Trust-based context contract models for the Internet of Things

Samer Machara Marquez, Sophie Chabridon, Chantal Taconet

Trust-based Context Contract Models for the Internet of Things

Samer Machara, Sophie Chabridon, Chantal Taconet
Institut Mines-Télécom/Télécom SudParis,
CNRS UMR 5157 SAMOVAR,
91011 Évry, France
Email: Samer.Machara@telecom-sudparis.eu

***Abstract*—With the Internet of Things (IoT) paradigm, potentially private data could be made available on the Internet. Such data could then, be consumed by a growing number of applications. The acceptance and success of new pervasive applications depend on both the protection of privacy and the guarantee of quality of context (QoC). As in the IoT producers and consumers of context are decoupled, they are not aware of each other. Therefore, it is essential to provide them with means to express their guarantees and requirements concerning QoC and privacy. For this purpose, we propose meta-models to design context contracts defining privacy and QoC agreements, independently of the consumer and the producer sides. The contracts are key to an autonomous management of QoC and privacy in the IoT. Firstly, contracts may be modified at runtime to add, edit or remove clauses. Secondly, the middleware in charge of transmitting data from context producers to context consumers (e.g., context managers) will be able to match QoC/privacy requirements and guarantees. Finally, the matching process can adapt dynamically, for instance, to the current trust level between the two parties. These contracts will participate to build trust among IoT participants.**

I. Introduction

Day after day, the Internet is extended with the interconnection of a big range of small devices (e.g., Sensors and smart objects) to build the Internet of Things (IoT) [1]. This enables everyday objects to share information among themselves and/or with other systems, thus providing events occurring in a world wide network to applications. As a consequence, mobile applications become context-aware taking into account events occurring in the environment. With the IoT, many applications can consume context data concerning user activities, behaviours, daily routines, health and welfare, which brings a lot of possible benefits to the user. However, as pointed out by Buckley [2], the IoT raises critical issues concerning privacy. Agre and Rotenberg [3] define privacy as the power someone has over a piece of information that belongs to him or her. Concretely, we define privacy as the capacity of control about what, how, when, where and with whom to share information.

Figure 1 introduces the vocabulary used in this paper. The *Context owner*, (i.e., a person, a group of persons, or an organization) is the entity having the capacity of decision and control about privacy rules over his/her/its context data. Software and hardware entities providing these context data are named *Context producers*. Entities that require context data, i.e., context-aware applications, are called *Context consumers*. Persons that access those applications are called *Context end-users*. The middleware between consumers and producers is the *Context manager*; it is in charge of allowing or denying access to context data autonomically to consumers with the appropriate QoC level, while preserving the privacy of the context owner.

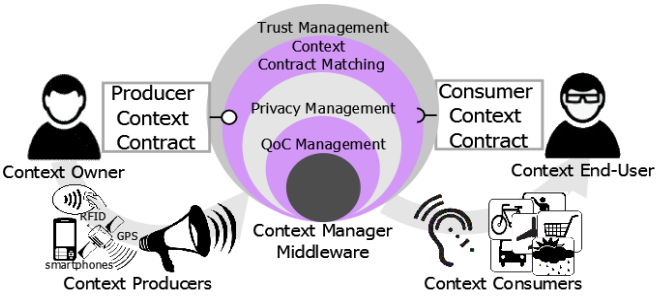

Fig. 1. Distributed view of producers, consumers and contracts

One of the main challenges of building trusted context-aware applications for the IoT is that context producers and context consumers are decoupled, i.e., they run on remote devices and are not aware of each other. Furthermore, in some cases, context owners want to remain anonymous to protect their privacy. It is therefore necessary to find a trade-off between the context owners'privacy requirements and the level of quality of context data required by applications. That is to say, how much context data is enough to achieve the objective of an application without giving all the information that is available so as to preserve privacy.

Quality of Context (QoC) is defined by a set of measurable quality criteria such as precision, error probability or freshness [4]. Through QoC, the worth of context data for a specific application is evaluated. In addition, the level of QoC delivered to context consumers is exploited for privacy purpose. For example, the level of QoC may be decreased by the context manager through obfuscation techniques to provide context data with a lower QoC level so as to protect privacy.

