

Towards Context-Aware Telecom End User Services through SOA

Philipp H. Mohr¹, Giuseppe Raffa¹, Marina Pettinari¹, Tullio Salmon Cinotti¹,
Cristina Frà², Claudio Venezia², and Paolo Protto³

¹ ARCES Centre of Excellence, University of Bologna, Italy
{pmohr,graffa,mpettinari,tsalmon}@arces.unibo.it

² Telecom Italia Lab, Torino, Italy
{cristina.fra,claudio.venezia}@telecomitalia.it

³ University of Torino, Italy
paolo.protto@guest.telecomitalia.it

Abstract. In this paper a SOA inspired context-aware platform for enabling context-aware Telecom services is presented. The SOA related components are highlighted and the platform's integration into a Telecom provider's service architecture is described — pointing out challenges in terms of protocols and semantic interoperability. An example application bridging the gap between data collected through a user's mobile phone and a web based portal in order to provide a Virtual Location Application through our platform is presented.

1 Introduction

As the terms Services Oriented Architectures (SOA) and context are being used in a number of different ways, we first clarify our working dependencies:

- **SOA** is a design philosophy which allows the integration and interoperability of a wide range of different services and data sources, where individual components do not need to know specific implementation details about others, but can solely access data through well defined interfaces and protocols.
- **Context** is any information that can be used to characterise the situation of an entity (Entity: person, place, object that is considered relevant to the interaction between a user and an application) [1].
- 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 [1].

SOA has reached a mature state in the web domain, but not yet in the area of Telecom providers, this is partly due to the previous focus of Telecom providers on their current core business: voice calls, messaging and broadband connectivity. Telecommunication services have been typically realised through separate, only vertically integrated service specific networks. Some consistencies exist at the transport layer, but at the service layer there are almost none, due to proprietary

platforms. Currently Telecom providers are evaluating the benefits of moving towards a Service Oriented Architecture which will simplify the integration and combination of existing web services with Telecom specific services, i.e. SLEE and IMS based ones [2], tailored towards the Telecom world. One of the biggest challenges for Telecom providers is to find a way of leveraging on their unique position of being in control of the network and of the information that passes through it, but still open up their network to a large enough extend to allow for innovation. By mastering this challenge, new revenue streams can be generated from context-aware services, which in addition also allow for a novel utilisation of the existing operators' network infrastructure and knowledge costumers about their customers.

Currently a lot of work is being performed concerning the integration of IT and Telecom services — a range of hybrid services are already offered by Telecom providers, but they are mainly static (i.e. pre-configured) and do not adapt to the needs of individual users. For Telecom companies to be able to provide novel and dynamic context-aware services which adapt to end users, a modular and context oriented architecture is required. This paper describes such a context-aware architecture, designed in the spirit of the SOA paradigm, which enables context-aware Information and Communications Technology (ICT) services — seamlessly integrated into existing SOA architectures. The SOA paradigm is well suited for the dynamic context-aware computing domain, as context sources are bound to change frequently, requiring them to be integrated on the fly, and end user services need to be dynamically adapted and composed if a context change occurs.

This paper is concerned with how a SOA enabling context-aware architecture can be integrated into the world of ICT providers, enabling novel context-aware end user services which also make use of Web 2.0 client capabilities. In section 2 our architecture is presented and how it can be integrated into the existing service world is presented in section 3. An example of an end user service which combines context collected through a mobile phone of a user with a web portal is described in section 4. Section 5 relates our architecture to work of others and section 6 provides a conclusion and suggests future work.

2 Context-Aware Platform

Our SOA enabling architecture, called Context-Aware Platform (CAP), for realising novel context-aware end user services, consists of six main components: a context broker, context sources, context providers, aggregators, context consumers, and an ontology aligner. These components enable CAP to provide end user services and external entities with context information acquired from external web services, user based sensors, learning, aggregation and reasoning modules. The interconnection between all components and their separation into areas of control is shown in Figure 1 in the form of a sample setup; all components are described in terms of their SOA properties below — for a more detailed description please refer to [3].

