Context modeling and the Web of Things

Deliverable 4.1 (version 1)

Mehdi Terdjimi, Lionel Médini and Michael Mrissa
27 May 2015

Project ASAWoO
Adaptive Supervision of Avatar / Object Links for the Web of Objects
Grant Agreement: ANR-13-INFR-0012-04

Abstract

The Web of Things (WoT) aims at connecting things to applications using web technologies. Built on top of the Internet of Thing's basic network connectivity, the WoT brings new challenges as it has to deal with heterogeneity and must adapt to changes in the environment wrt. security, privacy and quality of service. Thus, multi-level context models have to be designed. To reach those requirements, we define in this deliverable the architecture of a WoT application and review different context modeling approaches in various fields related to the WoT. According to our review, we analyze and identify different pieces of context information to characterize context on each layer composing the architecture of a WoT application.

Contents

1 Introduction

The Internet of Things (IoT) applies to various domains such as homes, entreprises, industry and the city. It consists in networks of sensors which share information using various protocols. However, the number of connected devices increases as the IoT becomes popular. They spread in different places, at different times, for different purposes. Various middleware solutions exist to exploit the information gathered from sensors and to deal with the heterogeneity, interoperability, security as well as context-awareness. A review of the most representative context-aware IoT applications is proposed in [33]. They use various context-models in the IoT for different purposes. Some of them rely on Web technologies and standards to break through silos caused by proprietary devices and frameworks. Built on top of the IoT, this approach is known as the Web Of Things (WoT): "things" are now connected to the Web using interaction models over the IoT's basic network connectivity [22]. The WoT aims at delivering services getting more and more smart and adapative wrt. the context. Thus, multi-level context models have to be designed.
In this deliverable, we describe the architecture of WoT applications and propose different fields for our study in Section 2. In Section 3, we review general definitions of context and survey context models for each field. We synthetize and discuss these works in Section 4. Based on this synthesis, we identify which context dimensions could be used in a WoT application wrt. its architecture and propose a context model in Section 5. We conclude and give perspectives in Section 6.

2 Web of Things architecture

In his PhD thesis, Guinard proposes a WoT application architecture [21] based on 4 layers. The first layer is Accessiblity, which provides a consistent access to physical objects, also called "smart things", and their functionalities, based on the principles of REST. The Findability layer allows users to find the right service for their application, by enabling indexation of smart things by search engines, lookup and discovery infrastructures. The Sharing layer address the concern of sharing smart things in social networks or RSS feeds (RSS, Twitter). The Composition layer enables composition of smart things functionalities by non-specialists through physical mashups [23]. Each layer must provide semantic annotations, ensuring interoperability.
In WoT applications, physical objects can be composed and accessed in various ways. One can consider a physical object as a Resource (in the meaning of the Web[1], not to be confused with the physical object's computing resources) and represent its characteristics (e.g. name, capabilities...) through a Web interface. We call this representation the "avatar" [29], which is the logical extension of the physical object. Hence, the main question is the ability for a physical object to host it. Different levels of physical objects have to be considered, from the basic object with no sensor nor computational resource to the object capable of embedding his virtual representation, without the requirement to be connected. The following classification of physical objects is based on Mrissa et al.[29] :

1. Objects having no sensor and no actuator. They cannot gather information, cannot compute but can indirectly communicate via the information they can carry through RFID tags or QR codes physically attached to them.

2. Objects having no sensor but still able to retrieve knowledge from Web services.

3. Objects having sensors, actuators but constrained by one or more of their computing resources. They can gather information by themselves but have not the ability to completely embed their avatar .

4. Objects incorporating knowledge, having computing abilities, being capable of embedding their avatar and sometimes embedding other's objects avatar, rendering services for constrained objects.

In our work as part of the ANR ASAWoO project[2], physical objects are exposed as Web resources. In some cases, all the features an object offer might not be available depending on its situation. Some features

[1] A Web Resource offers interaction possibilities and is accessible via an Uniform Resource Identifier (URI) and HTTP verbs.
[2] http://liris.cnrs.fr/asawoo/

might be disabled for technical reasons such as physical constraints, or due to security, privacy, or other policies. The decision to make features available or not depends on the context. To enable the adaptation of connected devices to the changes affecting them, appropriate context modeling is required.
If we consider Guinard's WoT architecture [21], ASAWoO's principles and physical object's classification from Mrissa et al. [29], the relevant elements we consider for our study are the physical objects and their surrounding physical world, their communication between each other and with the application, the application itself, the Web regarding the technologies and standards used for some applications, and how a society of entities[3] sets up within the WoT. As a consequence, we propose to survey works on context modeling in these different fields or domains: **Physical World**, **Network**, **Social computing**, **Web** and **Applications**.