Limiting the QoC provided for privacy purpose is interesting, but not sufficient. To solve the trade-off between privacy and QoC, trust plays an important role. While tuning QoC for privacy, we are interested in determining

if context owners trust context end-users, as well as, if context consumers trust the context data collected by context providers. The higher the level of trust between owners and consumers, the higher the level of QoC consumers will be allowed and the higher the quality and/or the performance of applications.

We propose context contracts compliant with International and European laws concerning privacy [5]–[7].

We define two kinds of context contracts. *Producer context contracts* define clauses for the production of context data with privacy requirements (indicating the context owner demands before accepting to provide context data) and QoC guarantees (establishing the guarantees the producer is ready to fulfill with respect to QoC). *Consumer context contracts* define clauses for the consumption of context data with QoC requirements (establishing the QoC the consumer is expecting for running an application), and privacy guarantees (indicating the guarantees the consumer agrees on to protect the privacy of the context owner). The solution presented in this paper describes privacy and QoC meta-models to create rules for combining them. These meta-models are then used to define a context contract meta-model.

The proposed context contracts are beneficial both: *(1)* for defining contracts, and *(2)* for automating the management of context delivery with appropriate QoC and privacy protection. Regarding the first benefit, contracts provide a guide to formulate and develop rules, constraints and models relevant and useful for building context-aware applications, whereas context owners independently define, and then adapt their privacy rules. Regarding the second benefit, the producer contract models are available at runtime and may be modified at anytime by context owners. Such contract models offer a comprehensive approach to dynamically match QoC/privacy requirements and guarantees. Each half contract participates to an autonomic matching performed by a context manager. The matching takes into account dynamic facts concerning the current trust among producers, consumers, context data coming from the environment, and the organization of the participants. Considering privacy protection, guaranteeing the smooth operation of applications through QoC management will enable trust among IoT participants.

The remaining of the paper is structured as follows. In Section II, we survey existing privacy and QoC models and position our work relatively to the state of the art. Section III describes the basic privacy meta-models and the QoC on which our contribution relies (namely purpose, visibility and retention). Section IV presents our contribution to the modeling of context contracts. In Section V, we illustrate through a bike sharing application example the use of the meta-models we propose. Finally, we conclude the paper and discuss perspectives of our work in Section VI.

II. State of the Art of Privacy and QoC Modeling

We discuss in this section some recent research works dealing with the modeling of quality of context and privacy for context-aware applications.

Filho [8] proposes CxtBAC which is a family of context-based access control models for pervasive environments. The approach defines reference conceptual models incrementally, where $CxtBAC_0$ is the base model with the minimum requirements for any access control system. Two models are of interest to our work: the *Q-CxtBAC* , which allows to take into account QoC constraints and the *P-CxtBAC* , which considers privacy constraints. CxtBAC targets to define access control policies that are able to dynamically adjust permissions. This is performed by taking into account context information in the definition of policies. We envision to push this concept even further by allowing to change the behaviour of a context-aware application dynamically by modifying the QoC and/or privacy requirements at runtime.

[9] presents the extension of MLContext, a Domain Specific Language (DSL) tailored to model context, so that it can also model the quality of context. The proposed approach follows model-driven engineering principles and relies on the code generator of the MLContext engine. While this work is promising and close to our proposal, it is restricted to the manipulation of QoC aspects and does not consider privacy. Moreover, even though the authors claim that their solution allows an autonomic behaviour at runtime, no evaluation results are provided with regard to the performance of the code generation and to the feasibility of the proposed solution. Their proposal relates more to code generation than to exploit the context model at runtime.

[10] introduces a new protocol called Obligation of Trust (OoT). This protocol allows to express requirements and capabilities in terms of privacy between a user and a remote Service Provider (SP) using XACML (eXtensible Access Control Markup Language) [11] from the OASIS consortium. The benefit of this proposal is to increase the trust of users with proofs that SPs can meet their requirements; users will have more trust in them and will accept to share more information. However, this protocol deals only with privacy concerns and should be extended for manipulating also QoC information.

Current works on context management are starting to consider QoC and/or privacy. However they do not yet provide autonomic solutions allowing to match consumers/producers requirements and guarantees at runtime. For this purpose, it is essential to enable autonomic solutions to have access to models at runtime; our proposed models can then be transmitted, updated and transformed along the processing chain from producers to consumers.

