Towards the definition of a framework for service development in the agrofood domain: a conceptual model

Donato Barbagallo[1], Cinzia Cappiello[1], Alberto Coen Porisini[2], Pietro Colombo[2],
Marco Comerio[3], Flavio De Paoli[3], Chiara Francalanci[1], Sabrina Sicari[2]

[1]*Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy*
{*barbagallo, cappiell, francala*}*@elet.polimi.it*
[2]*Dipartimento di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria, Varese, Italy*
{*alberto.coenporisini, pietro.colombo, sabrina.sicari*}*@uninsubria.it*
[3]*Dipartimento di Informatica Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, Milano, Italy*
{*comerio, depaoli*}*@disco.unimib.it*

Keywords: Non-functional Properties, Privacy&Security, Quality of Service, Quality of Data, Service Contract.

Abstract: This paper describes the work done in the project MoseForAgrofood (Ms4A) whose goal is to take advantage of mobile technologies to develop augmented Web services for the agrofood domain. The use of personal devices, such as smart phones and tablets, brings in specific issues such as the need to consider quality of protection, quality of services, quality of data at any stage of the service life cycle. In this paper we propose a conceptual model that aims to support the service development and provisioning addressing the above issues by managing suitable non-functional properties.

1 Introduction

Information Technology has already met the food industry and various applications have been developed for supporting products traceability and offering advanced information services to the final users. Recently, some contact-less services accessible via mobile phone have been proposed in the agrofood domain. They mainly supply users with augmented information on products (e.g., certifications, product provenance and description). An example is the *Dynamic Wine labels* service provided by the Adegga social network[1] that allows users to get more information and recommendations starting from the QR-Code on the labels attached to the bottles of wine. A limitation of such innovative services is that they provide the same information to any user, often with an unknown quality level. Services could be more effective if they would be able to offer costumized results according to user preferences and context. Customization requires the management of user personal data (e.g., preferences, allergies) and consequently it introduces the issue of *Quality of Protection* (QoP) to ensure a satisfactory level of privacy and security. Moreover, the fact that data are often collected from different sources introduces the issue of *Quality of Data* (QoD) that needs to be addressed in order to make the user aware of the dependability of the retrieved information. Finally, services are provided via technological channels and it is necessary to consider the issue of *Quality of Service* (QoS), to let the users evaluate if the service fulfills the desired requirements and constraints.

In this paper, we address these issues and we refer to QoP, QoD, and QoS with the comprehensive term of *non-functional properties* (NFPs). In particular, we describe the work done in the project MoseForAgrofood (Ms4A) whose goal is to deliver a framework for the development of personalized and integrated services that are accessible via mobile devices with contact-less interaction style. The paper provides a comprehensive conceptual model as a foundation for handling NFPs in a single framework. This approach considers the quality aspects since the beginning of the development process and incorporates such issues in the development framework and related platform. The adoption of the proposed model supports the automatization and generalization of discovery, selection and filtering activities, which is a pre-requisite to support new types of interaction with minimized

[1]http://www.adegga.com/

user involvement, as in the case of contact-less interaction with mobile devices. In fact, the model will drive the design of the MS4A platform that will be able to (i) handle the contracts (i.e., agreements on NFPs) defined with the users; (ii) select the proper services to be provided in order to both satisfy the quality requirements and enforce the privacy and security policies specified in the contracts; (iii) monitor the NF conditions under which the services are provided at run time; (iv) enforce proper actions in case the agreements are violated.

The rest of the paper is organized as follows: Section 2 proposes the comprehensive conceptual model. Section 3 contains a short state of the art of NFPs management in service-based solutions. Conclusions and future work end the paper in Section 4.

2 The conceptual model

The model proposed in this paper introduces the concepts needed to drive the development of a platform that enforces the NFPs discussed so far. Figure 1 shows the model by means of a UML Class diagram. Each concept has been detailed to give evidence to the involved elements.