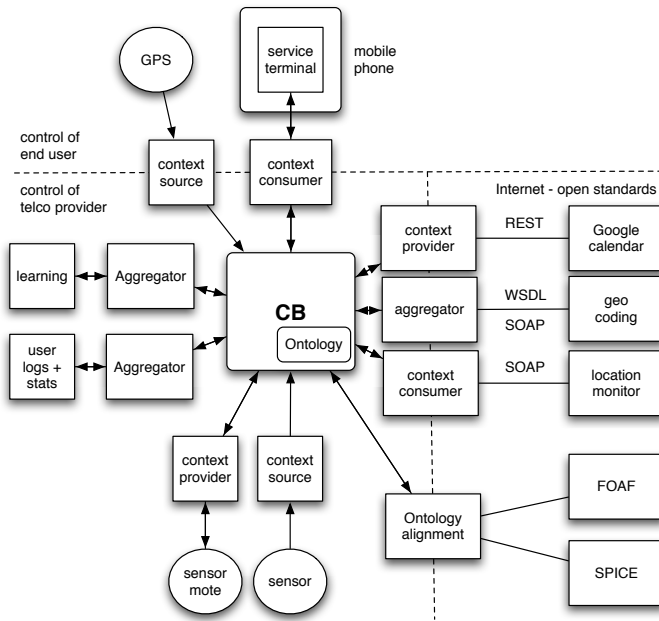


Fig. 1. CAP connected to the end user and the Internet

- The context broker (CB) is the central part of the platform, enabling the integration and combination of a wide range of different platforms and services, ranging from mobile devices, web services to low level sensors. It caches context information from heterogeneous sources and distributes it amongst all attached modules — enabling dynamic compositions of novel services based on context. Native to the CB is an ontology component which associates semantic meaning with context and enables the inference of novel context. The interaction between the CB and internal context providers is tightly coupled in order to increase performance. Providers external to the architecture are integrated through dynamic interfaces and context in the CB can be made public through context consumers. The CB also provides subscription / notification mechanisms, which allow other entities to subscribe to specific context changes. Context is encoded in ContextML [3], a XML based language specifically designed with efficient context exchange in mind. The language is currently still under development, with the possibility to incorporate further semantic meaning if required.
- The context providers and context sources enable context to be imported into the CB. External context from web based services is retrieved through open standards, i.e. SOAP and REST. Examples of implemented external context providers are: Google calendar is accessed through a REST interface, Microsoft MapPoint is accessed through SOAP and Magic Map [4] Wifi

localisation is also accessed through SOAP. Local context from within the Telecom provider network or user devices is collected through context providers and context sources following the ContextML guidelines. Where context sources simply push context to the CB, e.g. GPS data and inertial tracking co-ordinates [5], and context providers allow the CB to request specific context, e.g. from multi sensor platforms like Sun Spots [6] or Motes [7].

- The context consumers enable context to be retrieved from the CB for internal use or made available to external entities in the form of a web service through SOAP and REST interfaces.
- Context aggregators are a combination of context consumers and providers: they take context as input and produce higher level context by converting, fusing, reasoning on or learning from lower level context. Implemented aggregators are for example the reverse geocoding service accessible through the Microsoft MapPoint web-service interface, which enables GPS co-ordinates to be converted into street names and a learning module which learns common behaviour of users based on their GSM/UMTS cell IDs.
- The ontology alignment module allows interoperability with context provided to and consumed from external entities. A supportive point for ontologies often mentioned is their re-usability, which is valid when they are used for describing or annotating data, but when they are used for reasoning their structure needs to be optimised for performance and hence often requires a re-development. Interoperability amongst ontologies specifically developed for different platforms with different goals in mind is achieved by aligning the semantic meaning of attributes, e.g. of GPS location, with their counterparts in another ontology or knowledge representation. Currently alignments to the Mobilife [8], SPICE [9] and FOAF [10] ontologies, and the geo microformat [11] are implemented.

End user services have the potential of becoming context-aware, as more user specific context is made available through sensors integrated in the current and next generation mobile devices, e.g. GPS receivers, RFID readers and accelerometers, and lightweight online learning services tailored to the sensors for learning users common behaviour and preferences are developed. Through CAP a rapid development of such context-aware services is possible, only being required to implement a single interface to the CB which allows the required context for the service to be requested. The CAP takes away the complexity of the end user services from the user's device as it allows for the service to be developed in the form of a context consumer or aggregator situated within the network of the Telecom provider. This simplifies the process of introducing new services, as no or only limited updates at the client side are required. Furthermore, interoperability amongst context-aware applications/components is achieved, leading to a better re-utilisation of already implemented services.

