the advantage over other classifications in the methodology used for classifications, which is based on the entities using context as input information. Context awareness is therefore classified into **application-related category** and **network-related category**. Each category has its own sub categories that further specializes context in its domain. In the application-related category, the application aware subcategory has *consistency*, used resources and *conditions* as the core functionalities that use context information to facilitate the process of inter, and intra application interactions. The service aware category includes *discovery, join*, and *lookup processes* for sighting services matching the required services that are currently needed. The other root category is the network aware category with subcategories such as mobility-aware - an essential component in context aware systems with functionalities such as hand over, location detection, and auto configuration according to current location. Other categories include topology aware and QoS aware.

III. Ambient Networks Approach

The AN project will create the network solutions for mobile and wireless systems beyond 3G. It will enable scalable and affordable wireless networking while providing rich and easy to use communication services for all. It is geared towards increasing competition and cooperation in an environment populated by a multitude of user devices, wireless technologies, network operators and business actors. Probably, the best impression of the project framework can be obtained from AN work on deriving requirements from scenarios [15] developed by various workpackages.

The project structure reflects Functional Areas (FA) of ambient networking, namely:

- Multi-Radio Access - FA2
- Network Composition and Connectivity - FA3
- Mobility and Moving Networks - FA4
- Smart multimedia routing and transport - FA5
- Context Aware networking - FA6
- Security - FA7
- Network Management - FA8.

Each FA develops various Functional Area Controls (FAC); each FAC in turn is actually a set of controls, a number of controls within a FAC and their mechanisms are FA specific; this is a major source of complexity of context handling in AN. To cope with this complexity we need a construct that shall (i) hide particular controls within FAC sets; (ii) be context data format agnostic, however (iii) useful enough for reasoning on inter-FAC context exchanges, and (iv) be extendable to cope with the fact that the AN project is developing its functional areas concurrently. This, yet to be formally described construct can be dubbed **facet** (FACet); a facet of an ambient network functional area is a collection of control mechanisms and their parameters and state data having a potential to serve as context sources or context consumers; facets are yet to be developed within each FA.

As concluded in the previous section, all control mechanisms are potential sources and consumers of context information originated in other FAC sets. In simple words, one man's context is another man's text. With one exception probably – context information originated within FA6 will be consumed by its own controls.

All FAC sets altogether form ambient control space; from a context aware networking this control space is a multi-faceted one (Fig.2).

In the reminder of this section we look at some finer grained constructs of a context facet construct being discussed in the project, namely:

1. Context Information Base
2. Cross-Layer Context Handle
3. Context Control Plane
4. Inter Domain Context Manager
5. Quality of Context.

<u>Context Information Base (CIB)</u> is solving the task of collecting, processing and distributing of context information from and to the AN entities in a generic and not application- or device-specific format.