3 Context in various fields

3.1 Introduction

In the WoT, the context highly impacts the application's behavior. It can be modeled from various pieces of information. In such heterogeneous environment, the boundaries of context must be properly defined. These boundaries depends on the field or domain of the application. Consequently, a large variety of context models have been designed. The following survey explores the different fields identified in Section 2 and the major works on each of them. Our choices are based on how the approach is representative for the challenges addressed in the WoT. In order to limit the scope of this deliverable, we do not study the operationalization part of the context models.

3.2 General definitions of context

A first definition of context was given in 1994 by Schilit and Theimer [38]. They define context-aware computing as the ability of a mobile user's applications to discover and react to changes in the environment they are situated in. They define context as location, identities of surrounding entities (either people or physical objects), and changes to those entities. For Pascoe [32], context is subjective and specific to each particular entity that perceives it. He considers context as the subset of physical and conceptual states of interest to a particular entity. The initial definition of context proposed by Dey et al. [15] focuses on the user, consisting in the user's physical, social, emotional or informational state. This approach actually lacks generalization. Indeed, Abowd et al. [1] find those definitions too specific. They believe that context is all about the whole situation relevant to an application and its set of users. For instance, environment as a reference to the context may be irrelevant depending on the point of view of the entity. As a consequence, Dey [16] proposes a more generic definition of context:

Definition 1 (Dey, 1999) *Context is any information that can be used to characterize 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.*

A situation is an instance of each contextual parameter. Any change in these parameters defines a new situation [12].

3.3 Using context for the physical world

The information about the physical world is one of the main elements that define context: the entities and their characteristics, their activities and their surroundings. In PARCTAB [37], Schilit et al. identify three context information: Date and Time, Location and Co-location (i.e. what is nearby). The Conference Assistant from Dey et al. [17] uses three categories of context determined by a level of privacy, which are Public context (location, time, presentation's keywords, people presenting, media used), User context (time they entered/exited a room, location, question asked, element/slide pointed out) and Other users

[3] In our study, we consider "entities" as either a user or a physical object.

context (presence and question asked in presentations). Schmidt proposes in his PhD thesis [39] a three-dimensional context: Environment (physical and social), Self (device state, physiological, cognitive) and Activity (behavior, task).
There are discrepancies concerning these dimensions, however. Zimmerman et al. [47] object to Schmidt's three-dimensional context: "Self dimension introduces a relation of the context to one specific entity [...] which lacks an approach of how his model would capture a setting comprised of many interacting entities". In that reflection, they identify Relations and isolate the notion of Individuality from the Self dimension. They also explicitly consider Location and Time instead of Schimdt's implicit Environment [39]. For Dey and Abowd [1], considering Environment as a dimension for context is redundant as it is already a synonym for Context. They propose to replace Environment by Activity. Context is, in their view, the answer to four questions being *Where*, *Who*, *When* and *What*, answered respectively by Location, Identity, Time, and Activity. Additionaly, they exclude Schilit's *nearby* consideration [36] as it can be obtained by combining Location and Identity.
These works describe the context as the location, the surrounding entities or even the activity an entity or physical object is involved in. However, device limitations are not considered. Mascolo et al. [28] and Hofer et al. [25] point out the importance of considering them when modeling the context, as it impacts the network. For instance, if a device can't execute a task, the system must be aware of this before doing the network routing and service discovery jobs. Raverdy et al. refer to those limitations in [34] by considering resource-constrained networks.

