A previous version of this paper was published, as an invited paper, in the International Journal of Web Information Systems, 5(1):5 - 31, (c) Emerald, 2009.
Last update: 4 August, 2009

A Survey on Context-aware Web Service Systems

Hong-Linh Truong and Schahram Dustdar
Distributed Systems Group, Vienna University of Technology
{truong,dustdar}@infosys.tuwien.ac.at

Purpose of this paper	This survey aims at studying and analyzing current techniques and methods for context-aware Web service systems, discussing future trends and proposing further steps on making Web services systems being context-aware.
Design/methodology/approach	We analyzed and compared existing context-aware Web service-based systems based on techniques they support, such as context information modeling, context sensing, distribution, security and privacy, and adaptation techniques. Existing systems are also examined in terms of application domains, system type, mobility support, multi-organization support and level of Web services implementation.
Findings	Supporting context-aware Web service-based systems is increasing. It is hard to find a truly context-aware Web service-based system that is interoperable and secure, and operates on multi-organizational environments. Various issues, such as distributed context management, context-aware service modeling and engineering, context reasoning and quality of context, security and privacy issues have not been well addressed.
Research limitations/implications (if applicable)	The number of systems analyzed is limited. Furthermore, the survey is based on published papers. Therefore, up-to-date information and development might not be taken into account.
What is original/value of paper	Existing surveys do not focus on context-awareness techniques for Web services. This paper helps to understand the state of the art in context-aware techniques for Web services that can be employed in the future of services which is built around, amongst other, mobile devices, Web services, and pervasive environments.

A Survey on Context-aware Web Service Systems

Hong-Linh Truong and Schahram Dustdar
Distributed Systems Group, Vienna University of Technology
{truong,dustdar}@infosys.tuwien.ac.at

Abstract

Being context-ware will improve how software adapts to dynamic changes influenced by various factors during the operation of the software. Context-aware techniques have been widely applied in different types of applications, but still are limited to small scale or single-organizational environments due to the lack of well-agreed interfaces, protocols, and models for exchanging context data. Web services technologies can support and simplify the exchange of context information in large scale, multi-organizational environments, thus enable Web services systems to utilize various types of context information to adapt their behaviors and operations to dynamic changes. Until now, an overview of techniques and methods suitable for the development of context-aware Web services is missing. This survey aims at studying and analyzing current techniques and methods for context-aware Web services systems, discussing future trends and proposing further steps on making Web services systems being context-aware.

Keywords: Web services, context awareness

1 Introduction

Web services technologies have advanced the development of Internet-based applications by facilitating the integration and interoperability of services provided by different organizations. As a result, instead of developing monolithic applications, today, we build large scale software applications by composing loosely coupled services. By doing so, we can reuse and select suitable services as various organizations can provide similar services for the development of different applications. However, when services are loosely coupled to provide functions for a particular application, it is not easy to ensure that these services are aware of each other and of the overall context associated with the application and its usage. This awareness is, in particular, important because it improves the adaptation of services and applications to situational changes during their operations.

What context information and context-aware system are, has been defined and discussed in various papers (Dey, 2001) (Nihei, 2004) (Henricksen, et al., 2005) (Chen & Kotz, 2000) (Baldauf, Dustdar, & Rosenberg, 2007). Context information is dependent on individual systems, as a type of information might be considered as context information in one system but not in another one. Context-aware Web services are a subtype of context-aware systems defined in these papers. In this paper, we consider context information as any additional information that can be used to improve the behavior of a service in a situation. Without such additional information, the service should be operable as normal but with context information, it is arguable that the service can operate better or more appropriately. In this sense, a

context-aware service is considered as a smart Web service defined by Manes: *"a web service that can understand situational context and can share that context with other services"* (Manes, 2001).

When applications are built from different services provided by and hosted in multiple organizations, it is challenging to make the applications context-aware, as this requires services to be aware of each other and aware of the context of customers and applications. This challenge is due to the distributed, large-scale, diverse nature of Web service-based environments. Unlike past context aware systems of which components are tightly coupled and in a closed environment, such as (Yan, et al., 2000) (Roman, et al., 2002), as indicated by (Manes, 2001) (Nihei, 2004) (Truong, et al., 2008) and others, solutions for enabling context sensitivity and sharing in Web services-based environments must be open and standard based. These solutions cannot be proprietary and must be interoperable to ensure that they can be applicable in Web service-based systems. Furthermore, as like other types of information in Web service-based environments, context information can be sensitive and needs to comply with privacy and security rules, when sensed and shared across the boundary of a single organization. This makes techniques and methods for context-aware Web services very different from those of past context-aware systems. The centralized models, based on proprietary or application-specific protocols and data representations, as can be seen in the past context-aware systems, cannot be applicable in Web service-based environment.

Consider the trend of "the Future Internet"[1] which is built around, amongst other, mobile devices, Web services, and pervasive environments, and consider the lack of well understanding on state of the art in context-aware techniques for Web services, in this paper we aim at analyzing current techniques employed in context-aware Web services. This paper presents a survey on Web services context awareness, examines context sensitivity and sharing techniques, and discusses findings and recommendations for the development of context-aware Web service systems.

The rest of this paper is organized as follows: Section 2 discusses the motivation for this survey and related work, and outlines context-aware systems. Section 3 presents an overview of general context-aware systems and context-aware Web service systems. Section 4 provides a detailed analysis of existing techniques. We present our discussions in Section 5. A conclusion is given in Section 6.

2 Related Work and Motivation

Being context-aware allows software not only to be able to deal with changes in the environment the software operates in, but also being able to improve the response to the use of the software. That means context-awareness techniques aim at supporting both functional and non-functional software requirements. It is, therefore, not surprising that there are already many surveys related to context-awareness in general.

However, existing surveys are not performed for context-aware systems in the Web services domain. This is partly due to the fact that context-awareness computing has been focused on pervasive environments (Abowd, Ebling, Gellersen, Hunt, & Lei, 2002), HCI (Human-Computer Interaction) (Moran & Dourish, 2001), and mobile computing (Chen, et al., 2000) (Solarski, et al., 2004) for a long time, while Web services technologies have been relatively new. While many context-aware techniques could be useful for Web service-based systems, it is not clear to which extent they are useful and how to apply them. Web services technologies are designed for large-scale, loosely coupled environments that typically span multi-organization boundaries. Such environments have very different requirements compared with closed or

[1] http://www.future-internet.eu/. Last access: August 8, 2008.

single-organizational ones. Most existing context-aware techniques are designed for tightly-coupled systems, within the control of a single organization. The interaction model between components in a Web service-based system is also different from that of components in a single-organization environment. Therefore, in order to understand how existing techniques can be used for Web services, it makes sense to survey how context-awareness techniques are implemented in Web services and which ones can be reused.

On the other hand, today, pervasive environments include various services built atop Web services technologies. Mobile applications typically access Web services-based information systems. Web, mobile and desktop applications are blended into pervasive, Internet-based environments that rely heavily on Web services technologies. It is, therefore, useful to analyze the current practice and future trends in context-aware Web service systems.