III. Privacy and QoC Meta-models

This section introduces the meta-models we designed to represent the privacy and QoC concepts. To manipulate these concepts, we first identify four relevant dimensions in Section III-A. These dimensions are at the basis of the solution for the modeling of context contracts that we detail in Section IV.

A. From Privacy Taxonomy to Contract Dimensions

For identifying the relevant privacy protection dimensions, we follow the taxonomy proposed by Barker et

al. [12]. This taxonomy describes how to handle data privacy in practice in the domain of database systems. It contains four privacy dimensions along which data repository privacy is achieved. The four privacy dimensions are the following: *1)Purpose*: Defines for what goal the data is used; *2)Visibility*: Indicates who is authorized to access data; *3)Retention*: Specifies how long the data is retained; *4)Granularity*: Determines the level of detail at which the data is delivered. For more details about these dimensions, please refer to [13].

For the protection of privacy in the legislative domain, Greenleaf [14] has isolated ten global elements that are common to international directives (the U.S. Privacy Act of 1974 and the European laws [7]) and that are now universally accepted as part of a full data privacy law. We have correlated these ten precepts with the dimensions suggested by Barker et al. [12]. We concluded that each privacy concern is preserved through one or more privacy dimensions. This confirms the validity of Barker's approach. For context data, we extend the granularity dimension to the more general QoC concept which may actually subsume it. The granularity is considered as one QoC criterion among others.

B. QoC dimension

Various QoC criteria are associated to context data and offer many opportunities to adjust the quality level of the context data provided. This allows the context manager to define rich obfuscation solutions for the manipulation of context data at the appropriate QoC level in order to protect the privacy of the context owner. Marie et al. [15] proposed the QoCIM meta-model as a unified solution to model heterogeneous QoC criteria, after the analysis of several existing QoC models. We rely on QoCIM in our solution as it offers a generic, computable and expressive solution to handle and exploit any QoC criterion within distributed context managers and context-aware applications. QoCIM defines the concept of QoC indicator to associate a QoC criterion to a metric value. A QoC indicator can then be compared to a given QoC level. We present the way we use the QoCIM model in Section IV-B3.

IV. Modeling relationships between Producers and Consumers with Context Contracts

This section starts by discussing the various dimensions along which context contracts will be defined, based on the models introduced in Section III. We then detail the context contract meta-model used to define producer and consumer contracts.

A. Context Contract Dimensions

Figure 2 highlights the symmetry of the dimensions identified in Section III for expressing context contracts. Requirements and guarantees are defined in terms of visibility, retention, purpose on the privacy side and in terms of QoC. Each square encloses the party involved and its role in the contract. The first square (up-left) indicates the requirements of the context owner in terms of privacy. The second square (down-left) represents the guarantees offered by the context producer in terms of QoC. The third square (up-right) represents the QoC requirements of the context consumer. Finally, the fourth square (down-right) represents the guarantees in terms of privacy offered by the end-user and the context consumer.

We add the *Trust* dimension as a way to bridge the concerns of privacy and QoC. Next generation context management solutions for the IoT, have to deal with the decoupling of context producers and context consumers. Because, context producers and context consumers do not know each other, they use trust as a mechanism to establish the conditions in which both are willing to sign a context contract. As discussed in [16], trust has different implications depending on the considered point of view. From the point of view of context consumers, trust represents the consumers' degree of belief they rely on the context data that a context producer has provided them with. And, from the point of view of context producers, trust indicates the producers' degree of belief that a consumer will use the information they provide in the full respect of the expressed rules in terms of purpose, retention and visibility. The factor of trust is thus a critical element to be considered while handling context contracts at runtime. In our model, the trust determines which rules can be applied (see the trustCondition association in Figure 4). Each context consumer and context producer will have a value that indicates the degree of trust that other consumers and producers have in them. This value determines which guarantees can be applied. For instance, a context consumer with a low trust level may receive context data with lower QoC than others with a high trust level.

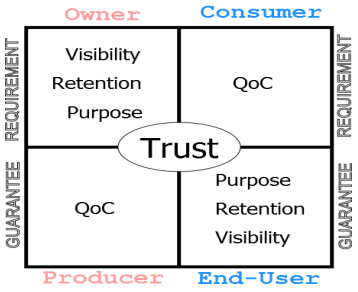

Fig. 2. Context Contract Dimensions

B. Context Contract Meta-Model

This section describes the meta-model we propose for the definition of context contracts.