Abstract class *Service* models the evident concept of service. Service providers and service users represent the stakeholders involved in the service domain. Abstract class *ServiceProvider* represents a generic entity that supports the provisioning of services. Class *Ms4APlatform* extends *ServiceProvider* and is aimed at modeling the provider of the services defined for the Ms4A Project. Similarly, *ExternalProvider* models the provider of additional services required by the Ms4APlatform to provide its services.

In order to enable the personalization of Ms4A services, users have to provide information related to their identity, life style and interests. The consumer of the services is modeled by the *ServiceUser* class and the associated data by the abstract class *UserData*. The classes *Identifiable*, *Sensitive*, and *Generic*. introduce specific categories of *UserData*. *Identifiable* deals with data referred to the identity of a user (such as family name). *Sensitive* deals with attributes related to the user's private life such as health conditions and religious creed. *Generic* defines general information that do not belong to previous classes.

Considering Ms4A platform, class *Frontend* models the client side of a *Service*, while class *Backend* the server side. The client side manages (i) the mobile device used by the user; (ii) the part of the service application that is executed on this device; and (iii) the user's profile data that need to be forwarded to the back-end for the execution of the requested service.

The abstract class *FrontendData* models generic data that are handled by the frontend. *Profile* extends *FrontendData* by defining data types required at service execution time. Particularly, a profile includes 1) identifiable, sensitive and generic data concerning personal characteristics of the user, 2) the user's preferences on the use of the service, 3) the acknowledged contract, and 4) the encryption keys that are used to address protection issues. Moreover, a service may provide various functionalities, but the user might be interested only in part of them. Therefore, class *PreferenceOnService* is used to customize the service use on the basis of the users' preferences. In particular, customization regards both the preferred usability settings and the selection of the basic functionalities.

Contract specifies the NFPs the service provider considers to provide the service and the actions that will be executed on the users data. The user must consent to the contract in order to use the service. Class *ConsentData* models the acceptance of the agreement that is established with the service provider, that includes also the consent to handle his/her data.

Due to the sensitive data that can be exchanged between user and service provider, encryption keys are used to protect the transmitted information. Class *Key* models the encryption keys that are used 1) to exchange messages and 2) to encrypt the sensitive data that refer to instances of identifiable data.

Some basic functionalities need to be supported by the service client side. These functionalities include: 1) to access data stored on the mobile device, 2) to handle the interaction with the server-side via message passing, 3) to encrypt data, and 4) to handle visualization and rendering activities. Class *FrontendActivity* models the basic functionalities of the client-side service. Advanced client-side activities are defined on top of these basic functionalities. For instance, *ConsentAcquisition* models the acceptance of the agreement that enables the user to use the service, and the provider to handle users data according to the policy specified in the contract. *ProfileDefinition* models the acquisition of the user's data required for the execution of the service, and his/her preferences with respect to the enjoyment of the service functionality. This activity generates data of type *Profile*.

The *Backend* side collects all the services used to provide the desired functionalities, including the control activities that are modeled by class *BackendActivity*. A control activity introduces the actions that are taken to monitor, correct and report of violations of conditions during and after the execution of a service. Class *Obligation* models the actions that the processor guarantees to perform, after the data have been