2.1 Context-aware Systems

There is a wide range of context-aware systems. Many systems have been studied in various papers. First, we overview some generic context-aware systems, which are not based on Web services. We then examine systems which are partially or completely based on Web services technologies.

2.1.1 Non Web Service-based System

Various context-aware systems have been developed. While they are not Web services-based, some of them are distributed systems, and they have introduced and implemented various context-aware techniques.

RCSM (Yau & Karim, 2004) is a middleware supporting context sensitive applications based on an object model: context-sensitive applications are modeled as objects. RCSM supports situation awareness by providing a special language for specifying situation awareness requirements. Based on these requirements, application-specific object containers for runtime situation analysis will be generated. RCSM runtime system obtains context data from different sources and provides the data to object containers which conduct the situation analysis.

The JCAF (Java Context Awareness Framework) supports both the infrastructure and the programming framework for developing context-aware applications in Java (Bardram, 2005). Contextual information is handled by separate services to which clients can publish and from which they can retrieve contextual information (Bardram, 2005). JCAF is based on the peer-to-peer model but it does not support automatic discovery of peers or a super-peer. Also, the communication is based on Java RMI (Remote Method Invocation). JCAF, however, supports various sensors for monitoring locations and base classes for describing relevant entities used in context-aware applications.

The AWARENESS project[2] provides an infrastructure for developing context-aware and pro-active applications. It targets applications in mobile networks for the healthcare domain. In AWARENESS, applications will be context-aware and pro-active and are executed on top of a service infrastructure which

[2] The AWARENESS project (AWARENESS: Context AWARE Mobile NEtwork and ServiceS), http://www.freeband.nl/project.cfm?language=en&id=494. Last access: August 6, 2008.

provides generic components and manages context, security, and identity. The service infrastructure will operate on a network infrastructure which supports context-aware mobility.

QoSDREAM (Naguib, et al., 2001) is a middleware based on a component model that supports the development of context-aware multimedia applications. QoSDREAM gathers location-dependent information from various sensors and delivers that information to applications through an event messaging service.

The PACE middleware (Henricksen, et al., 2005) provides context and preference management together with a programming toolkit and tools for assisting context-aware applications to store, access, and utilize contextual information managed by the middleware. PACE supports context-aware applications to make decisions based on user preferences.

The GAIA project is a CORBA-based middleware supporting active space applications (Roman, et al., 2002). GAIA middleware provides a context service to support applications to retrieve as well as to publish contextual information.

CAMUS is an infrastructure for context-aware network-based intelligent robots (Kim, Cho, & Oh, 2005). It supports various types of context information, such as user, place and environment, and context reasoning. However, this system is not based on Web services and it works in a close environment.

SOCAM is a middleware for building context-aware services (Gua, Punga, & Zhang, 2008). It supports context modeling and reasoning based on OWL. However, its implementation is based on RMI.

Note that many context-aware systems in this category are network-based. Some rely on RMI and CORBA technologies. However, they do not rely on Web service technologies and are not designed to work on Web service-based environments. In addition, most of them have been surveyed in other papers (see Section 2.2). Therefore, we do not survey them in this paper, but refer to their techniques in case similar techniques are used in context-aware Web service systems.

2.1.2 Web Service-based Context-aware Systems

This section aims at providing an overview of context-aware systems built around Web services technologies. Not all parts of a Web service-based context-aware system might entirely be based on Web services. In this case, we will highlight only the context-aware Web service related parts.

The Akogrimo project[3] aims at supporting mobile users to access data, knowledge, and computational services on the Grid. With respect to context awareness, Akogrimo concentrates on context that is related to situations of mobile users, such as user presence and location, and environmental information (Osland, et al., 2006). The core context-related components in Akogrimo are a context manager which collects contextual information and delivers it to applications.

[3] The Akogrimo project (Akogrimo: Access to Knowledge through the Grid in a mobile World) http://www.mobilegrids.org/. Last access: August 5, 2008.

Chen and colleagues describe their CA-SOA (Context-aware Service Oriented Architecture) (Chen, et al., 2006). It uses ontologies to model context description linking service consumers and services. Based on the context model, CA-SOA proposes different components to support context-aware discovery and access Web services. The context-awareness aspect in CA-SOA is mainly for service discovery.

CoWSAMI is a middleware supporting context-awareness in pervasive environments (Athanasopoulos, et al., 2008). CoWSAMI provides a context manager to manage context sources. Context information is represented as relations and defined by using context collectors and context information can be queried.

The ESCAPE framework (Truong, et al., 2007) (Truong, et al., 2008) is a Web services-based context management system for teamwork and disaster management. ESCAPE services are designed for a front-end of mobile devices and the back-end of high end systems. The front-end part includes components support for context sensing and sharing that are based on Web services and are executed in an adhoc network of mobile devices. The back-end includes a Web service for storing and sharing context information among different front-ends.

Keild et al. present a generic framework to support the development of context-aware adaptable Web services (Keidl & Kemper, 2004). This framework separates clients/Web services from the context framework which supports clients and services. The transfer of context information is performed through SOAP message header. Context information can be explicitly and directly processed by clients or Web services or be automatically handled by the context framework.

Han and colleages present the Anyserver platform which supports context-awareness in mobile Web services (Han, Jia, Shen, & Yuen, 2008). The Anyserver platform utilizes various types of context information, such as device information, networks, and application type. However, it does not support direct context sharing and is not fully based on Web services.

The inContext project[4] (Truong, et al., 2008) provides various techniques for supporting context-awaress in emerging team collaboration. It is designed for Web services-based collaborative working environments. inContext provides techniques for modeling, storing, reasoning, and exchanging context information among Web services.

Matsumura and colleages present SiWS (Situated Web service) framework (Matsumura, Ishida, Murakami, & Fujishiro, 2006). SiWS framework aims at improving communication among Web services based on context-aware. The communication is optimized based on information about available protocols and operation usage.

Omnipresent is a context-aware LBS (location-based service) system based on Web services (De Almeida, et al., 2006). In Omnipresent, context is modeled based on OWL. Various services are developed to provide location-based information, such as maps and routes.

Prezerakos and colleagues uses UML to model context and Web services (Prezerakos, Tselikas, & Cortese, 2007). Based on UML models, a model-driven approach for composing context-aware Web services is proposed.

[4] http://www.in-context.eu. Last access: August 5, 2008.

Another type of context-aware systems in Web service-based environments is context-aware security systems. Zuidweg and colleagues discussed how context information can be used for ensuring security (Zuidweg, et al., 2003). An access control based on context is also presented in (Wang, Li, & Feng, 2008).

2.2 Existing Surveys on Context-aware Systems and Frameworks

Context-awareness is an attractive topic, therefore, it is not surprising that many surveys have been made for understanding context-aware systems.

Chen and colleagues presented a survey on context-awareness in mobile computing (Chen, et al., 2000). This survey does not address Web services issues.