According to U.S. Legal Inc. [17], the elements of a contract are "offer" and "acceptance" by "competent party" having the capacity to exchange "guarantees" to create "mutuality of obligation". Where an offer "is a promise to act or refrain from acting, which is made in exchange for a return promise to do the same". The acceptance of an offer "is the expression of the assent to its terms". A guarantee provided by a party induces the other party to enter the agreement. We designed a context contract meta-model to establish the terms of the exchange of context data between context owners and context end-users while respecting their respective requirements. The

context contract meta-model is shown in Figure 3 and the clauses condition meta-model is depicted in Figure 4.

A consumer context contract and a producer context

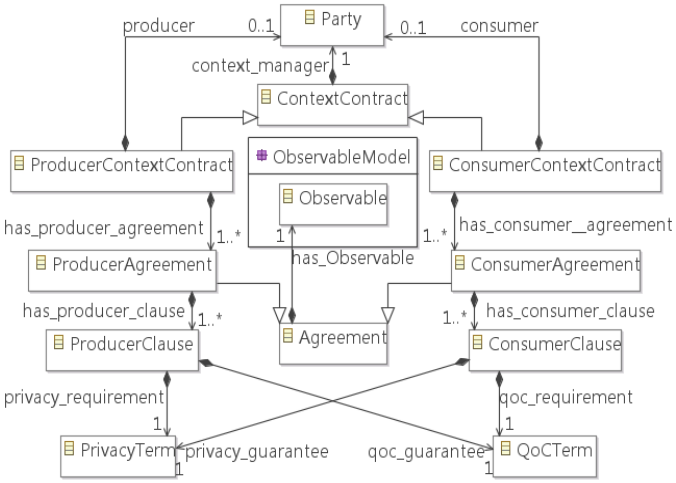

Fig. 3. Context Contract Meta-Model

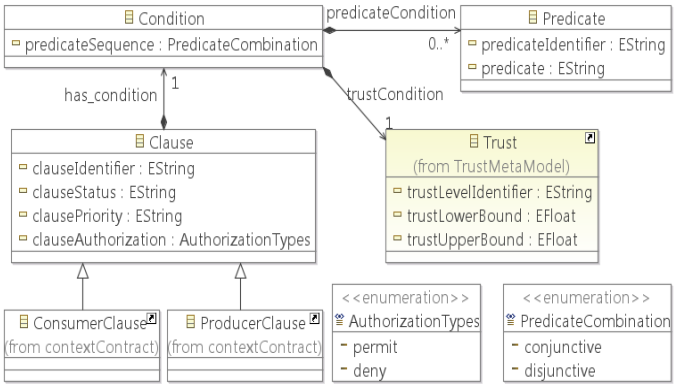

Fig. 4. Clause Meta-Model

contract establish agreements between context producers and context consumers to give access to context data while respecting the precepts of privacy and guaranteeing a good quality of context. The specificity of a context contract is that the parties enter into voluntarily without prior knowledge of each other. A context contract is composed of a producer and a consumer (one of them represented by the context manager, context_manager relation) and at least one agreement. A producer context contract and a consumer context contract are two half-contracts. These half-contracts will be matched at runtime by the context manager. An agreement is defined for one observable (abstraction that defines something to watch over (observe) [18]) on which the terms of the contract will be applied. The agreement also contains one or several clauses, which are expressed by terms. A term specifies the contractual limits and obligations that must be followed by each party. Privacy and QoC terms are described in Section IV-B3.

1) Producer Context Contract: The ProducerContextContract class identifies the contract, specifies the involved party and sets up one or several producer agreements. A producer clause contains two terms, one to set up the privacy requirements represented by the PrivacyTerm class and another one to set up the QoC guarantees represented by the QoCTerm class.

2) Consumer Context Contract: It differs on three aspects from a producer contract. First, the prefix of each class changes from producer to consumer. Secondly, the party involved in a consumer contract possesses the consumer role. And finally, the ConsumerClause class composition roles are reversed compared to the ProducerClause class. That is to say the QoCTerm class sets up the QoC requirements and the PrivacyTerm class sets up the privacy guarantees.