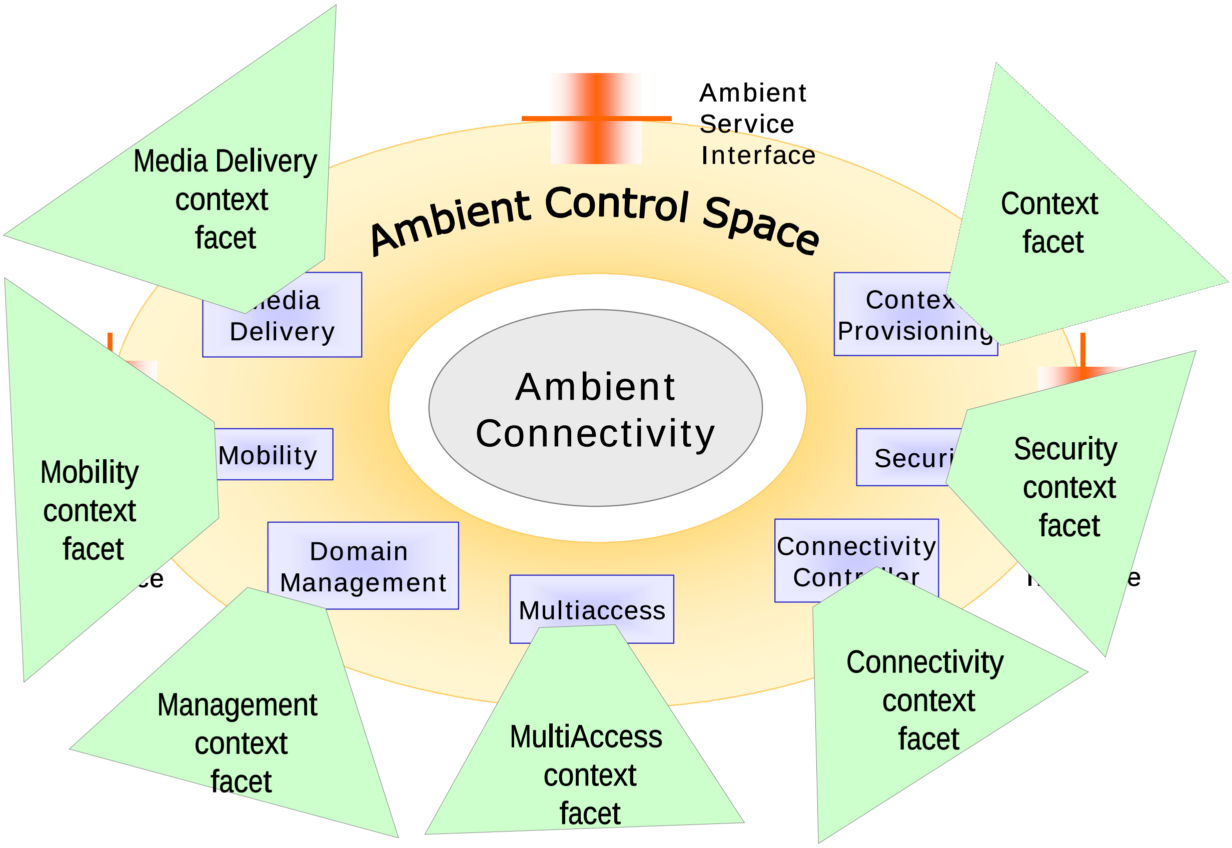

Figure 2: Context in multi-faceted ambient control space

CIB can be seen as a logical construct to represent a distributed repository, which is the source of up-to-date, coherent context information. The CIB concept separates concerns of a FAC and of its facet: FAC's concern is that of control of respective functionality (e.g. radio access, mobility, media routing, etc.), while facet's concern is on demand export of FAC's state data.

The CIB functionality can be summarised as:

- to encompass the logical source and sink for context data
- to provide efficient context handling mechanisms for
  - Context formalisms
  - Context data updatemaintenance and composition rules
  - Context integration
  - Context distribution

The CIB collects, disseminates and manages context information coming from different layers. These context sources at different layers (e.g., sensors detecting user location, bandwidth monitors) must interact with the specific context controls (e.g. notifying running applications of changes as soon as they happen).

<u>Cross-Layer Context Handle</u> construct captures all mechanisms and protocols needed for providing efficient handling of context information within a facet; it represents the “work behind the scenes” needed to ‘populate’ timely the Context Information Base with needed information given the diverse and distributed nature of this information.