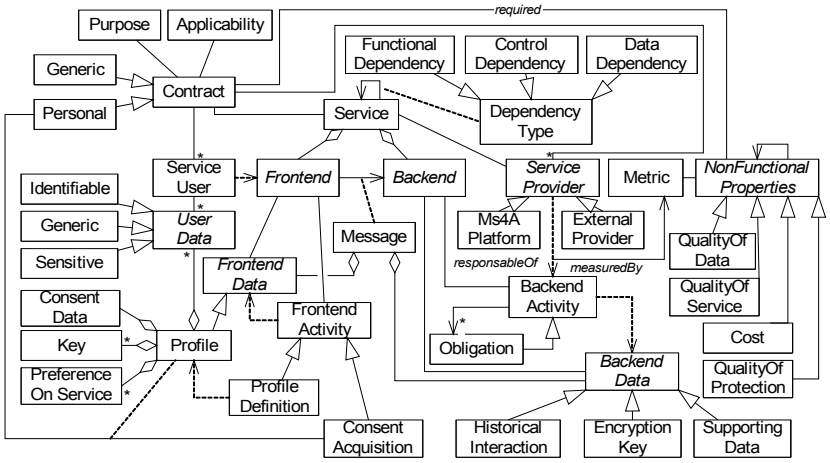

Figure 1: The main concepts of the conceptual model.

processed in case particular conditions are verified.

The abstract class *BackendData* models generic data used during the execution of a service by the service provider. This generic data type is refined into some concrete classes. Class *HistoricalInteraction* models information about the execution of the service. Class *SupportingData* models data owned by a service provider that are required to support the execution of a service. Finally, class *EncryptionKey* models the encryption keys that are used to satisfy protection requirements included in the contract.

Both frontend and backend services may depend on each other. We can identify three types of dependency: i) *Functional Dependency* models the dependencies in the execution of services; ii) *Data Dependency* models situations in which there is a data flow between services and therefore a service relies on data generated by another service; iii) *Control Dependency* models the situation in which a service is the supervisor of another one and has to check the satisfaction of NF and functional constraints. This is the case of the Ms4A platform that governs and monitors the execution of the external services.

Class *Contract* models the agreement that is established between the platform, which is responsible for providing the services, and the user(s). The agreement is said *Generic* when it is offered to a wide group of potential users, and *Personal* when it is signed between a provider and a single user that also specifies preferences on the provisioning. The agreement specifies the NFPs the Ms4A platform undertakes to provide the service. Abstract class *NonFunctionalProperty* models a generic NFP. Concrete extensions of *NonFunctionalProperty* are introduced to model properties belonging to different NF categories. More specifically, class *QualityOfService* models provisioning requirements that concern performance and use of resources. *QualityOfProtection* models security and privacy requirements. *QualityOfData* models the state of completeness, validity, consistency, timeliness and accuracy of the data managed by the service. Finally, *Cost* models the financial terms and conditions (e.g., price, insurance and compensation agreements) associated with the service. All the NF dimensions are evaluated by means of metrics. Class *Metrics* has been introduced to specify the assessment algorithms used to measure the NFPs.

Besides all these aspects, the contract specifies the general service aims and functionalities. Class *Purpose* models the service aims including the description of 1) the service functionalities, 2) the user's data that are required during the service execution and 3) the activities that will be performed on such data. Moreover, if a service includes an *Obligation*, *Purpose* will describe it together with the conditions under which the functionality is executed. Finally, *Purpose* specifies the service provider that is in charge to execute the service. Finally, a service can be thought for a specific category of users. Class *Applicability* models the conditions under which the contract can be offered to a specific class of users.

3 Related Work

In the literature, the mutual understanding between providers and consumers is typically established by specifying policies (Bajaj et al., 2006) and Service Level Agreements (Keller and Ludwig, 2003). Policies and SLAs are commonly referred to as *Service Contracts* (Comerio et al., 2009). Besides a functional description of the service, a service con-

tract includes the specification of *contractual terms* that are constraints on NFPs that can include QoS, QoD, and QoP aspects. However, QoS, QoD, and QoP are not typically considered together. The literature provides a variety of QoS models and QoS-based Web service discovery and selection approaches. Proposals such as (Wang et al., 2006) tackle the discovery by defining specific sets of quality dimensions to consider. Due to the fact that the authenticity of the advertised QoS information may be questionable, some papers in the literature (e.g., (Xu et al., 2007)) propose a model of reputation-based Web services discovery. The QoD is another aspect that must be checked during the service selection. Data quality literature provides a thorough classification of data quality dimensions, e.g., (Batini and Scannapieco, 2006). Several architectures for the data quality management have also been proposed. In particular, in (Scannapieco et al., 2004), a scenario similar to the one proposed in this paper is considered: user needs are satisfied by integrating data gathered from heterogeneous sources.