In (Pokraev, 2003), context-aware techniques are analyzed. This paper discussed context-awaress in terms of sensing, modeling, monitoring and utilization of context information from a network perspective. It also presents a survey of some context-aware systems mentioned in Section 2.1.

Strang and Linnhoff-Popien presented a survey on context modeling (Strang & Linnhoff-Popien, 2004). This survey, however, focuses only on how to model context information.

Singh and Conway presented a survey of context-aware framework from architecture perspective (Singh & Conway, 2006). Their survey includes Context Toolkit, CorBrA, CMF and SOCAM. They consider software framework, context representation, and intelligence support.

Martin overviewed the relationship between context and Web services and discussed challenges in associating and handling contextual knowledge with/in Web services (Martin, 2006). This overview, however, does not provide a detailed analysis of context-aware Web services systems.

A recent survey on context-aware systems is presented in (Baldauf, Dustdar, & Rosenberg, 2007). In this survey, design principles, including architecture and context types, in context-aware systems are discussed. Based on these principles, this survey discussed some context-aware frameworks and systems, including many systems given in Section 2.1, such as Gaia, Context Toolkit, SOCAM and CorBrA.

Bolchini et al. presented a survey on data properties in existing context systems (Bolchini, Curino, Quintarelli, Schreiber, & Tanca, 2007). They examined various properties such as time, space, security, and privacy. Since this survey focused on data properties, it discussed mostly context representations and storages in existing systems. Similar to other surveys, many context-aware systems in Section 2.1 are included in this survey. However, their work does not analyze Web services aspects.

A survey of context adaptation in autonomic computing is presented in (Klein, Schmid, Leuxner, Sitou, & Spanfelner, 2008). This survey focuses on the evaluation of how existing frameworks fulfil requirements for autonomic computing, including adaptability, awareness, monitoring, dynamicity, autonomy, robustness, mobility and traceability. However, the surveyed frameworks are not Web service-based and the evaluation is only based on the architectural framework.

Because all the above-mentioned surveys do not focus on context-awareness techniques for Web services, they do not cover many Web service-related aspects that are discussed in this paper. Many Web service-based context-aware systems discussed in this paper have not been surveyed before. However, common terminologies and views on context-aware architecture principles from previous works are shared.

3 Context-aware Systems in Web services Environments

Before we examine context-aware techniques for Web service systems, we discuss our assumptions representing the environment for the survey.

As discussed and observed in many works, such as (Chaari, et al., 2004) (Keidl & Kemper, 2004) (Mikalsen, et al., 2006) (Henricksen, et al., 2005), a context-aware system have many components, such as context sensor, context storage, context reasoner, context consumer, to name just a few. These components are logically separated from applications that they support. In this paper, we separate between

- Context-aware services and applications: they utilize context information and supporting tools in order to act "smarter" (being context-aware).
- Supporting components for context-awareness: they support the applications and services by sensing and providing context information. Examples of supporting components are sensors, context storages, and context reasoning engines.

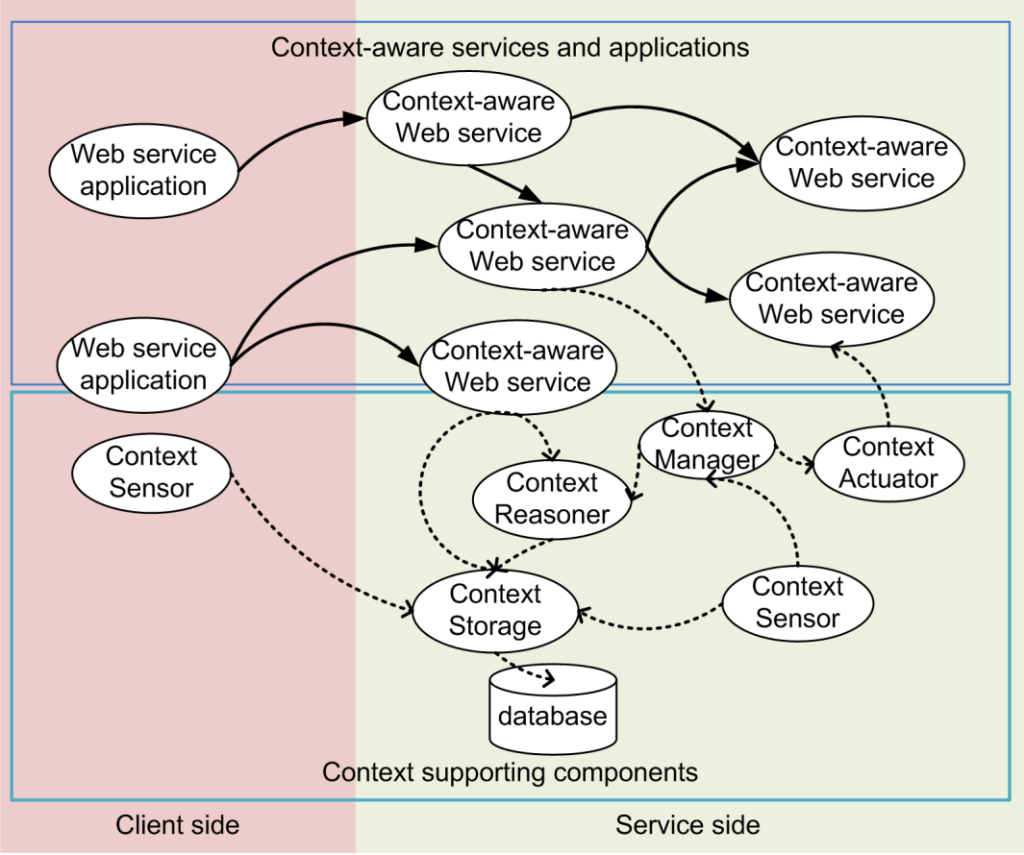

Figure 1: Basic components in a Web service-based context-aware system. Arrowed black lines indicate invocations between applications and Web services. Arrowed dash lines indicate invocations between Web services and supporting components.

Figure 1 describes basic components in a Web service-based context-aware system. These basis components are conceptual. It means that from the implementation point of view, many components can be implemented within a single entity, e.g., a sensor can be part of a context-aware service. Nevertheless,

by separating conceptual view of components from their implementation detail, we can analyze existing techniques in a better way.

In the view of applications/services and supporting tools, we conceptually divide basic components into the top part and the bottom part. In the top part, we have context-aware applications and services. All services are Web services and the communication among applications and context-aware services is based on Web services standard protocols, such as HTTP and SOAP. Context-aware applications may or may not be Web services but they interact with the context-aware services through Web services interfaces, typically described by WSDL[5]. In the bottom part, various supporting components for context-awareness are present. These components are either part of some Web services or Web services themselves. With this view, we focus on context-awareness techniques related to the interaction between two Web services. Depending on specific context-aware systems, the number of supporting components is different. However, the list of supporting components is generic enough to reflect the current Web service context-aware systems.