3.4 Using context for the network

The network layer must ensure QoS and security. This needs flexibility, reactivity, fault-tolerance and disruption-tolerance. Protocols, network topologies and access control policies can help to achieve this, using information such as bandwidth, communication interfaces, number of nodes and security levels. We here review how context is modeled in the domain of networking to assess these concerns.
Gold and Mascolo [20] exploit the context for mobile peer-to-peer (P2P) networks using computing resources (availability, remaining battery power) and network information (services in reach, distances). This optimizes the routing structures of the P2P network. Musolesi and Mascolo propose in [31] the Context-aware Adaptive Routing (CAR) system. CAR's network context includes dynamic information such as the change rate of connectivity, i.e. the number of (dis)connections a host experienced over a certain time. To allow an accurate routing depending on availability, CAR relies on his predictability system.
In distributed architectures, Mascolo et al. [28] characterize two types of network connection: permanent, via continuous high-bandwidth links, or intermittent, when encountering disconnections due to unpredictable failures. In intermittent networks, the performance of wireless networks may vary depending on the protocols and technologies being used. In a similar manner, Wei et al. [44] separated context in static and dynamic context. Static information could be either user's profile and history, network location (location of access points), capacities and services of the network, charging models and network policies. Dynamic information is related to location prediction and status, and current load of the network. This static/dynamic context has been also used in service discovery applications. As the heterogeneous nature of mobile environments can make service discovery tricky, Raverdy et al. deal with this issue by developing the middleware platform Multi-Protocol Service Discovery and Access (MSDA) [34]. In their network context model, static parameters could be the type of network, supported protocols, security levels, etc. Dynamic parameters are number of active users, the number of available services, the load, with a regard to control policies (e.g. incoming and outgoing messages). This implementation allows an accurate routing for service discovery thanks to context information.
Media content delivering services may also rely to network context for personalization. Yu et al. proposed in 2006 [46] a recommendation system based on a three-axis context model : user media preferences, user situation context and terminal capability context. The terminal capability contains the network profile context (including bandwidth availability) in which rules are inferred, based on network information combined with supported modality (image, video, text, each depending of the mobile device's type). Their approach combines two methods on different aspects of context: bayesian classification using the user's history and preferences (cognitive agent context) with ontology rules using information about the network's capability (terminal capability context).

3.5 Using context for social computing

In the WoT, entities form a society of objects which communicate: this is called social computing. Multi-agent models can help improving decision making and adaptivity for social computing, given the richness of context information in the WoT.
In the domain of artificial intelligence, the lack of an operational definition of context explains several failures noted in knowledge based systems according to Brézillon and Pomerol [6]. Brézillon states in [8] "these problems concern the exclusion of the user from the problem solving, the misuse and lack of knowledge-based systems and the impossibility to generate relevant explanations". Thus, an explicit context model is required. He explains in [8] the lack of consensus behind the definition of context, describing a two-sided context at different levels: *static or dynamic knowledge*, respectively any constant knowledge or changing knowledge through the whole interaction, and *contextual or contextualized knowledge*, respectively the explicit knowledge or the implicit knowledge (namely constraints) intervening in the problem solving. He defines context as a shared knowledge space that is explored and exploited by agents in the interaction. One can organize contexts into a hierarchy, by creating a context from already existing ones. Shared knowledge includes elements from the domain, the users, their environment and their interaction with the system.
Bucur et al. [9] propose to consider the Environmental context of an agent and the Organizational context resulting from their interactions. They define context as "a finality and the set of attributes that are relevant for that finality". Thus, similarly to [8], each agent has its own context and can share his knowledge through interactions. According to [7], this knowledge is divided in three categories: *proceduralized knowledge*, the shared knowledge between agents involved in the decision making step that is used directly for the problem solving, *contextual knowledge*, the implicit knowledge that influences the problem solving and *external knowledge*, the knowledge known by involved agents but that is not used in the current decision making step. The dynamic nature of the context comes from switching between *proceduralized* and *contextual knowledge*. Hence, between each decision making step, pieces of *contextual knowledge* becomes either *procedural* or *external*, and vice versa. In [5], Brézillon identifies the different sources used to build the shared knowledge: Domain, User, Environment and Interaction. To structure this knowledge, Bazire and Brézillon propose in [4] a context model that represents the components of a situation, based on Definition 1. A situation could be defined by a User, an Item in a particular Environment, and eventually an Observer ; each of these elements interferes with a related context.