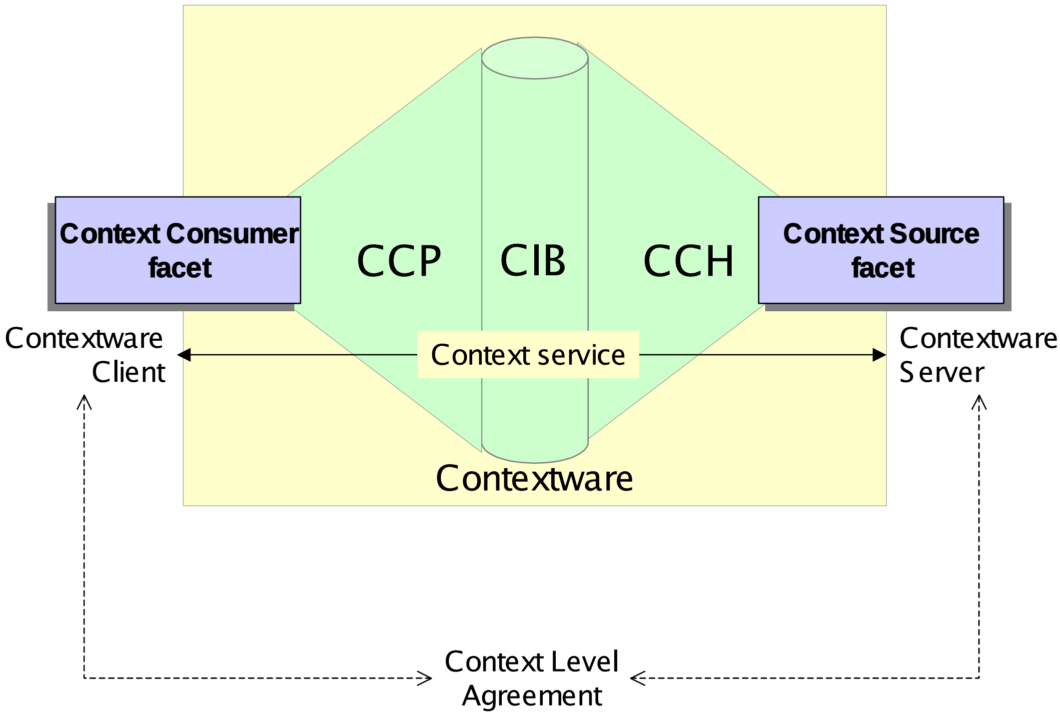

Figure 3: Major contextware constructs

Up-to-date, valid context information is the key requirement for successful utilisation of context information in services offered by Ambient Networks. Service adaptation based on outdated context information could definitely be most irritating for users. This in turn demands just-in-time collection, dissemination and management of context information. Therefore, the CIB must be kept up-to-date reflecting as much as possible a current situation (of a user, device, facet, network, etc.). This requires efficient collection, processing, management, and dissemination of context data, and consequently requires efficient interactions between context sources and context services.

The CCH construct can be implemented by a number of ways, including:

- Push mechanism: Context sources periodically push updated context information to the context service;
- Pull mechanism: Context service must explicitly request context information via polling or on-demand.
- Tuple shared space: Context sources publish their information there and context service read it whenever they need.
- Persistence mechanism: context consumers subscribe with certain context sources and these sources will provide them with the required context accordingly.

Context information can be categorised into several classes based on factors such as (volatile, non-volatile), (real time, non-real time), (private, public), (network-centric, user-centric), etc. Context information in these different classes may require different handling policy. For example, public and network-centric context information can be disseminated to everyone needing the information, but private and user-centric context information must be securely handled and disseminated only to authorised requestors.

The CIB presents this information to consumers; it may also cache the context information or further process context information in order to provide support for predictions about possible future states or actions. For example, for the mobility and session adaptability, the management station using further processed context information may constantly 'push' options for user's next move, based on previous mobility patterns.

Context Control Plane is a key concept to the definition of our contextware. As opposed to the previous concepts, the idea behind this one is satisfying more the "how do we make use of context information". The Context Control Plane represents a collection of policies and control functions delivering information to our contextware clients (i.e. it enables context-aware support of mobility, multi-access, media routing, network composition etc.)

The CCP functions include:

- Totality of context policies and agreements for an AN domain;
- Context agreements and lifecycle management of corresponding policies;
- Conflict detection and resolution mechanisms for policies and agreements;
- Inter-domain context negotiation to facilitate connections of AN context control functions with relevant functions from an other domain.
- Context registration and registrar functions, which advertise the presence of context control space functions in adjacent domains to allow dynamic composition of control functions.
- Aggregation functions, which support a simple plug-and-play and unplug-and-play context composition between AN domains;

Inter Domain Context Management: extending on our second construct, given the intrinsic diversity of domains present in Ambient Networks, Inter Domain Context Manager includes all the mechanisms that must be in place to facilitate collection and aggregation of context information potentially owned and used by different players.

The goal of the inter-domain context management is aggregation of context information from multiple domains. The domains hold separate parts of the needed context information, which must be available and coordinated as one. The first step is to reach an agreement on how to achieve this through a negotiation of the policies for context information, common formats, and methods of achieving the same level of context. The relevant context information that will be made available to other domains depends on the requested service, and especially for privacy reasons it is important that not more information than is needed is included.

Once an agreement on how to handle context information is reached between domains, the user's context information is dynamically updated across the domains according to the negotiated policies etc., and made available in aggregated and/or concatenated form to requesting context services.

Quality of Context: context information can be uncertain or incomplete by its nature. This last construct is meant to capture this specific aspect, which might eventually influence both the way control functions are enforced and context information is actually utilized.

Contextware networking services are dependent on the availability of context information, which must be provided at the right time and place (i.e. the availability of a distributed CIB). These parameters pertaining to context data give rise to the idea of Quality of Context (QoC). It is essentially any information that describes the quality of information used as context information, such as precision, probability of correctness, trust-worthiness, resolution and validity period.