In the client-service view, we can distinguish between components with the client role and those with the service role. On the left-hand side, Web applications utilize Web services. Some supporting components can also provide context information associated with clients, such as sensors. On the right-hand side, various context-aware services offer different services to the clients. These services can interact with each other and will utilize various supporting components in order to be context-aware. When a service invokes another one, the service becomes a client of the invoked one.

Given that common structure of context-aware Web service systems, we need to understand techniques developed in these systems by answering the following questions:

- Context information and context representations: which techniques are used for modeling context information in Web services? They help us understand the context information exchanged among applications, context-aware services and supporting components.
- Context sensor techniques: how context information is measured and sensed? They help us understand how context information is measured and how sensors are implemented.
- Context storage techniques: how context information is stored and how the information is accessed from its storage? They are also related to context information and distribution techniques.
- Context distribution techniques: how context information is distributed and propagated to different relevant supporting components? How applications and services can retrieve context information?
- Security and privacy techniques: how context information is protected? Which authentication and authorization mechanisms are used for context information? Which techniques are used to ensure the privacy in connection to context sharing?
- Context adaptation techniques: how context information is actually used in Web services?

[5] http://www.w3.org/TR/wsdl. Last access: August 8, 2008.

Clearly, a context-aware system might not cover all above-mentioned aspects or part of it is not clear from its description. In this case, we will not be able to analyze these aspects. Besides the techniques, it is also important to analyze whether a system and its techniques are domain-specific or not, which parts can be reused, mobile devices are support or not, or context sharing and sensing services are performed in single or multiple domains.

Table I summarizes Web service-based context-aware systems discussed in Section 2.1.2. We examine these systems based on the following properties

- Application domain: whether a system is designed and developed for a specific application domain (e.g., healthcare) or for a generic domain.
- System type: whether a system is a middleware, a framework or an application.
- Multi-organization support: whether a system can be deployed and utilized in a multi-organizational environment.
- Mobility support: whether a system supports mobile devices and services.
- Level of Web service implementation: whether a system is fully or partially implemented in Web services

We use "+" to denote a feature that a system supports/meets and "o" to denote a feature that a system does not support/meet. When a feature is left blank, it means that it is not clear whether a feature is supported or not.

System	Application domain		System type		Multi-organization support	Mobility support	Level of Web service implementation	
	Domain specific	Generic	Framework/ Toolkits	Application			Full	Partially
Akogrimo		+	+		+	+		+
Anyserver	+		+	+		+		+
CA-SOA		+	+				+	
CoWSAMI		+	+			+	+	
ESCAPE	+	+	+	o	o	+	+	
Keidl & Kemper, 2004		+	+				+	

inContext		+	+	o	+	+	+	
Omnipresent	+			+		+	+	

Table I: Overview of context-aware systems based on Web services

As observed in Table I, not all systems are fully implemented as Web services. The main reason is that many systems are designed to support pervasive devices and services that require them to be able to deal diverse legacy systems, devices and protocols. Because Web service technologies are not suitable for all types of devices, these systems have to rely on other technologies. Another observation is that mobility is on the focus of many systems.

4 Techniques

4.1 Context Information and Context Representations

It is impossible to give an exhaustive list of types of context information as the type of context information is dependent on individual systems. One of the most popular types of context information is location information. Other types of common context information are device profiles, user preferences, user's activities and interactions, devices, and the network status. We survey types of context information in terms of common types of context information and historical support.

System	Type of information									Historical support
	Calendar	Location	Activity/ Task	Presence	Individual Profile/Preference	Team	Service/Application	Machine/ Device	Network	
Akogrimo		+		+	+		+	+	+	
Anyserver			o		+		+	+	+	
CA-SOA	+	+		+	+		+	+	+	+
CoSAr	+	+	+	+	+		+	+		
ESCAPE		+	+	+	+	+	+	+	+	+
inContext	+	+	+	+	+	+	+	+	o	+
Omnipresent		+	+		+					
SiWS							+		+	

Table II: Overview of types of context information supported

Table II summarizes types of context information supported in Web service context-aware systems. While some types of context information, such as Location, Presence, Individual Profile, Machine/Device and Network, have been widely used in many context-aware systems for a long time, other types of context

information, such as Service/Application, Activity/Task and Team, are also considered in Web service context-aware systems. These types of context play an important role when context is used to support adaptation in service/task selection (Prezerakos, Tselikas, & Cortese, 2007) and teamwork (Truong, et al., 2008).

To describe context information, different languages and models can be used. We distinguish between those languages/models for modeling and implementation. Given a system, a language can be used to model representation of the context information (modeling phase), while in the implementation of the system, concrete, instance context information might be described by the same or another representation. One example is to use UML to model the context information but the instance data is described in XML.

In existing context-aware systems, XML is already used widely for modeling and implementing context information. For example, The PPDL (pervasive profile description language) is an XML-based language that can be used to describe preferences of peers within situations peers operate (Ferscha, et al., 2006). In the CoSAr prototype (Dorn & Dustdar, 2007), a context sharing architecture is defined. Some XML schemas are defined for describing contextual information of mobile networks, for example, activity, device status and reachability. RDF[6] and OWL[7] are also used to describe context information. CC/PP[8] (Composite Capability/Preference Profile) can be used to describe contextual information of capabilities and user preferences. CC/PP is based on RDF. CC/PP, however, does not specify how contextual information can be stored. CC/PP is widely used in context-aware middleware and applications. For example, the CARMEN middleware (Bellavista, et al., 2003) uses CC/PP for describing metadata of user/device profiles while the Mercury middleware prototype (Solarski, et al., 2004) describes user, terminal, network, and service profiles using CC/PP. In (Goslar, et al., 2004), contextual information is represented using a topic map. COBRA-ONT is an ontology which can describe various entities in pervasive context-aware systems (Chen, et al., 2003); the ontology is based on OWL. SOCAM is used OWL to describe context information.

When analyzing context modeling, we observed that various works support the modeling of context together with the modeling of services, thus context and Web services are specified and coupled at the design time. This approach is different from other approaches in which Web services are modeled as normal, e.g., based on plain WSDL, and the services utilize context information using supporting components. Therefore, besides the two categories "Modeling" and "Implementation" representations, we also survey how context is modeled together with Web services modeling.

System	Modeling	Implementation	Modeling context with service

[6] http://www.w3.org/RDF/ (Resource Description Framework). Last access: July 30, 2008.

[7] http://www.w3.org/2004/OWL/. Last access: August 8, 2008.

[8] http://www.w3.org/Mobile/CCPP/. Last access July 30, 2008.