3.6 Using context in the Web

Context-aware applications that rely on Web technologies aim at providing open, standard-based and interoperable services. Some approaches aim at reducing mobile user's effort when searching the Web (which is more tedious than using a desktop computer). In [11, 10] user sessions (search queries and click-through) are used as context information for query recommendation. Xiang et al. propose in [45] a context-aware query ranking system based on the consecutive user query reformulations and query specialization/generalization. A context-aware query auto-completion system is proposed in [3], using User profile, Device and Browser, Geospatial, Environment and Date/Time dimensions.
However, the problems that mobile users encounters do not only concern Web search. Mobile users also expect an adaptive Web content while browsing due to PDA and mobile phone's limited graphic interfaces. The Context-Aware Web Browser from Coppola et al. [13] allows automatic retrieval and constant update of the contextual information gathered from the surrounding environment. They rely on context descriptors using the Location, Time, Activity, Posture and Privacy dimensions, as well as information that characterize the context itself (Probability, Importance, Description and Name). Alti et al. propose in [2] a system for media content adaptation in mobile environments, based on context information. They use four context axis: User (profile, preferences), Mobile Device, Document (modality) and Service (QoS, role).
Some Context-aware systems are also Web service based. In their survey [41], Truong and Dustdar reveal that Web service context-aware systems bring additional context dimensions, such as Service/Application, Activity/Task and Team. They still widely use the Location, Presence, Individual Profile, Machine/Device and Network dimensions, which is the case for ESCAPE [43] and inContext[42] frameworks.
In the Thing-REST architecture, He et al. [24] introduce the context of things. Context is separated in

two kind of information: Semantic (human knowledge about the thing, static and predictable) and Sensing (dynamically changeable and unpredictable knowledge, gathered from sensors). Information about the user's situation, devices and preferences are also used. Each piece of information is reused across Web services.

3.7 Using context in applications

The context used depends on the application's business logic, which leads to models that correspond to the functional needs. These models are therefore highly specific, as Abowd et al. explained in [1]. However, their specificity is necessary to correctly respond to these functional needs. According to [19] and [12], the application core must be designed regardless of the context information and processing. Chaari et al. defines in [12] the context for applications as "the application's external parameters which impacts its behavior, defining new views for its data and services". They propose five dimensions: Communication, User, Terminal, Localisation and Environment.

Munnelly et al. [30] propose a modularized approach for designing context-aware applications that rely on the usage of devices. They also address the problem of "tangled code", proposing an aspect-oriented approach. They model context using eight dimensions: Device, Location, User, Social, Environmental, System, Temporal and Application-specific context. In such approach, Device, Location, Temporal, Environment and Social dimensions are quite similar to those proposed by Schmidt [39] as they are based on the earlier works from him and colleagues [40]. However, this approach uses new dimensions such as the Application context (e.g. the core functionality of an application) [30].

Kirsch-Pinheiro et al. adress in [26] the necessity for groupware systems to rely on the user's preferences to provide a better adaptation to the context. They propose a five-dimension context model, specific to such applications: Space (physical location), Tool (device and application), Time (group calendar), Community (referring to the concepts of group, users and roles) and the Process viewpoint (activities with shared objects, handled by the group). They also represent user profiles, being the preferences and the constraints to be satisfied by the system for a group member, a role or a device. This approach confronts the current user context and the profiles and situations in which they are valid, i.e. the application context [27].

According to Euzenat et al., [18], different representations of context can be modeled for the same situation. They believe that a context representation must enable the aggregation and separation of context, allowing pervasive applications to share their context, go in and go out from contexts. For Coutaz et al. [14], these applications must provide a network of context services, accessible from anywhere, structured into multiple abstraction layers. These layers are the Sensing layer (numeric observables), the Perception layer (symbolic observables) and the Situation and Context Identification layer (conditions for moving between situations and contexts). Thus, [18] and [14] share the fact that context information must be shared, accessible, separated and aggregated.

4 Synthesis

In our survey, we identified the pieces of information used to characterize the context in different fields, although this is not exhaustive. Information related to the network allows to set up accurate routing strategies in heterogeneous environments and enables efficient discovery. This also improves fault-tolerance and adaptability of systems. Some applications choose to separate network context information into static and dynamic information[44, 34]. This concept is reused in [24]: sensing/dynamic context and semantic/static context. We also see that the user's context is central in many applications, such as context-aware Web search querying [11, 10, 45, 3] and content adaptation [13, 2]. In social computing, many entities interact with each other. Each of them have their own context knowledge. For MAS, different types of knowledge are proposed in [7]: external, proceduralized and contextual.