3) PrivacyTerm and QoCTerm classes: A term specifies the contractual limits and obligations that must be followed by each party. In our meta-model, a term acts as a requirement or as a guarantee depending on the contract type.

PrivacyTerm Class: The PrivacyTerm class is composed of three items, one for each privacy dimension (visibility, purpose and retention). As a requirement, these elements define the access restrictions and actions to be taken when a piece of context data is accessed. As a guarantee, this class expresses the way and purpose at which context data will be accessed.

In Figure 5, the three meta-models proposed in Section III are represented by their corresponding package. We show their main inner classes and the relationships that exist among them.

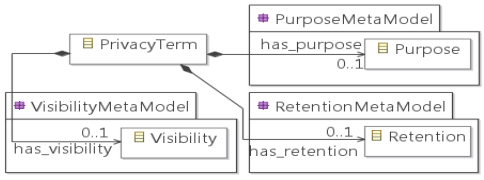

Fig. 5. Privacy Term Meta-Model

QoCTerm Class: The QoCTerm class is composed of a QoCCondition class that relates to QoCIM [15]. The qocLowerBound and qocUpperBound attributes in the QoCCondition class define the bounds of a given QoC level. In Figure 6, QoCIM is represented by the QoCIM package showing its main inner class, QoCIndicator, and the relationships with the context contract meta-model. When considered as a guarantee, the QoCTerm class expresses the way in which context data will be delivered by context producers. On the side of context consumers, QoC requirements are expressed. A context consumer therefore indicates that it needs a certain QoC level to make its calculation or to run its process without error.

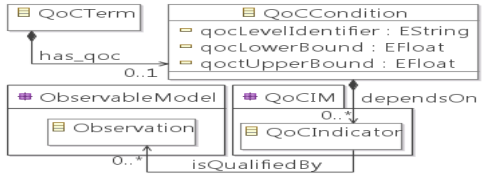

Fig. 6. QoC Term Meta-Model

V. Experimentation

The meta-models and models have been defined and validated using the Eclipse Modeling Framework (EMF) [19]. In this section, we present the models of context contracts produced for an example application of a bike sharing system. We first describe the "Meet Friends on the road" scenario,we then briefly describe the integration of the scenario in the proposed context management system and we finally detail the two half context contracts that have been established on the producer side and on the consumer side.

A. Scenario "Meet Friends on the road"

This scenario highlights how the context contracts allows the context management system to handle dynamic and autonomic aspects.We consider a bike sharing system enabling an end-user to meet his/her friends on the road or to remain hidden from them, if he/she prefers. The system lets end-users be aware of friends' location when they are nearby. Nevertheless, the bike sharing system is customizable, making it possible to share the location only with some friends. The system will only deliver the end-user's location to authorized riders as specified by contracts. It may alternatively deliver a modified location (with a decreased QoC level) in order to be only roughly located.

We illustrate this scenario with a personalized use case. David is riding back home. He decides to reveal his exact location to his acquaintances. He would appreciate crossing them on his path. The context manager is then able to distinguish situations where David meets friends on the road and agrees to share his location with them and when he is close to other end users with whom he does not want to share his location at all. After some time, David changes his mind and would prefer not to cross his office mates. So, David adds a new clause indicating to the context management system to deliver his location with a lower QoC level to his co-workers circle. Thus, the system will from now on obfuscate his exact location for those end-users.

In this example, David's and the other end-users' locations are observables. The context manager also needs to access David's contact list on his phone and we suppose that a social network API provides the trust level of end-users. The stimulus for QoC and privacy adaptation is that David is close to other riders. Using David's location and knowing the location of his acquaintances, the system calculates and offers a new path, where friends can join and share their trip, or on the contrary a new path to avoid to cross them.

B. Biking Use Case in the Context Management System

In this scenario, David is one context owner whose geo-location is watched. There is a software application in David's phone, corresponding to a context producer, which collects GPS coordinates. The "Meet friends" context consumer application, which runs on other participants phones, is dedicated to help meeting friends on the road. The context manager is a middleware entity in which both producers and consumers have trust.

We identify several autonomic cases that are handled by the context manager: *(1)* adding, modifying and removing clauses at runtime. *(2)* taking into account dynamic events such as modification of trust, visibility circles and context data. *(3)* autonomic matching of context contracts and choice of appropriate clauses. We illustrate each of these cases in the following sections.