	XML	UML	OWL/Ontology	Tool-specific	XML	RDF	OWL	Tool-specific	Separated from WSDL	Integrated with WSDL
Akogrimo			+					+		
Anyserver	+				+					+
CA-SOA			+				+			+
ContextUML		+								+
CoWSAMI	+				+					+
ESCAPE		+			+					
Keidl & Kemper, 2004	+				+					
inContext		+	+		+	+				
Omnipresent			+				+			
SiWS	+				+					

Table III: Overview of context representation techniques

Table III summarizes techniques used to describe context information. XML, RDF, and OWL-based representations are widely used because they are considered open and interoperable, while others are not. In particular, RDF and OWL approaches facilitate the reuse of common vucabularies. From a software engineering point of view, UML approaches offer advantage over other approaches because they can be seamlessly integrated in MDE (Model Driven Engineering) (Schmidt, 2006). While XML is considered open and interoperable, yet there is no standard way of sharing XML based vocabularies in current approaches.

With respect to the modeling of context together with services, various systems exist. Soukkarieh and Sedes presented in their short paper about context infomration integrated into Web services (Soukkarieh & Sedes, 2007). Sheng and colleages have presented ContextUML which can be used to model context and context-awareness for Web services (Sheng & Benatallah, 2005). ContextUML specifies classes for context types, sources, and context services as well as models context-aware objects within a service (services, operations, messages) and the binding of context-aware mechanism. Unlike other representations which focus on context information only and which can be used separated from the service modeling and development, ContextUML is strongly linked to service modeling that can be considered an integrated part of the service modeling. Based on an extension of ContextUML, Prezerakos and colleages have presented a model-driven approach for composing context-aware Web services (Prezerakos, et al., 2007). CoWSAMI also supports the specification of context during the service modeling phase. In our view, this approach will be increasingly adopted. In particular, it can be combined with MDE to enginering context-aware Web services.

4.2 Context Sensor Techniques

Contextual information is normally measured by hardware- or software-based sensors, such as GPS and monitoring programs, or provided by users. Typically, sensors rely on low level communication protocols to send the collected context information or they are tightly coupled within their context-aware systems. In Web service environments, we also focus on sensors which can be sensed and provided context information through Web services.

To analyze techniques for sensors, we examine mode, sensing techniques, sensor interface and how sensor data is retrieved and published. Table IV summarizes techniques used in sensors.

System	Mode		Sensing techniques		Sensor interface		Data retrieval and publishing			Quality of context support
	Automatic	Manual	Instrumentation	Polling	Web service	Specific	Query	Subscription	Push Only	
Akogrimo	+			+		+		+	+	
Anyserver	+			+		+				
CA-SOA	+	+	+	+		+				
ESCAPE	+	+	+		+		+	+		+
inContext	+			+	+	+			+	o
SiWS	+		+							

Table IV: Overview of techniques used in sensors

What is new in the development of sensors in Web service-based context aware systems is not how the sensors measure context data, but how they provide the context data. Since sensing techniques are well developed, existing sensors utilize these techniques through instrumentation or polling mechanisms, and extend their capability by acquiring context information from existing systems, e.g., using polling technique to obtain presence information from a service. However, the way in which sensors provides context information changes. There are more sensors that provide Web services interfaces, even in mobile devices, for other clients/services to access context information, realizing the vision of "sensor as a service". For example, in ESCAPE, sensors can be Web services that allow the subscription and publishing of context information.

In existing systems, sensors typically produce raw context information. However, the quality of context (Buchholz, Kpper, & Schiffers, 2003) which is useful for determining the quality associated with context information has not been well addressed. Of surveyed systems, only ESCAPE supports quality of context as mentioned in a recent paper (Manzoor, Truong, & Dustdar, 2008).

Another observation is that since context-aware systems in Web service-based environments have to deal with pervasive devices and services, sensors in these systems have to support various protocols and interact with third party systems. As a result, the interface between sensors and other supporting components (such as sensor managers) are diverse. For example, in Akogrimo sensors communicate with context manager using different protocols, including SIP (Session Initiation Protocol)[9] and Web services.

4.3 Context Storage Techniques

Relational databases are widely used to store context information in non Web services-based context-aware systems, such as in (Naguib, et al., 2001), (Mantoro, et al., 2003) and (Henricksen, et al., 2005). From a technology point of view, a context storage without a Web services interface can be wrapped into a newly-created Web service-based storage. From an application point of view, storages based on non Web service technologies are well understood and they are not suitable for Web service-based environments. Therefore, we will focus only on context storage techniques built around Web service technologies.

We analyze context storage in terms of storage databases and access interface. Table V summarized the techniques used for context storages. Note that a context storage might not be provided as a separate service but part of another service. For example, Akogrimo does not provide context storage as a separate service. Instead, context storage is part of a context management. While storage databases are different, most systems provide Web service interface for accessing context information. In cases of XML or ontology-based systems, although from the application point of view, context information is retrieved in XML or ontology, the back-end databases in many of those systems are actually relational databases. It is because the native XML databases are not mature as relational ones which are equipped with strong many libraries supporting the serialization of XML to relational tables.

System	Storage model		Storage database				Access interface		Request specification			
	Centralized	Distributed	Relational	XML	RDF/OWL	Others	Web service	Others	SQL	XPath/XQuery	SPARQL	Specific
Akogrimo	+		+					+	+			
CoWSAMI		+	+				+		+			
ESCAPE		+		+			+			+		
inContext	+	+			+		+				+	

Table V: Overview of context storages

[9] http://www.ietf.org/html.charters/sip-charter.html. Last access: August 6, 2008.

4.4 Context Distribution Techniques

Context information can be distributed in different ways. We consider distribution techniques in three aspects: using direct transport protocols, using overlay network protocols, and supporting access mechanisms.

In using direct transport protocol, context information will be transferred between two parties based on SOAP messages. OASIS proposed a Web services context specification (WS-Context) that describes a mechanism and service structure for sharing and passing context information between services and clients (Little, Newcomer, & Pavlik, 2004). The basic idea is to define context in XML and use the SOAP message header to transfer the context information. Techniques to manipulate SOAP header extension are widely used in practice. Various systems utilize SOAP headers to transfer context information. We obversed two different techniques in this aspect. The first type is to insert context information directly into SOAP message headers and the second type is to use the header to transfer only the reference. Keild and colleagues present a framework for context-aware Web services that inserts context information into the SOAP message header (Keidl & Kemper, 2004). In the inContext system, only the reference to the context source is transferred using SOAP message headers and context information can be retrieved from the source specified by the reference.

Most Web service-based context-aware systems rely on distribution techniques built atop an overlay network of services. The protocols are used either based on Web services, such as WS-Notification[10], or specific protocols. Furthermore, the distribution model can be centralized as well as P2P. For example, the inContext distribution technique supports both centralized and distributed models for context distribution in which context information can be accessed from a context store or different services which provide context information. In the ESCAPE framework, context distribution is based on a specific protocol as well as Web services. On the one hand, with sensors as Web services, context information can be retrieved through Web services interface. On the other hand, context information can be retrieved from the context management service by query or subscription. In many systems, the subscription is performed based on a specific implementation of Web service call-backs.