We present in Table 1 the different uses of context information for each surveyed field. We do not consider Schilit's "Nearby" dimension [36] because of its redundancy discussed in [1] and 3.3. We also do not consider general categories of context dimensions, such as static/dynamic used in [44][34][8] and semantic/sensing context from [24], because we need more specific dimensions for our study. Furthermore, we propose the following merges for context dimensions: we decided to group the Activity dimensions with the

Process and Task dimensions, as they share the same definition [26, 41]. The relations (represented by the dimension of the same name[39]) between entities are established to carry out these activities. They form groups [26, 27], teams [41] and organizations [9]. As these relations characterize the social [15, 39, 30] aspect of these activities, we group the Relation, Organization, Team and Group dimensions under a single Social dimension. There is a redundancy between the Tool, Item and Device dimensions. As the Device dimension is the most used vocabulary for this, we put the Tool [26] and Item [4] dimensions into this category. We omit the Identity[1] as it already groups devices, users and their profiles. Network capabilities [46] and capacities [44] belong to the Communication dimension [12, 43, 42].

	Location	Time	Activity	Environment	Privacy	User	Self	Profile	History	Communication	Security	Device	Social	App-Specific	Preference	Observer
Physical World	[37] [1] [47]	[37] [1] [47]	[39] [1]	[39]	[17]	[17]	[39] [47]					[39]	[39] [47]			
Network	[44]					[44] [46]		[44]	[44]	[44] [46]	[34]	[46]			[46]	
Applications	[30] [26] [12]	[30] [26]	[26]	[12] [30]		[30] [26] [12]		[26]		[12]		[12] [30] [26]	[30] [26]	[30] [26] [27]	[26]	
Web	[3] [13] [43] [42]	[3] [13]	[43] [42] [13]	[3]	[13]	[11] [10] [45] [3] [2]		[3] [43] [42] [2]	[11] [10] [45] [3]	[43] [42]		[2] [43] [42] [3]	[43] [42]	[2] [43] [42]	[2]	
Social Computing				[9] [5]		[5]						[5]	[9]			[5]

Table 1: Summary of context dimensions used for each field

5 Discussion

5.1 Identification of context dimensions for a WoT application

We now study how these dimensions are relevant according to the WoT application architecture described in Section 2. Each of the four layers introduced by Guinard[21] has its own requirements, although we can identify common dimensions as follows: the Device dimension is obviously required by the four layers as it is what is accessed, found, shared and composed. The Security and Privacy dimensions are needed on each layer to preserve the application's privacy policies and security requirements. The spatial and temporal information are needed on each layer, so we also use the Location and Time dimensions on each layer. Preferences and profiles are also needed on each layer to take decisions that are appropriate for each client. Now we identify the main interactions that take place on each layer when an action is performed. 1) The initiative is taken by the client. In many cases, the contractor is the user (directly or indirectly), but observers (software components) have also the responsibility to check device's accessibility. 2) The client's preferences and profile may be used to appropriately respond to a demand, in compliance with his privacy parameters . 3) The action resulting from the request is performed, using secondary pieces information related to the current domain. This domain may be related to the physical environment, the society or may be specific to a particular application when composing a service.
We present in Figure 1 the context information identified for each layer of a WoT application architecture, namely Accessibility, Findability, Sharing and Composition. The first column contains dimensions related to the client (the User or the Observer). The second are dimensions related to the action actually performed: it can be the actual activity, the current communication or could be based on past history for search adaptation. The third column groups domain-related dimensions as we explained above. The others dimensions on the right (Device, Location, Time, Security, Privacy, Preferences and Profile) are the common dimensions for each field.