C. Producer Context Contract Instance "Location Protection Contract"

We design a location producer context contract that is compliant to our context contract meta-model. Using a tool like the EMF suite guarantees that the modifications made on a model still conform to the associated meta-model.

The LocationProtectionContract establishes an agreement between a producer party (PhoneGPSsensor) and a context manager party (Broker1). This contract creates the obligation to protect privacy and defines the QoC guarantees over the location observed by applying the appropriate clause. These three clauses are detailed below.

Clause share_with_friends: This clause allows David to share his location with new friends automatically. It evaluates three predicates stating that David social trust level towards the end-user is high. Both David and the end-users agree to share their phone numbers after they have met at least 5 times. To activate this clause, the predicates are context-aware. For example, the system requests several dynamic pieces of information, i.e., social trust, contacts and number of encounters. The clause is defined for a given purpose, which is to_meet_me_on_bike. Concerning retention, the end-user should agree to delete David's location as soon as they are far from each other. This clause illustrates the autonomic case *(2)*, as described in section V-B, with the context-aware predicates and the dynamic modification of David's visibility circles.

Clause hide_me: This clause will lead to the obfuscation of David's location for some specific end-users (David's co-workers). This corresponds to the autonomic case *(1)* with the addition of a new clause without stopping the system. The context manager evaluates clauses at each context change. Hence, next time the context changes, new clauses are included in the evaluation. To avoid conflicts, if several applicable clauses have the same privacy requirements, the one with the higher QoC will be applied. On the contrary, if several applicable clauses have the same QoC, we use the clausePriority attribute (see Figure 4) to determine which one to apply. In this clause, David modifies the visibility, indicating what QoC level should be provided to all the end-users that belong to his co-worker circle. And he is more restrictive with respect to the trust level than in the previous clause.

D. Consumer Context Contract Instance: "Where are you"

Figure 7 represents the object instance diagram of the When_owner_decides clause from the consumer context contract. We show the model view based on the XMI file generated by the context contract editor. This

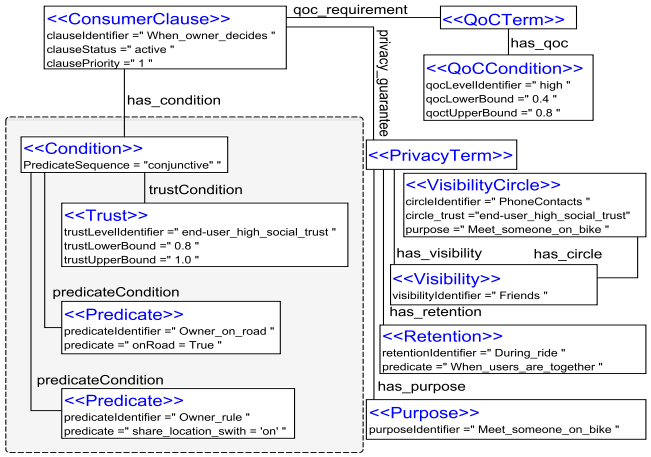

Fig. 7. Consumer Context Contract "Where are you" - Clause When_owner_decides - Model view

view highlights the correspondence of the classes of the application model with the meta-classes of the context contract meta-model. This clause asks for the access to the context owner's geo-location requiring a high level of quality (between 0.4 and 0.8) if the predicate conditions are met. The consumer also contractually declares privacy guarantees (PrivacyTerm) decomposed in the dimensions of visibility, purpose and retention. The visibility is limited to friends in which the context owner has high trust and who belong to the context owner's PhoneContacts visibility circle. The purpose is Meet_someone_on_bike. Likewise, the consumer agrees to delete the geo-location as soon as the context owner and the end-user are apart physically (retention condition).

At runtime, the context manager is responsible to match context contracts. This corresponds to the autonomic case *(3)* and is based on the symmetry in the clauses of producer and consumer half context contracts. For example, an equivalent purpose, compatible retentions and QoC levels in one of their clauses allow the context manager system to link producers and consumers.