Finally, users want to subscribe a service also taking into account the satisfaction of their QoP needs. As regard the research efforts in the QoP field, requirement engineering methodologies, such as Kaos (Lamsweerde et al., 2000) and Tropos (Liu et al., 2002), can be used for the analysis and specification of security & privacy requirements. Moreover, several techniques such as anonymizing mechanisms based on data suppression or randomization (Mielikinen, 2004; Narayanan and Shmatikov, 2005) have been proposed to protect private data from unauthorized accesses. All these techniques do not require the definition of any privacy policies; rather they can be used as building blocks for realizing them. Finally, papers such as (Coen-Porisini et al., 2010) presents a conceptual model aiming at supporting the specification of privacy policies. In the present work, we adopt a similar approach to (Coen-Porisini et al., 2010) to define our conceptual model but we want to capture not only privacy but different NFPs, such as QoS and QoD.

4 Conclusions and Future Work

This paper proposes a comprehensive conceptual model to handle multiple NFPs in order to support design, selection and composition of added-value services in the Ms4A scenario. An innovative feature of our model is the capability to capture several types of NFPs, i.e., QoS, QoD, QoP, and the relationships among them. Future work focuses on the investigation of the requirements for the service design and on the development of a set of tools for the management of service contracts and for the description of NFPs associated with the services and the platform.

Acknowledgement

This work is partially funded by the regional (Lombardy) MoseForAgroFood project.

REFERENCES

Bajaj, S., Box, D., and et Al., D. C. (2006). *Web Service Policy 1.2 - Framework*. Available at: http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/.

Batini, C. and Scannapieco, M. (2006). *Data Quality: Concepts, Methodologies and Techniques*. Data-Centric Systems and Applications. Springer.

Coen-Porisini, A., Colombo, P., and Sicari, S. (2010). Dealing with Anonymity in Wireless Sensor Networks. In *Proc. of ACM SAC*.

Comerio, M., Truong, H.-L., De Paoli, F., and Dustdar, S. (2009). Evaluating contract compatibility for service composition in the seco2 framework. In *Proc. of International Conference on Service Oriented Computing (ICSOC)*, Stockholm, Sweden.

Keller, A. and Ludwig, H. (2003). The WSLA framework: Specifying and monitoring service level agreements for web services. *Journal of Network and Systems Management*, 11(1).

Lamsweerde, A., Letier, and E.Handling (2000). Obstacles in Goal-Oriented Requirement Enginnering. *IEEE TSE*, 26:978–1005.

Liu, L., Yu, E., and Mylopoulos, J. (2002). Analyzing Security Requirements as Relationships among Strategic Actors. In *SREIS 2002)*.

Mielikinen, T. (2004). Privacy Problems with Anonymized Transaction Databases. In *Int. Conf. on Discovery Science (DS 2004), Vol 3245 of Lecture Notes, Springer.*

Narayanan, A. and Shmatikov, V. (2005). Obfuscated databases and group privacy. In *CCS*, pages 102–111.

Scannapieco, M., Virgillito, A., Marchetti, C., Mecella, M., and Baldoni, R. (2004). The architecture: a platform for exchanging and improving data quality in cooperative information systems. *Inf. Syst.*, 29(7):551–582.

Wang, X., Vitvar, T., Kerrigan, M., and Toma, I. (2006). A qos-aware selection model for semantic web services. In *ICSOC*, pages 390–401.

Xu, Z., Martin, P., Powley, W., and Zulkernine, F. (2007). Reputation-enhanced qos-based web services discovery. In *ICWS 2007*, pages 249 –256.