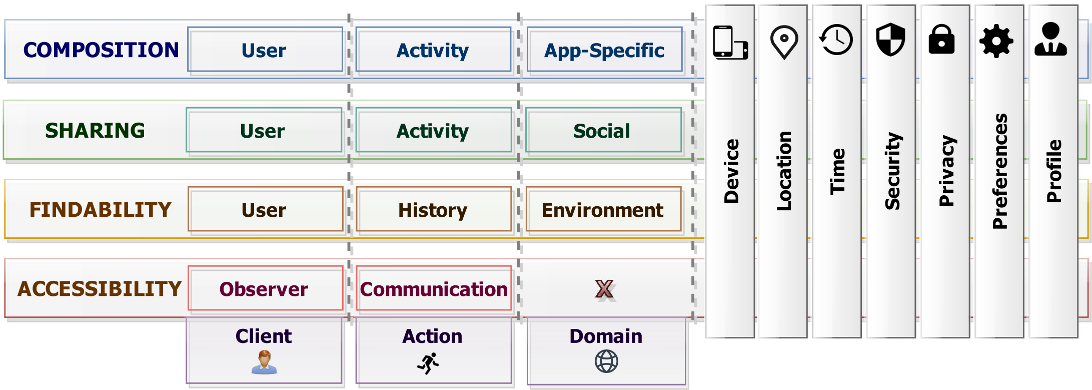

Figure 1: Context information identified regarding the architecture of a WoT application

5.2 A context model for the WoT

A WoT application consists in mobile, distributed clients and sensors connected to an application. A client could be a user using his smartphone, computer, as well as any connected device not directly controlled. Thus, information related to the communication between these entities is needed for the application to adapt its behavior (e.g. video quality adapted to the bandwidth). Clients also interact between each others. The information about their characteristics and interactions would help the application to deliver, for example, the appropriate format depending on the client's supported modality (image, sound, etc.). Each application has a proper domain, and we saw in Section 4 that domain-specific information is widely used in the literature [8, 30]. This is also the case for WoT applications. Furthermore, these applications are composed in various way and can use local services as well as the cloud. Information concerning its architecture is therefore to be considered.
In the following, we propose a context model for WoT applications. Our purpose is to identify each information that will impact the behavior of the WoT application, to properly adapt to context changes. We rely on the different context information used in the surveyed works to justify our choices. Moreover, we categorize each information into dimensions that characterize a WoT application, based on the architecture we presented in the previous paragraph.

1. **Physical.** The information about the physical world is directly collected by sensors. They capture any phenomenon that affects it and is related to space, time and nature (weather, ambient sound...).

2. **Domain.** Any domain-specific information that changes the behavior of the application. This is similar to the Application-Specific dimension but we rename it to avoid confusion with the application itself. For example, health-care applications will rely on domain-specific information to manage the privacy concerning the patient's data. Thus, User profiles can be considered as Domain-Specific information.

3. **Architecture.** The information related to the architecture of the application. An application can be composed in various ways, using different services, locally or in the cloud.

4. **Communication.** Any information related to the communication between clients and applications. For example, a context information related to Communication can be *"wireless or wired connection"*.

5. **Social.** This dimension regroups any information characterizing the client and its interaction with other clients in the application.

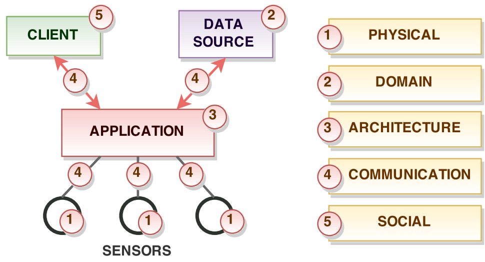

Figure 2: Context dimensions and their use in the parts of the WoT application

6 Conclusion and perspectives

In this deliverable, we described the generic WoT application architecture and identified the different fields related to these applications. We reviewed works and approaches about context modeling in these fields. Based on our study of related works, we discussed their pertinence in a WoT application and identified the layers in which they are relevant. We deduced a context model based for each part composing a WoT application based on the previously identified dimensions.
As future work, we plan to describe this context model in an ontology. We also aim at distributing these dimensions among the physical parts of WoT applications. We look forward to rely on some framework, such as MUSIC [35], to exploit these dimensions and add semantics in order to reason on them. Our main perspective is therefore to enable context-awareness while preserving the efficiency of WoT applications by designing an adaptation engine to ensure dynamic reconfiguration and adaptation for WoT application components.

References

[1] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*, pages 304–307. Springer, 1999.

[2] a Alti, P Roose, R Saffidine, and S Laborie. Towards an automatic adaptation of heterogeneous multimedia mobile applications. *Conference Internationale: Nouvelles Technologies de la Repartition - Colloque Francophone sur l'Ingenierie des Protocoles, NOTERE/CFIP 2012*, 2012.

[3] Mario Arias. Context-Based Personalization for Mobile Web Search. *Context*, pages 33–39, 2008.

[4] M Bazire and P Brézillon. Understanding context before using it. *Modeling and using context*, 2005.

