Towards a Service Ontology Pattern Language

Glaice K. Quirino[1], Julio C. Nardi[2], Monalessa P. Barcellos[1], Ricardo A. Falbo[1], Giancarlo Guizzardi[1], Nicola Guarino[3], Mario Bochicchio[4], Antonella Longo[4], Marco S. Zappatore[4], Barbara Livieri[4]

[1] Federal University of Espírito Santo, Vitória, Brazil
{gksquirino, monalessa, falbo, gguizzardi}@inf.ufes.br
[2]Federal Institute of Espírito Santo, Campus Colatina, Colatina, ES, Brazil
julionardi@ifes.edu.br
[3] Lab. for Applied Ontology, Institute of Cognitive Sciences and Technologies, Trento, Italy
nicola.guarino@cnr.it
[4] University of Salento, Lecce, Italy
{mario.bochicchio, antonella.longo, marcosalvatore.zappatore, barbara.livieri}@unisalento.it

Abstract. In this paper we partially present an initial version of an Ontology Pattern Language, called S-OPL, describing the core conceptualization of services as a network of interconnected ontology modeling patterns. S-OPL builds on a commitment-based core ontology for services (UFO-S) and has been developed to support the engineering of ontologies involving services in different domains. S-OPL patterns address problems related to the distinction of general kinds of customers and providers, service offering, service negotiation and service delivery. In this paper, we focus on the first two. The use of S-OPL is demonstrated in a real case study in the domain of Information and Communication Technology services.

Keywords: Service, ontology pattern, ontology pattern language.

1. Introduction

Ontologies have been recognized as a useful instrument for reducing conceptual ambiguities and inconsistencies [1–3]. Ontology solutions are gaining importance in all activities that require a deep understanding of an enterprise and the business sector in which it operates [4]. Thus, ontologies, as reference models, may be useful to share the conceptualization inherent to business models within the company (e.g., among divisions and departments, favoring communication between Business/IT), as well as, to some extent, outside the company, when business models are communicated to third parties (e.g., in communication to investors and business partners).

Building ontologies, however, has not been considered an easy task. In this context, reuse is currently recognized as an important practice for Ontology Engineering, and pattern-oriented approaches are promising for supporting ontology reuse [5]. An ontology pattern describes a particular recurring modeling problem that arises in specific ontology development contexts and presents a well-proven solution [6].

Ontology Pattern Languages (OPLs) are networks of interconnected ontology modeling patterns that provide holistic support for solving ontology development problems in a given field. An OPL is not a mere catalogue of patterns. It offers a set of interrelated patterns, plus a process guiding how to use these patterns, how to combine them in a specific order and suggesting one or more patterns for a given modeling problem. An OPL addresses several types of relationships among patterns, such as: dependence, temporal precedence of application, and mutual exclusion [7]. Thus, an OPL provides explicit guidance on how to reuse and integrate related patterns into a concrete ontology conceptual model. In summary, an OPL gives concrete guidance for developing ontologies, addressing at least the following issues [7]: (i) What are the key problems to solve in the domain of interest? (ii) In what order should these problems be tackled? (iii) What alternatives exist for solving a given problem? (iv) How should the dependencies between problems be handled? (v) How to resolve each individual problem most effectively in the presence of its correlated problems?

In this paper, we present part of the initial version of S-OPL (Service OPL), an OPL that can be used for building service domain ontologies. S-OPL's ontology patterns were extracted from UFO-S [2], a commitment-based core ontology for services. .As a *core ontology* [8], UFO-S favors ontology patterns extraction [9], since it presents general concepts that span across several applications domains (e.g., education, transportation, and insurance) and can be reused when ontologies are developed to these domains. UFO-S is grounded on the Unified Foundational Ontology (UFO) [10], and it is represented in OntoUML [10], an UML profile that incorporates the foundational distinctions of UFO. By being based on UFO-S, the patterns of S-OPL are also grounded on such foundational distinctions. Moreover, by using OntoUML, it is possible to count on a well-maintained ontology engineering apparatus that can be applied for building service-related domain ontologies. Such apparatus includes model verification and validation via visual simulation [11], as well as model transformation to languages such as OWL (Web Ontology Language) [12] and support for ontology patterns application [9].