IV. Future Research

The QoC is a construct that bears the essence of our research agenda; this concept would require a more precise definition based on solutions to the following problems and issues:

- What are the formal notations, specifications and operations for QoC?
- What is the appropriate value chain in terms of quality and price of aggregated context?
- How do we manage QoC?
- What is the impact of QoC on contextware service creation, deployment and management?
- What is the relationship between QoC and QoS?
- What is the lifecycle of context data? What determines its usefulness and validity from a temporal aspect?
- What are the relationships between the roles and actors within a contextware enterprise that may include for example Context Providers, Context Owners, Context Aware Services Providers, Context Brokers, Context Users and Context-aware Service users?
- What are the QoC agreements and contracts? Although a well structured information model may greatly help interpretation and comprehension of context data, looking at a higher layer of interaction between administrative domains, how do service providers conduct service negotiations that incorporate a user's dynamic context information?
- What is the right measure for Context aggregation? How to provide aggregation and/or concatenation of context information from single and multiple providers ensure 'pervasiveness' of service experience for end user?

The issue list can be continued of course; the area of context aware communication requires fundamental research that probably will contribute to creation of a new –network – information theory. Ambient Networks project will contribute to this research agenda, it is aiming to find and prototype pragmatic solutions for contextware. The contextware constructs presented in this paper is our first step in understanding of the power of context awareness.

V. References

[1] M. Khedr and A. Karmouch "Exploiting SIP and agents for smart context level agreements", 2003 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, BC, Canada, Aug 2003.

[2] S. S. Yau and F. Karim, "Context-Sensitive Object Request Broker for Ubiquitous Computing Environments", Proc. 8th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS 2001), Bologna, Italy, Oct 31-Nov 2, 2001.

[3] S. S. Yau and F. Karim, "An Adaptive Middleware for Context-Sensitive Communications for Real-Time Applications in Ubiquitous Computing Environments", t Real-Time Systems, The International Journal of Time-Critical Computing Systems, Kluwer Academic Publishers, Dordrecht, The Netherlands, vol. 26, no. 1, pp. 29-61, Jan 2004.

[4] Dey, A. K., Understanding and using context, Journal of Personal and Ubiquitous Computing, Volume 5 (1), pp. 4-7, 2001

[5] Brown, P.J.; Bovey, J.D.; Chen, X., Context-aware applications: from the laboratory to the marketplace, IEEE Personal Communications, Volume: 4, Issue: 5, pp. 58 – 64, October 1997

[6] Pascoe, J., N. Ryan, N., Morse, D., Issues in developing context-aware computing, In Proc. First International Symposium on Handheld and Ubiquitous Computing (HUC'99), 1999

[7] Pascoe, J., The Stick-e Note Architecture: extending the interface beyond the user, in Proc. 2nd International Conference on Intelligent User Interfaces, January 1997

[8] Gray, P.D., Salber, D., Modelling and Using Sensed Context Information in the Design of Interactive Applications, in Proc. 8th IFIP Working Conference on Engineering for Human-Computer Interaction (EHCI'01), May 2001

[9] Salber, D, Dey, A.K., Abowd, G.D., The Context Toolkit: Aiding the Development of Context-Enabled Applications, in Proc. CHI'99, May 1999

[10] Schilit, B., Adams, N., Want, R., Context-aware computing applications, in IEEE Workshop on Mobile Computing Systems and Applications, 1994

[11] Dey, A.K., Salber, D., Abowd, G.D., Futakawa, M., An architecture to support context-aware applications, GVU Technical Report Number:GIT-GVU-99-23, 1999

[12] Crowcroft, J., Hand, S., Mortier, R., Roscoe, T., Warfield, A., Plutarch: An argument for network pluralism, ACM SIGCOMM 2003 Workshops, August 2003

[13] Tuulari, Esa. Context aware hand-held devices. Espoo, VTT Electronics, 2000. VTT Publications

[14] Chen, G., Kotz, D., A survey of context-aware mobile computing research, Technical Report, TR2000-381, Dept. of Computer Science, Dartmouth College, November 2000

[15] Börje Ohlman, B.J. Busropan, Jens Gebert, Nick.Papadoglou' Experiences from using scenarios to derive requirements for the Ambient Networks project, Ambient Networks, 2004, WWRF 11 Proceedings (this volume)