[5] P Brézillon. Representation of procedures and practices in contextual graphs. *The Knowledge Engineering Review*, pages 1–26, 2003.

[6] P Brézillon and JC Pomerol. Misuse and nonuse of knowledge-based systems: The past experiences revisited. *... Systems for Supporting Management Decisions*, 1996.

[7] P Brézillon and JC Pomerol. Contextual knowledge sharing and cooperation in intelligent assistant systems. *Le Travail Humain*, pages 1–33, 1999.

[8] Patrick Brézillon. Context in Artificial Intelligence: II. Key elements of contexts. *Computers and artificial intelligence*, pages 1–27, 1999.

[9] O Bucur, P Beaune, and O Boissier. Representing context in an agent architecture for context-based decision making. *Proceedings of the Workshop on ...*, 2005.

[10] Huanhuan Cao, Derek Hao Hu, Dou Shen, Daxin Jiang, Jian-Tao Sun, Enhong Chen, and Qiang Yang. Context-aware query classification. *Acm Sigir*, 106(3):3, 2009.

[11] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. Context-aware query suggestion by mining click-through and session data. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '08*, page 875, 2008.

[12] Tarak Chaari, Frédérique Laforest, André Flory, Avenue Albert Einstein, and Villeurbanne Cedex. Adaptation des applications au contexte en utilisant les services web. *Proceedings of the 2nd French-speaking conference on Mobility and uibquity computing - UbiMob '05*, pages 3–6, 2005.

[13] Paolo Coppola, Vincenzo Della Mea, Luca Di Gaspero, Davide Menegon, Danny Mischis, Stefano Mizzaro, Ivan Scagnetto, and Luca Vassena. The context-aware browser. *IEEE Intelligent Systems*, 25(1):38–47, 2010.

[14] Joëlle Coutaz, James L Crowley, Simon Dobson, and David Garlan. Context is key. *Communications of the ACM*, 48(3):49–53, 2005.

[15] Anind K Dey. Context-aware computing: The cyberdesk project. In *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, pages 51–54, 1998.

[16] Anind K Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.

[17] Anind K Dey, Daniel Salber, Gregory D Abowd, and Masayasu Futakawa. The conference assistant: Combining context-awareness with wearable computing. In *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on*, pages 21–28. IEEE, 1999.

[18] JÉrÔme Euzenat, JÉrÔme Pierson, and Fano Ramparany. Dynamic context management for pervasive applications. *The Knowledge Engineering Review*, 23(01):21–49, 2008.

[19] Jérôme Gensel, Marlène Villanova-Oliver, and Manuele Kirsch-Pinheiro. Modèles de contexte pour l'adaptation à l'utilisateur dans des systèmes d'information web collaboratifs. In *Workshop from" 8èmes journées francophones". Sophia-Antipolis, France*, pages 5–16, 2008.

[20] Richard Gold and Cecilia Mascolo. Use of context-awareness in mobile peer-to-peer networks. In *Distributed Computing Systems, 2001. FTDCS 2001. Proceedings. The Eighth IEEE Workshop on Future Trends of*, pages 142–147. IEEE, 2001.

[21] Dominique Guinard. *A Web of things application architecture*. PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 19891, 2011, 2011.

[22] Dominique Guinard, Vlad Trifa, Friedemann Mattern, and Erik Wilde. From the internet of things to the web of things: Resource-oriented architecture and best practices. In *Architecting the Internet of Things*, pages 97–129. Springer, 2011.

[23] Dominique Guinard, Vlad Trifa, Thomas Pham, and Olivier Liechti. Towards physical mashups in the web of things. In *Networked Sensing Systems (INSS), 2009 Sixth International Conference on*, pages 1–4. IEEE, 2009.

[24] Jing He, Yanchun Zhang, Guangyan Huang, and Jinli Cao. A smart web service based on the context of things, 2012.

[25] Thomas Hofer, Wieland Schwinger, Mario Pichler, Gerhard Leonhartsberger, Josef Altmann, and Werner Retschitzegger. Context-awareness on mobile devices-the hydrogen approach. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2003.

[26] Manuele Kirsch-Pinheiro, Jérôme Gensel, and Hervé Martin. Representing context for an adaptative awareness mechanism. In *Groupware: Design, Implementation, and Use*, pages 339–348. Springer, 2004.