In UFO-S, service relations are characterized by the commitments and claims established between service participants, which drives the actions in service delivery. Following the modular structure of UFO-S, S-OPL comprises patterns covering four main areas: (i) *Service Offering*, which includes patterns to model a service offering to a target community; (ii) *Provider and Target Customer*, which deals with defining types of service providers and target customers; (iii) *Service Negotiation*, which deals with the negotiation between provider and customer in order to get an agreement; and (iv) *Service Delivery*, which models aspects related to the actions performed for fulfilling a service agreement. In this paper, we focus on the first two groups of patterns.

The remainder of this paper is organized as follows. Section 2 partially presents S-OPL. Section 3 demonstrates the use of S-OPL by discussing a real case study in the Information and Communication Technology (ICT) domain. Section 4 presents the final considerations of the paper.

2. S-OPL: A Service Ontology Pattern Language

S-OPL comprises a set of ontology patterns plus a process describing how to combine them in order to build a service domain ontology (an ontology about services in a specific application domain). Such patterns are organized in four groups: *Service Offering*, *Provider and Target Customer*, *Service Negotiation and Agreement*, and *Service Delivery*. Due to space limitation, only the patterns of the two first groups are presented.

The *Service Offering* group consists of five patterns, as Fig. 1 shows. The main pattern in this group is *SOffering*, which deals with the problem of modeling a service offering made by a service provider (e.g., a car rental company "XCompany") to a *Target Customer Community* (e.g., community of people and organizations that can rent cars). *Target Customer Community* is the collective of agents that constitute the community to which the service is being offered. *Target Customer,* in turn, is the role played by the agents when they become members of a target customer community. The *TCCMembership* pattern addresses this membership relation between *Target Customer Community* and *Target Customer*. Finally, the *SDescription* pattern allows describing a service offering by means of *Service Offering Descriptions*, such as folders, registration documents in a chamber of commerce, and so on.

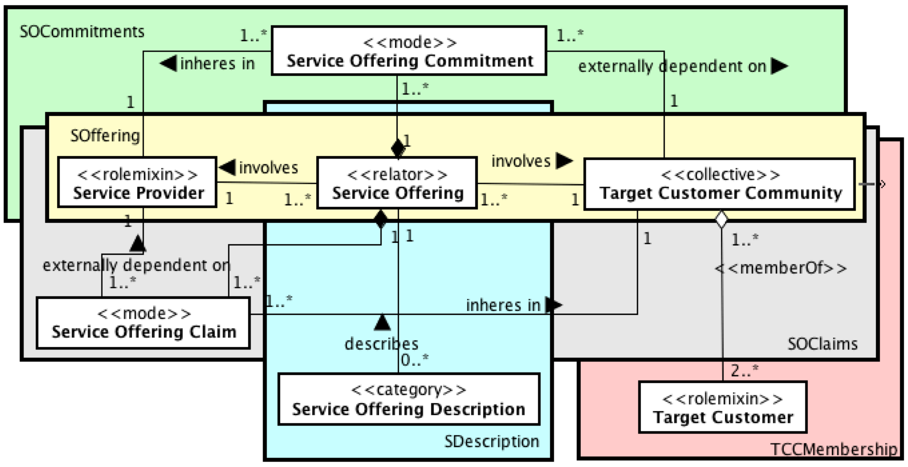

Fig. 1. *Service Offering* group.

A service offering is composed of *Service Offering Commitments* from the service provider towards the target customer community (e.g., "to grant temporary use of a vehicle to the customer") and *Service Offering Claims* from target customer community towards the service provider (e.g., "having a car available with a tank full of fuel"). *SOCommitments* and *SOClaims* patterns address, respectively, these aspects.