VI. Conclusion

In this paper, we identify two essential concerns: Privacy and QoC, for the adoption by end-users of new context-aware applications relying on the IoT. We define meta-models to create rules for combining these two concepts in order to define contracts according to producers and consumers' specifications. As an example, we have described the class models designed for the scenario of a bike sharing system and derived from the context contract meta-model using a context contract model editor. The most notable advantage of our solution is its runtime adaptability. Dynamic modifications of the contract models are indeed possible as long as they remain compliant with the meta-model. We are currently implementing our context contract solution in the frame of the INCOME project [20] intended to provide multi-scale context management solutions for the IoT.

Acknowledgments

This work is part of the INCOME project (ANR-11-INFR-009, 2012-2015, http://anr-income.fr) from the French National Research Agency (ANR).

References

[1] H. Ma, "Internet of Things: Objectives and Scientific Challenges," *Journal of Computing Science and Technology*, vol. 26, no. 6, 2011.

[2] J. Buckley, "From RFID to the Internet of Things: Pervasive Networked Systems," in *Final Report Conf. DG InfSo, Brussels*. European Commission, 2006.

[3] P. Agre and M. Rotenberg, *Technology and privacy: The new landscape*. The MIT Press, 1998.

[4] T. Buchholz, A. Kupper, and M. Schiffers, "Quality of Context Information: What it is and why we Need it," in *10th Int. Workshop HPOVUA*, Geneva, July 2003.

[5] OECD, "Guidelines on the Protection of Privacy and Transborder Flows of Personal Data." 1980.

[6] EU, "EU Directive 95/46/ec on the protection of individuals with regard to the processing of personal data and the free movement of such data," *Official Journal of the European Communities*, Nov. 1995.

[7] ——, "EU Directive 2002/58/ec on the processing of personal data and the protection of privacy in the electronic communications sector," *Official Journal of the European Communities*, Nov. 2002.

[8] J. B. Filho, "A Family of Context-Based Access Control Models for Pervasive Environments," Ph.D. dissertation, Grenoble, France, 2010.

[9] J. Hoyos, D. Preuveneers, J. García-Molina, and Y. Berbers, "A DSL for Context Quality Modeling in Context-Aware Applications," in *ISAmI*. Springer, 2011.

[10] U. Mbanaso, G. Cooper, D. Chadwick, and A. Anderson, "Obligations for privacy and confidentiality in distributed transactions," in *Emerging Directions in Embedded and Ubiquitous Computing*. Springer, 2007.

[11] XACMLv3. (2013) eXtensible Access Control Markup Language (XACML) Version 3.0, OASIS Standard. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.

[12] K. Barker, M. Askari, M. Banerjee, K. Ghazinour, B. Mackas, M. Majedi, S. Pun, and A. Williams, "A Data Privacy Taxonomy," in *Proc. of the 26th British Conf. on Databases. Dataspace : The Final Frontier*. Springer-Verlag, 2009.

[13] K. Ghazinour, M. Majedi, and K. Barker, "A lattice-based privacy aware access control model," in *Proceedings CSE, Vol. 03*. Washington, DC, USA: IEEE Computer Society, 2009.

[14] G. Greenleaf, "The influence of European data privacy standards outside Europe: Implications for globalisation of Convention 108," *International Data Privacy Law*, 2012.

[15] P. Marie, T. Desprats, S. Chabridon, and M. Sibilla, "QoCIM : A Meta-model for Quality of Context," in *CONTEXT, LNCS 8175*, Oct. 2013.

[16] C. Bisdikian, M. Sensoy, T. J. Norman, and M. B. Srivastava, "Trust and Obfuscation Principles for Quality of Information in Emerging Pervasive Environments," in *IEEE PerCom Workshop Proc., Lugano, Switzerland*, 2012.

[17] U.S. Legal Inc. (2013, Jun) Elements of a Contract. http://contracts.uslegal.com/elements-of-a-contract/.

[18] C. Taconet and Z. Kazi-Aoul, "Building Context-Awareness Models for Mobile Applications," *JDIM*, Apr. 2010.

[19] Eclipse Foundation. (2010) Eclipse Modeling Framework (EMF). http://www.eclipse.org/modeling/emf/.

[20] J.-P. Arcangeli and et al., "INCOME - Multi-scale Context Management for the Internet of Things," in *Int. Conf. on Ambient Intelligence (AmI), LNCS 7683*, Nov. 2012.