On one hand, the CB architecture can lead to a bottleneck in terms of performance due to being centralised, but this shortcoming can be overcome by introducing multiple CBs, each handling a limited number of clients. On the

other hand, due to being centralised the billing process is simplified and security policies can be enforced more rigorously.

3 CAP Integration into SOA Architectures

The integration of CAP into SOA architectures provides existing IT architectures with context information — working towards solving one of the challenges next generation ICT SOA platforms are facing: context-aware services. In order to fully integrate CAP into existing IT architectures, allowing them to become context-aware, a two way integration is required: CAP needs to provide context to third parties through standardised Web-Service interfaces with context notification capabilities and third party IT Web Services need to be able to provide context information to CAP. The first integration is performed through context consumers, which implement standard Web-Service protocols and the second one is performed through context providers, which also implement standard Web-Service protocols.

By providing additional extensions for asynchronous context change notifications, i.e. WS-Notification and WS-Event, the full context provider, consumer and aggregator capabilities can be utilised and a seamless integration achieved. The requirement of supporting these capabilities is widely agreed and a number of SOA extensions to support event based message exchange exist — known as Event Driven Architectures [12].

The above enables an integration of CAP into SOA architectures, but with limitations common to the Web-Services world: the lack of semantic compatibility. A tight and seamless integration also requires alignment of context on a semantic level. The introduction of semantics improves the service context matching phase dramatically and allows a more straightforward integration of new context sources. In CAP this task is performed by the ontology alignment component, which can read in attributes in one semantic representation and output the attribute in another representation.

4 CAP Integration in Web 2.0 Applications

CAP does not only enable end user services running on mobile devices, but also ones bridging the world of a Telecom provider and Web 2.0 applications, an example of such a service is the Virtual Location Application (VLA) developed by TiLab [13] — VLA is currently in a prototype state and undergoes testing with 20 users. It allows users to manage their visited locations through a web application — Figure 2 and 3 show screen shots of the Virtual Location Portal. Through the CAP a user can request the VLA to be activated, which results in the gathering of context from the user's GPS receiver and cell ID monitor through context sources by the CB. Once the VLA has been activated the user can access the application via a Web 2.0 interface which retrieves user localisation data through a Web Service and performs a mesh up, giving the user the possibility to visualise his preferred locations directly on a map. A Tilab