Service Provider and *Target Customer* are roles (in fact, technically, they are *role mixins*, i.e., roles that are played by entities of different *kinds* [10]). They can be played by a *Person*, an *Organization*, or an *Organizational Unit*. These different types of providers and target customers are addressed by the patterns of the *Provider and Target Customer* group. Fig. 2 shows the patterns of this group that describe the types of *Target Customer*. Since the patterns addressing types of providers are analogous to the ones addressing target customer types, we omit them here. The prefix of pattern names indicates the types of agents that play the roles of *Provider* or *Target Customer*: *P = Person*, *O = Organization*, *OU = Organizational Unit*.

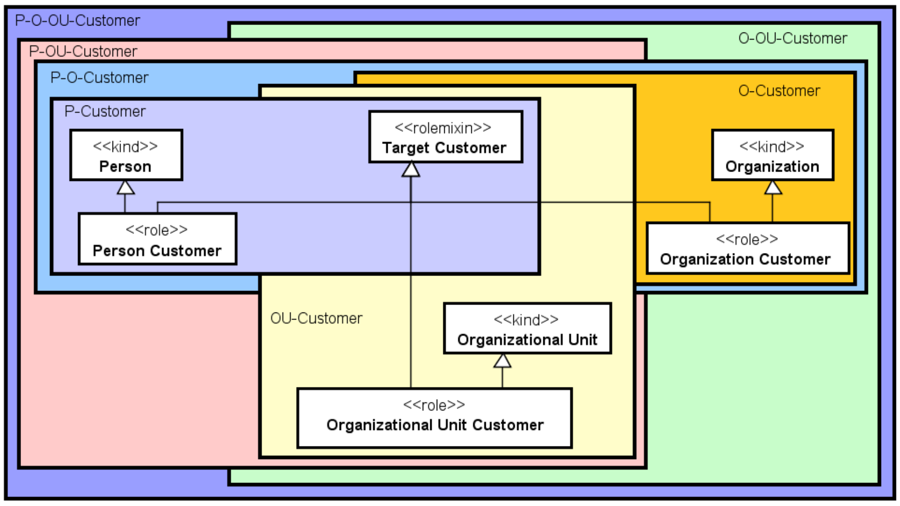

Fig. 2. Patterns for the *Provider and Target Customer* group (the provider view is omitted).

Target customers can be instances of *Person Customer, Organization Customer* or *Organizational Unit Customer*, i.e., people, organizations or organizational units. Each pattern in this group offers a different option for the ontology engineer to decide who are the provider and the target customer in the domain being modeled. The *P-Customer* and *P-Provider* patterns should be used when only persons can play these roles. *O-Customer* and *O-Provider* should be used when only organizations can play these roles. *OU-Customer* and *OU-Provider* should be used when only organizational units can play these roles. *O-OU-Customer* and *O-OU-Provider* should be used when both organizations and organizational units can play these roles. *P-O-Customer* and *P-O-Provider* should be used when both persons and organizations can play these roles. *P-OU-Customer* and *P-OU-Provider* should be used when both persons and organizational units can play these roles. Finally, *P-O-OU-Customer* and *P-O-OU-Provider* should be used when any of these types of entities (persons, organizations and organizational units) can play these roles. The patterns *P-Customer*, *O-Customer*, *OU-Customer*, *O-OU-Customer*, *P-O-Customer*, *P-OU-Customer* and *P-O-OU-Customer* are alternatives, i.e., the ontology engineer should select and use only one of them. The same occurs with the corresponding patterns related to *Provider*.