System	Overlay network distribution		Direct transport distribution		Access mechanism			
	Centralized	P2P	SOAP extension	Web proxy	Query through Web service	Subscription through Web services Notification/ Pubscribe	Subscription through WS call back	Subscription through specific protocol
Akogrimo	+					+	+	
CoWSAMI	+				+			
ESCAPE	+	+			+		+	+

[10] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn. Last access: August 7, 2008.

inContext	+		+		+			
Keidl & Kemper, 2004			+					
SiWS				+				

Table VI: Overview of context distribution techniques

Table VI summarized context distribution techniques. Although various works are based on SOAP message header and proxy, which offer simple and direct context distribution, security and privacy issues have not been well researched for this approach. On the other hand, P2P-based approach seems suitable for Web service-based context-aware systems, especially those span multiple organizations. However, current P2P-based distribution techniques built atop Web service technologies are not popular. Thus, still many systems rely on centralized distribution techniques.

4.5 Context Reasoning Techniques

Context information can be provided by sensors in a form that can be used immediately. However, reasoning techniques can be also used to infer new types of context information. As described in Section 4.1, various models have been used to represent context information. Obviously, the representation used in a system has a strong impact on the context reasoning techniques implemented in the system. When the context information is described by OWL and ontologies, typically reasoning techniques will be based on semantic approach, such as SPARQL[11].

System	Reasoning capability support	Semantic-based reasoning		Specific reasoning
		SPARQL	Others	
Akogrimo	o			
CA-SOA	+		+	
CoWSAMI	o			
ESCAPE	o			
inContext	+	+		

Table VII: Overview of context reasoning techniques

Table VII summarizes reasoning techniques support in existing systems. Overall, context reasoning is not well supported and only few systems are based on semantic-reasoning. It is due to the fact that still many context systems rely on XML-based context information which is not well backed up with reasoning

[11] http://www.w3.org/TR/rdf-sparql-query/. Last access: August 8, 2008.

techniques. On the other hand, OWL/ontologies approaches have strong reasoning support but they are heavy and difficult to be implemented and integrated into context-aware systems.

4.6 Security and Privacy Techniques

Context information might be sensitive. Security and privacy techniques for context-aware systems are required in various places, such as sensing, distributing and accessing context information. We analyze security and privacy techniques in terms of encryption, authentication, authorization, and privacy specification. Note that we consider security and privacy techniques different from the utilization of context information for security and privacy.

System	Security			Privacy specification
	Encryption	Authentication	Authorization	
Akogrimo		+	+	+
CA-SOA				+
CoWSAMI	o	o	o	o
ESCAPE	o	o	o	o
Keidl & Kemper, 2004			+	
inContext	o	+	o	o

Table VIII: Overview of security and privacy techniques

Table VIII summarized security and privacy techniques. While security and privacy issues are a highly related topic in context-aware systems, these issues have not been well addressed in context-aware Web services. In the inContext project, security and privacy rules have not been applied for context information. Akogrimo supports privacy, however, only limited. For example, privacy policy may be used to indicate which consumer may require which types of context information (Solsvik, 2006).

4.7 Context Adaptation Techniques

Adaptation based context information is typically application-specific. Many context-aware middleware allow the developer to specify actions that should be performed, given particular contexts. In most cases, the middleware might just support the management and exchange of contextual information.

Although the reasons for performing context adaptation are diverse, we observe the following main purposes:

- Service selection and task adaptation: context information is mostly used to select the most suitable service and task to perform actions, given a situation. One example of this purpose is how to use context for service provisioning (Maamar, AlKhatib, & Mostefaoui, 2004).

- Security and privacy control: context information is used to support adaptive control in security and privacy management. Examples of this purpose are how context information can be used for security and privacy protection (Zuidweg, et al., 2003) and access control based on context (Wang, Li, & Feng, 2008).

- Communication adaptation: context information is used to select communication protocols and optimize the communication. One example of this purpose is to utilize context for optimizing communication in SiWS.

- Content adaptation: context information is used to adapt content resulting from a request and to return the content in a form suitable to the context of the requester. One example is to adapt content in mobile Web services (Han, Jia, Shen, & Yuen, 2008).

Given a purpose, adaptation can be specified at runtime or modeling time. The adaptation in a service can also be supported by the middleware and framework or has to be implemented by the service itself by using supporting components for context-awareness.

System	Adaptation purpose					Adaptation specification		Adaptation layer		
	Content adaptation	Communication adaptation	Service and task selection	Information protection	Others	Modeling	Runtime	Unspecified	Middleware	Application/Service
Akogrimo	+		+							+
Anyserver	+	o	o	o					+	
ESCAPE	o	o	+	o						+
inContext	o	o	+	o			+		+	
(Wang, Li, & Feng, 2008)	o	o	o	+					+	
(Zuidweg, et al., 2003)	o	o	o	+					+	
SiWS		+		o		+			+	

Table IX: Overview of adaptation techniques

Table IX summarizes adaptation techniques. Overall, most systems support adaptation at middleware layer. In most cases, Web services rely on supporting components for acquiring context information and performing the adaptation. In this case, the Web services can decide whether to use context information or not.

5 Discussion

Sharing context vocabularies: There are domain independent and domain-specific context information concepts. Various representations cover many common concepts modeling context information, such as individual profile, device, machine, network, activity, and service. Although some specifications have been introduced, such as FOAF[12], inContext model[13], and SIOC[14], how to share and integrate common context concepts for context-aware Web services systems still requires a major research and standardized effort. This sharing and integration aspect is important as context-aware Web services systems need to utilize open, interoperable context information.

Distributed context management techniques need to be developed: The context information in Web services-based system is naturally available in different Web services, potentially belonging to different organizations. Therefore, a centralized context management technique does not work well. In particular, context information is shared based on various policies, imposed by different organizations. Currently, most works rely on dedicated centralized context storages or a service registry for locating context storages. In some senses, one can say that existing work support basic P2P model of context storages but in fact most are based on consumer-registry-publisher model rather than an adaptive P2P context registry system with distributed query and subscription support. A fully distributed P2P-based context management and sharing system would be beneficial. Koskela and colleagues have discussed some models which can be used for sharing context information in Web 2.0 (Koskela, et al., 2007).

The use of ontology for modeling context information increases, yet engineering challenges remain: Context information is mostly modeled in XML, UML and ontologies. We observed the trend to use ontologies for modeling context information. In fact, various frameworks support ontology-based context information. The use of ontologies allows the inclusion of existing vocabularies. For example, in the inContext project, context information has been reused from various common ontologies. However, technical issues related to context update, security and access actually prevent the widely used of ontologies context in Web services-based environments. It is still an open question of how to support rich-semantic context information and adaptation in mobile devices.

Context reasoning and quality of context are not well developed: Support of context reasoning is limited. Consider a rich set of context information in Web service-based environment, context reasoning techniques should be the next research focus. Furthermore, when context information is exchanged in the Web environment, it is naturally to question about the quality of context information, because context sources are diverse. However, until now techniques for evaluating quality of context information in Web services systems are still under developed. Only a few work support quality of context in context-aware Web service systems, such as (Manzoor, Truong, & Dustdar, 2008).