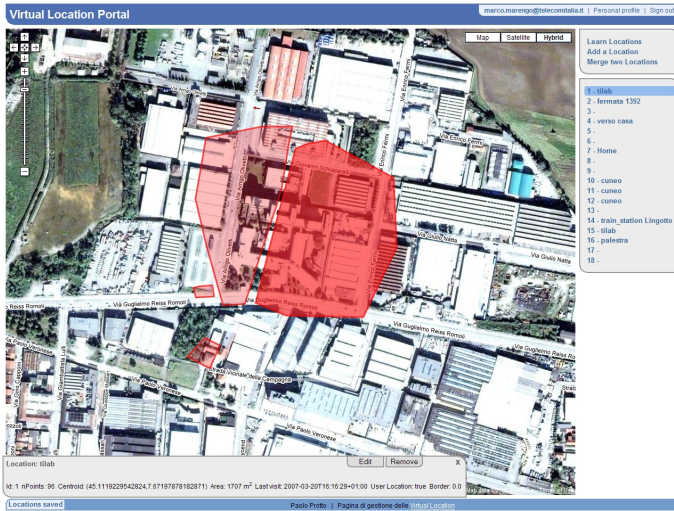


Fig. 2. Virtual Location Application screen shot

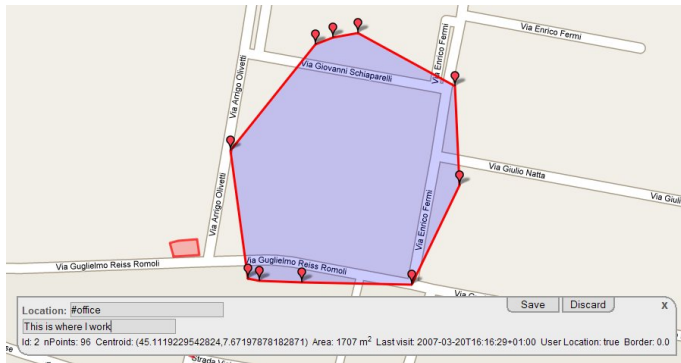


Fig. 3. Virtual Location Application screen shot

proprietary algorithm clusters the locations to ease visualisation: frequently visited areas are presented as graphic polygons which can be altered by the user through moving the edges of the polygons. In future versions AJAX capabilities could enable the overlay localisation data to be update on the map in real-time as the user moves around, as well as identifying colleagues or friends as they enter the area by simply subscribing to their localisation data. All recorded GPS coordinates and cell IDs are processed by a clustering algorithm which identifies the recurring places of a user. The portal allows the management of identified places: the user can add, delete or modify each place, by choosing a description and / or place type from a list of predefined semantic tags, e.g. home, office and restaurant. The CAP components involved in the VLA are shown in Figure 4.

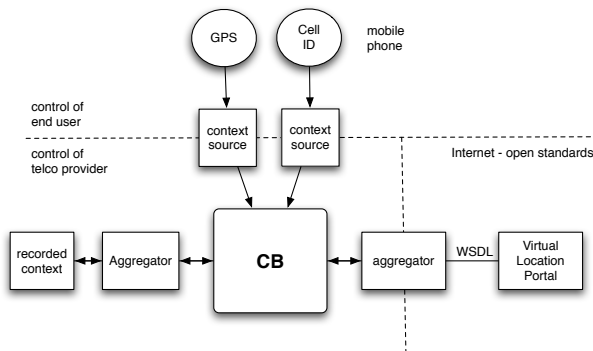


Fig. 4. CAP components utilised by the VLA

5 Related Work

A number of architectures for managing context exist, e.g. ContextToolkit [14], MobiComp [15] and the Context Kernel [16]. What sets CAP apart from the others is that it has been specifically designed for the requirements of a Telecom provider, for details see [3]. Only the Context Kernel has been designed as a true web service for storing and retrieving context, following the SOA paradigm. It allows an interaction through standardised protocols as it is possible with CAP's web service interfaces. In addition to Arruda *et al.* [16], the advantages of enabling context-awareness in a SOA are also pointed out by Capilla [17] and Almeida [18] — namely that context enables novel services adapting to the current situation, from virtual reality applications to routing services.

The integration of Telecom provider services and web services is an active area of research. Solutions are described and discussed by Venezia and Falcarin [2].

Another area of interest to our platform are adapting user interfaces, i.e. the W3C Delivery Context Interface (DCI) [19]. The DCI defines a framework for mobile browser interfaces, enabling web applications to dynamically adapt to the context of end users, e.g. by collecting user preferences, device capabilities and environmental conditions. It is a generic framework and specific implementations depend, e.g. on device manufacturers and are often subject to changes. The framework is a joined effort of different groups inside and outside the W3C. In order to participate in the DCI specification, a feedback loop between the device attributes present in the ontology used within CAP and the DCI ontology could be created.

6 Conclusion

This paper has described the integration of CAP, a SOA oriented context-aware Telecom platform, into existing SOA architectures, enabling the creation of context-aware services tailored towards individual users. Due to the modularity

of CAP and its possibility to distribute diverse context, it can enable a wide range of services. Despite the general move towards a truly distributed service realisation, a central CB approach was chosen as it allows for more straight forward and optimised learning and reasoning algorithms, as well as tighter security and privacy policies fundamental when offering personalised end user services. Shortcomings in terms of performance can be overcome by introducing multiple CBs, each handling a limited number of clients.

Through the existence of web and Telecom standards an interoperability on a protocol level of service components from different vendors has been made possible, but interoperability for context exchange on a semantic level has not reached the same maturity yet. A number of ontologies and microformats [11] providing such semantic interoperability have been released, but they are not widely accepted yet. One initiative trying to achieve a wider acceptance is the European initiative Service Platform for Innovative Communication Environment (SPICE) [9]. It is a consortium of Telecom providers, infrastructure manufactures and universities, trying to enable and unify the fusion of various services in the telecom domain. CAP has been designed with interoperability in mind, which can be achieved with SPICE on a semantic level through ontology alignment or even through an integration of the SPICE ontology once it is fully developed and has been widely accepted.