[27] Manuele Kirsch-Pinheiro, Marlene Villanova-Oliver, Jerôme Gensel, and Hervé Martin. A Personalized and Context-Aware Adaptation Process for Web-based Groupware Systems. In , pages 884–898, 2006.

[28] Cecilia Mascolo, Licia Capra, and Wolfgang Emmerich. Mobile computing middleware. In *Advanced lectures on networking*, pages 20–58. Springer, 2002.

[29] Michael Mrissa, Lionel Médini, Jean-Paul Jamont, Nicolas Le Sommer, and Jérôme Laplace. An avatar architecture for the web of things. 2014.

[30] Jennifer Munnelly, Serena Fritsch, and Siobhan Clarke. An aspect-oriented approach to the modularisation of context. In *Pervasive Computing and Communications, 2007. PerCom'07. Fifth Annual IEEE International Conference on*, pages 114–124. IEEE, 2007.

[31] Mirco Musolesi and Cecilia Mascolo. Car: context-aware adaptive routing for delay-tolerant mobile networks. *Mobile Computing, IEEE Transactions on*, 8(2):246–260, 2009.

[32] Jason Pascoe. Adding generic contextual capabilities to wearable computers. In *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, pages 92–99. IEEE, 1998.

[33] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context aware computing for the internet of things: A survey. *Communications Surveys & Tutorials, IEEE*, 16(1):414–454, 2014.

[34] P-G Raverdy, Oriana Riva, Agnes de La Chapelle, Rafik Chibout, and Valerie Issarny. Efficient context-aware service discovery in multi-protocol pervasive environments. In *Mobile Data Management, 2006. MDM 2006. 7th International Conference on*, pages 3–3. IEEE, 2006.

[35] Roland Reichle, Michael Wagner, Mohammad Ullah Khan, Kurt Geihs, Jorge Lorenzo, Massimo Valla, Cristina Fra, Nearchos Paspallis, and George A Papadopoulos. A comprehensive context modeling framework for pervasive computing systems. In *Distributed applications and interoperable systems*, pages 281–295. Springer, 2008.

[36] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90. IEEE, 1994.

[37] Bill N Schilit, Norman Adams, Rich Gold, Michael M Tso, and Roy Want. The parctab mobile computing system. In *Workstation Operating Systems, 1993. Proceedings., Fourth Workshop on*, pages 34–39. IEEE, 1993.

[38] Bill N Schilit and Marvin M Theimer. Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5):22–32, 1994.

[39] Albrecht Schmidt. *Ubiquitous computing-computing in context*. PhD thesis, Lancaster University, 2003.

[40] Albrecht Schmidt, Michael Beigl, and Hans-W Gellersen. There is more to context than location. *Computers & Graphics*, 23(6):893–901, 1999.

[41] Hong-Linh Truong and Schahram Dustdar. A survey on context-aware web service systems. *International Journal of Web Information Systems*, 5(1):5–31, 2009.

[42] Hong-Linh Truong, Schahram Dustdar, Dino Baggio, Stephane Corlosquet, Christoph Dorn, Giovanni Giuliani, Robert Gombotz, Yi Hong, Pete Kendal, Christian Melchiorre, et al. Incontext: A pervasive and collaborative working environment for emerging team forms. In *Applications and the Internet, 2008. SAINT 2008. International Symposium on*, pages 118–125. IEEE, 2008.

[43] Hong-Linh Truong, Lukasz Juszczyk, Atif Manzoor, and Schahram Dustdar. *ESCAPE–an adaptive framework for managing and providing context information in emergency situations*. Springer, 2007.

[44] Qing Wei, Károly Farkas, Christian Prehofer, Paulo Mendes, and Bernhard Plattner. Context-aware handover using active network technology. *Computer Networks*, 50(15):2855–2872, 2006.

[45] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. Context-aware ranking in web search. *Sigir 2010*, page 451, 2010.

[46] Zhiwen Yu, Xingshe Zhou, Daqing Zhang, Chung-Yau Chin, Xiaohang Wang, et al. Supporting context-aware media recommendations for smart phones. *Pervasive Computing, IEEE*, 5(3):68–75, 2006.

[47] Andreas Zimmermann, Andreas Lorenz, and Reinhard Oppermann. An operational definition of context. In *Modeling and using context*, pages 558–571. Springer, 2007.