---

[12] http://www.foaf-project.org/. Last access: September 19, 2008.

[13] http://www.in-context.eu/uploads/files/20070627D2.2v1.0Context20model20design20and20prototype20implementation.pdf. Last access: September 19, 2008.

[14] Semantically-Interlinked Online Communities, http://sioc-project.org/. Last access: September 19, 2008.

Security and privacy issues have not been well addressed: Most Web service-based context-aware systems lack the security and privacy enforcement. This enforcement is important when the context information is stored in centralized data store. While security and privacy techniques have not been adequately developed for context-aware systems, there is a research trend to exploit the context for security monitoring and enforcement.

Modeling service description with context information is not enough: Many works have developed context-aware Web services by enriching service descriptions with context information at the design time. This approach helps to improve the service discovery and composition and simplifies the engineering process for context-aware Web services, e.g., based on MDE. However, it does not address the issue of context sensing, management and distribution. This modeling approach should be enhanced to allow also the modeling of connections between service interfaces and other supporting components for context-awareness.

In this survey, context representations, sensing and storages techniques for Web services are mature and quite similar to that of context-aware systems without Web services. However, context distribution techniques, quality of context, security and privacy issues, and engineering methodology for context-aware Web service systems are still under-researched.

6 Conclusion

"The Future Internet" will consist of pervasive devices and complex Web services from different organizations. Being context-aware will help services to adapt to changes. Considering the important role of Web services in future context-aware systems, this paper has analyzed existing Web service-based context aware systems. Our study mainly focused on how context-aware techniques are applied in Web services systems. While there are many existing context-aware systems, few are based on Web services. Various techniques exist, yet few are applicable for Web service environments.

It is hard to find a truly Web service-based context-aware system that is interoperable and secure, and operates on multi-organizational environments. It is understandable because Web services technology is rather new and there are various challenges that prevent the widely-implementation of context-awareness techniques in Web service-based environments. To further support the development of context-aware Web services, we need to focus on distributed context management, security and privacy techniques as well as interoperable representations for context information.

Acknowledgments

The work mentioned in this paper is partially funded by the EU FP6 WORKPAD, EU FP6 inContext and EU FP7 COIN projects.

References

Abowd, G. D., Ebling, M., Gellersen, H.-W., Hunt, G., & Lei, H. (2002, Jul-Sept). Guest Editors' Introduction: Context-Aware Computing. *IEEE Pervasive Computing , 1* (3), pp. 22-23.

Athanasopoulos, D., Zarras, A., Issarny, V., Pitoura, E., & Vassiliadis, P. (2008). CoWSAMI: Interface-aware context gathering in ambient intelligence environments. *Pervasive Mob. Comput. , 4* (3), 360-389.

Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing , 2* (4), 263-277.

Bardram, J. (2005). The java context awareness framework (jcaf) - a service infrastructure and programming framework for context-aware applications. *Pervasive* (pp. 98--115). Springer.

Bellavista, P., Corradi, A., Montanari, R., & Stefanelli, C. (2003). Context-Aware Middleware for Resource Management in the Wireless Internet. *IEEE Trans. Software Eng. , 29* (12), 1086-1099.

Bolchini, C., Curino, C., Quintarelli, E., Schreiber, F. A., & Tanca, L. (2007). A data-oriented survey of context models. *SIGMOD Record , 36* (4), pp. 19-26.

Buchholz, T., Kpper, A., & Schiffers, M. (2003). Quality of context: What it is and why we need it. *Proceedings of the 10th International Workshop of the HP OpenView.* Hewlet-Packard OpenView University Association.

Chaari, T., Laforest, F., & Celantano, A. (2004). *Design of context-aware applications based on web services.* Technical report, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard .

Chen, G., & Kotz, D. (2000). *A survey of context-aware mobile computing research.* Dartmouth College. http://www.cs.dartmouth.edu/reports/TR2000-381.pdf

Chen, H., Finin, T., & Joshi, A. (2003). An ontology for context-aware pervasive computing environments. *Knowl. Eng. Rev. , 18* (3), 197--207.

Chen, I. Y., Yang, S. J., & Zhang, J. (2006). Ubiquitous Provision of Context Aware Web Services. *Proceedings of the IEEE international Conference on Services Computing* (pp. 60-68). Washington: IEEE Computer Society.

De Almeida, D. R., De Souza Baptista, C., Da Silva, E. R., Campelo, C. E., De Figueiredo, H. F., & Lacerda, Y. A. (2006). A Context-Aware System Based on Service-Oriented Architecture. *Proceedings of the 20th international Conference on Advanced information Networking and Applications - Volume 1 (Aina'06) - Volume 01* (pp. 205-210). IEEE Computer Society.

Dey, A. K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing , 5* (1), 4-7.

Dorn, C., & Dustdar, S. (2007). Sharing hierarchical context for mobile web services. *Distributed and Parallel Databases , 21* (1), 85-111.

Ferscha, A., Hechinger, M., Riener, A., Schmitzberger, H., Franz, M., Rocha, M. d., et al. (2006). Context-Aware Profiles. *2006 International Conference on Autonomic and Autonomous Systems (ICAS 2006).* IEEE Computer Society 2006.

Goslar, K., & Schill, A. (2004). Modeling Contextual Information Using Active Data Structures. *EDBT Workshops* (pp. 325-334). Springer-Verlag.

Gua, T., Punga, H. K., & Zhang, D. Q. (2008). A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications , 28* (1), 1-18.

Han, B., Jia, W., Shen, J., & Yuen, M.-C. (2008). Context-Awareness in Mobile Web Services. *Parallel and Distributed Processing and Applications* (pp. 519-528). Springer-Verlag.

Henricksen, K., Indulska, J., McFadden, T., & Balasubramaniam, S. (2005). Middleware for Distributed Context-Aware Systems. *OTM Confederated International Conferences* (pp. 846-863). Springer-Verlag.

Keidl, M., & Kemper, A. (2004). A Framework for Context-Aware Adaptable Web Services. *Advances in Database Technology - EDBT 2004*, (pp. 826-829).

Kim, H., Cho, Y.-J., & Oh, S.-R. (2005). CAMUS: a middleware supporting context-aware services for network-based robots. *IEEE Workshop on Advanced Robotics and its Social Impacts, 2005.*, (pp. 237-242).

Kjær, K. E. (2007). A survey of context-aware middleware. *Proceedings of the 25th conference on IASTED International Multi-Conference: Software Engineering* (pp. 148-155). ACTA Press.

Klein, C., Schmid, R., Leuxner, C., Sitou, W., & Spanfelner, B. (2008). A Survey of Context Adaptation in Autonomic Computing. *Fourth International Conference on Autonomic and Autonomous Systems, 2008. ICAS 2008* (pp. 106-111). IEEE Computer Society.