In order for context-aware services to widely emerge, service matching and dynamic composition based on context are essential. One of the next steps in our service architecture development is the integration of such matching and composition mechanisms.

References

1. Dey, A.K., Abowd, G.D.: Towards a better understanding of context and contextawareness. Technical Report GIT-GVU-99-22, Georgia Institute of Technology, College of Computing (June 1999)
2. Venezia, C., Falcarin, P.: Communication web services composition and integration. In: ICWS 2006: Proceedings of the IEEE International Conference on Web Services (ICWS 2006), Washington, DC, USA, pp. 523–530. IEEE Computer Society Press, Los Alamitos (2006)
3. Lamorte, L., Licciardi, C.A., Marengo, M., Salmeri, A., Mohr, P., Raffa, G., Roffia, L., Pettinari, M., Cinotti, T.S.: A platform for enabling context aware telecommunication services. In: Third Workshop on Context Awareness for Proactive Systems, Guildford, UK (2007)
4. Humboldt-Universität zu Berlin : Magicmap: Wifi location system (accessed 11-July-2007), <http://www.magicmap.de>
5. Cinotti, T.S., Di Stefano, L., Raffa, G., Roffia, L., Pettinari, M., Mola, M.: Dead reckoning supports stereo vision in pedestrians tracking. In: UbicompFourth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2006), Work-in-Progress Session, Pisa, Italy, pp. 628–631. Springer, Heidelberg (2006)
6. Sun Microsystems: Sun spots (accessed 11-July-2007), <http://www.sunspotworld.com>

7. Intel: Motes (accessed 11-July-2007), <http://www.intel.com/research/sensornets/>
8. Floreen, P., Przybiski, M., Nurmi, P., Koolwaaij, J., Tarlano, A., Wagner, M., Luther, M., Bataille, F., Boussard, M., Mrohs, B., Lau, S.L.: Towards a context management framework for mobilife. In: IST Mobile & Communications Summit, Dresden, German (2005)
9. Cordier, C., Carrez, F., Van Kranenburg, H., Licciardi, C., Van der Meer, J., Zoric, J., Spedalieri, A., Le Rouzic, J.-P.: Advanced beyond 3g service delivery environment: the spice service platform design principles. In: 16th Wireless World Research Forum Meeting The needs of Emerging Markets (WWRF16), Shanghai, China (2006)
10. Friend of a Friend (accessed 11-July-2007), <http://xmlns.com/foaf/spec/>
11. Microformats (accessed 11-July-2007), <http://microformats.org>
12. Michelson, B.M.: Event-driven architecture overview (February 2006) (accessed 11-July-2007), <http://soa.omg.org/Uploaded%20Docs/EDA/bda2-2-06cc.pdf>
13. Telecom Italia Lab (accessed 11-July-2007), <http://www.telecomitalialab.com>
14. Dey, A.K., Salber, D., Abowd, G.D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction* 16 (2001)
15. MobiComp: Context service (2007) (accessed 22-June-2007)
16. Arruda Jr., C.R.E., Neto, R.B., da Graça Campos Pimentel, M.: Open context-aware storage as a web service. In: *Middleware Workshops*, PUC-Rio, pp. 81–87 (2003)
17. Capilla, R.: Context-aware architectures for building service-oriented systems. In: *CSMR*, pp. 300–303. IEEE Computer Society, Los Alamitos (2006)
18. Almeida, D.R.D., Baptista, C.D.S., Silva, E.R.D., Campelo, C.E.C., Figueiredo, H.F.D., Lacerda, Y.A.: A context-aware system based on service-oriented architecture. In: *AINA 2006: Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA 2006)*, Washington, DC, USA, vol. 1, pp. 205–210. IEEE Computer Society Press, Los Alamitos (2006)
19. W3C: Delivery context interface (accessed 10-September-2007), <http://www.w3.org/2006/10/uwa-charter.html>