Koskela, T., Kostamo, N., Kassinen, O., Ohtonen, J., & Ylianttila, M. (2007). Towards Context-Aware Mobile Web 2.0 Service Architecture. *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies 2007 (UBICOMM 2007)* (pp. 41-48). IEEE Computer Society.

Little, M., Newcomer, E., & Pavlik, G. (2004). *Web Services Context Specification (WS-Context).* http://docs.oasis-open.org/ws-caf/ws-context/v1.0/wsctx.html

Maamar, Z., AlKhatib, G., & Mostefaoui, S. (2004). Context-based personalization of Web services composition and provisioning. *Proceedings. the 30th of Euromicro Conference* (pp. 396 - 403). IEEE Computer Society.

Manes, A. T. (2001). Enabling Open, Interoperable, and Smart Web Services - The Need for Shared Context. Sun Microsystems, Inc. http://www.w3.org/2001/03/WSWS-popa/paper29

Mantoro, T., & Johnson, C. (2003). Location history in a low-cost context awareness environment. *ACSW Frontiers '03: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003* (pp. 153--158). Australian Computer Society, Inc.

Manzoor, A., Truong, H.-L., & Dustdar, S. (2008). On the Evaluation of Quality of Context. *Proceedings of the 3rd IEEE European Conference on Smart Sensing and Context (EuroSSC).* Springer-Verlag.

Martin, D. (2006). Putting Web Services in Context. *Electr. Notes Theor. Comput. Sci. , 146* (1), 3-16.

Matsumura, k., Ishida, T., Murakami, Y., & Fujishiro, Y. (2006). Situated Web Service: Context-Aware Approach to High-Speed Web Service Communication. *International Conference on Web Services 2006 (ICWS'06)* (pp. 673 - 680). IEEE Computer Society.

Mikalsen, M., Floch, J., Paspallis, N., Papadopoulos, G. A., & Ruiz, P. A. (2006). Putting Context in Context: The Role and Design of Context Management in a Mobility and Adaptation Enabling

Middleware. *7th International Conference on Mobile Data Management (MDM 2006).* Nara, Japan: IEEE Computer Society.

Miller, N., Judd, G., Hengartner, U., Gandon, F., Steenkiste, P., Meng, I.-H., et al. (2004). Context-aware computing using a shared contextual information service. *Pervasive 2004 Hot Spots.*

Moran, T. P., & Dourish, P. (Eds.). (2001). Context-Aware Computing. *Human Computer Interaction (HCI) Journal (2-4)* .

Naguib, H., Coulouris, G., & Mitchell, S. (2001). Middleware Support for Context-Aware Multimedia Applications. *the IFIP TC6 / WG6.1 Third International Working Conference on New Developments in Distributed Applications and Interoperable Systems*, (pp. 9-22).

Nihei, K. (2004). Context sharing platform. *NEC Journal of Advanced Technology , 1* (3), 200-2004.

Osland, P.-O., Viken, B., Solsvik, F., Nygreen, G., Wedvik, J., & Myklbust, S. E. (2006). Enabling context-aware applications. *Proceedings of ICIN2006: Convergence in Services, Media and Networks.*

Pokraev, S. (2003, November). Context-aware services - State of the art. Telematica Instituut, Ericsson, CTIT.https://doc.freeband.nl/dsweb/Get/Document-27859/Context-aware%20services-sota,%20v3.0,%20final.pdf

Prezerakos, G. N., Tselikas, N. D., & Cortese, G. (2007). Model-driven Composition of Context-aware Web Services Using ContextUML and Aspects. *2007 IEEE International Conference on Web Services (ICWS 2007)* (pp. 320-329). IEEE Computer Society.

Roman, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R., & Nahrstedt, K. (2002, Oct-Dec). Gaia: A middleware infrastructure to enable active spaces. *IEEE Pervasive Computing* , pp. 74--83.

Schmidt, D. C. (2006, February). Model-Driven Engineering. *IEEE Computer , 32* (2), pp. 25-31.

Sheng, Q. Z., & Benatallah, B. (2005). ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services Development. *International Conference on Mobile Business (ICMB'05)* (pp. 206-212). IEEE Computer Society.

Singh, A., & Conway, M. (2006, 7). Survey of Context aware Frameworks - Analysis and Criticism. http://its.unc.edu/teap/tap/core/caf_review.pdf

Solarski, M., Strick, L., Motonaga, K., Noda, C., & Kellerer, W. (2004). Flexible middleware support for future mobile services and their context-aware adaptation. *INTELLCOMM* (pp. 281--292). Springer.

Solsvik, F. (2006). *D4.2.3: Final Integrated Services Design and Implementation Report.* Akogrimo deliverable.http://www.akogrimo.org/modules.php?name=UpDownload&req=viewdownloaddetails&lid=110

Soukkarieh, B., & Sedes, F. (2007). Integrating a Context Model in Web Services. *IEEE International Conference on Web Services (ICWS 2007)* (pp. 1195-1196). IEEE Computer Society.

Strang, T., & Linnhoff-Popien, C. (2004). A Context Modeling Survey. *Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing.*

Truong, H. L., Juszczyk, L., Manzoor, A., & Dustdar, S. (2007). ESCAPE - An Adaptive Framework for Managing and Providing Context Information in Emergency Situations. *Smart Sensing and Context, Second European Conference, EuroSSC 2007* (pp. 207-222). Springer-Velag.

Truong, H.-L., Dustdar, S., Baggio, D., Corlosquet, S., Dorn, C., Giuliani, G., Gombotz, R., et al. (2008). inContext: a Pervasive and Collaborative Working Environment for Emerging Team Forms. *The 2008 International Symposium on Applications and the Internet (SAINT2008).* Turku, Finland: IEEE Computer Society.

Truong, H.-L., Juszczyk, L., Bashir, S., Manzoor, A., & Dustdar, S. (2008). Vimoware - a Toolkit for Mobile Web Services and Collaborative Computing. *Special session on Software Architecture for Pervasive Systems, the 34th EUROMICRO Conference on Software Engineering and Advanced Applications,*. IEEE Computer Society.

Voida, S., Mynatt, E., MacIntyre, B., & Corso, G. (2002). Integrating virtual and physical context to support knowledge. *IEEE Pervasive Computing , 1* (3), pp. 73--79.

Wang, C.-D., Li, T., & Feng, L.-C. (2008). Context-Aware Environment-Role-Based Access Control Model for Web Services. *2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008)* (pp. 288-293). IEEE Computer Society.

Yan, H., & Selker, T. (2000). Context-aware office assistant. *IUI '00: Proceedings of the 5th international conference on* (pp. 276--279). ACM Press.

Yau, S., & Karim, F. (2004). A context-sensitive middleware for dynamic integration of mobile. *J. Parallel Distrib. Comput. , 64* (2), 301--317.

Zuidweg, M., Goncalves Filho, J., & van Sinderen, M. (2003). Using P3P in a web services-based context-aware application platform. *EUNICE 2003 9th Open European Summer School and IFIP WG6.3 Workshop on Next Generation Networks*, (pp. 238-243).