As previously discussed, an OPL includes, besides the patterns, a process suggesting an order in which they can be applied. Fig. 3 presents the fragment of the S-OPL process used in this paper, which is represented by means of an extended UML activity diagram as proposed in [7]. In that figure, patterns are represented by action nodes (the labeled rounded rectangles); patterns groups are delimited by blue lines; initial nodes (solid circles) are used to represent entry points in the OPL, i.e., patterns in the language that can be used without using other patterns first. Control flows (arrowed lines) represent the admissible sequences in which patterns can be used. Fork nodes (line segments with multiple output flows) are used to represent independent and possibly parallel paths. Join nodes (line segments with multiple input flows) are used to represent path junctions. Dotted lines delimit variant patterns, i.e., a set of patterns from which it is necessary to select only one to be used. Patterns in grey and thick black lines are the patterns used and paths followed in the example discussed in Section 3.

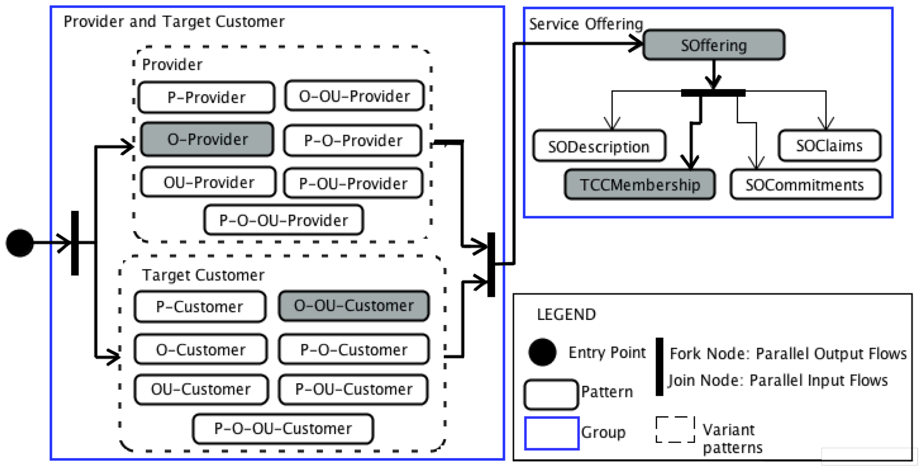

Fig. 3. The fragment of S-OPL Process used in this paper.

As Fig. 3 shows, S-OPL has only one entry point. Thus, the ontology engineer must start developing the specific service domain ontology by deciding which types of providers and target customers are involved in the service being modeled. As previously discussed, providers and target customers can be people, organizations or organizational units. Therefore, the ontology engineer must select one of the patterns of the *Provider* sub-group and one of the patterns of the target customer sub-group. Once the types of providers and target customers are modeled, the ontology engineer can focus on the service offering. The next pattern to be used is *SOffering*. Next, the following patterns can be used: *TCCMembership*, for modeling the members of the Target Customer Community; *SOCommitments* and *SOClaims*, if the ontology engineer is interested in modeling service offering commitments and claims, respectively; and *SODescription*, if the ontology engineer is interested in the problem of describing the service offering by means of a service offering description.

3. Applying S-OPL: A Case Study

In this section, we shortly present a case study applying S-OPL to develop a service ontology in a particular application domain: the Email Service Ontology (ESO). The case is concerned with an Email Service internally delivered to a big Italian company with more than 5000 employees spread out into more than 100 offices all over the country. The IT Department of the company is responsible for providing this Email Service. For doing this, the department hires two underpinning ICT (Information and Communication Technology) services provided by two different organizations: the “emailbox service” and the “networking service”. ESO was developed for a two-fold reason: (i) to integrate and make consistent the stakeholders' perspectives on the service, defining an information model able to fill the communication gap; and (ii) to design appropriate views over the model for the stakeholders.

Fig. 4 presents the Service Offering sub-ontology, which has been achieved by first eliciting a set of competency questions and then by selecting the ontology patterns capable of answering such questions properly. The following competency questions were considered: (CQ1)Which are the types of service providers? (CQ2) Which are the types of target customers? (CQ3) What is the established service offering? (CQ4) Which are the members of the target community?

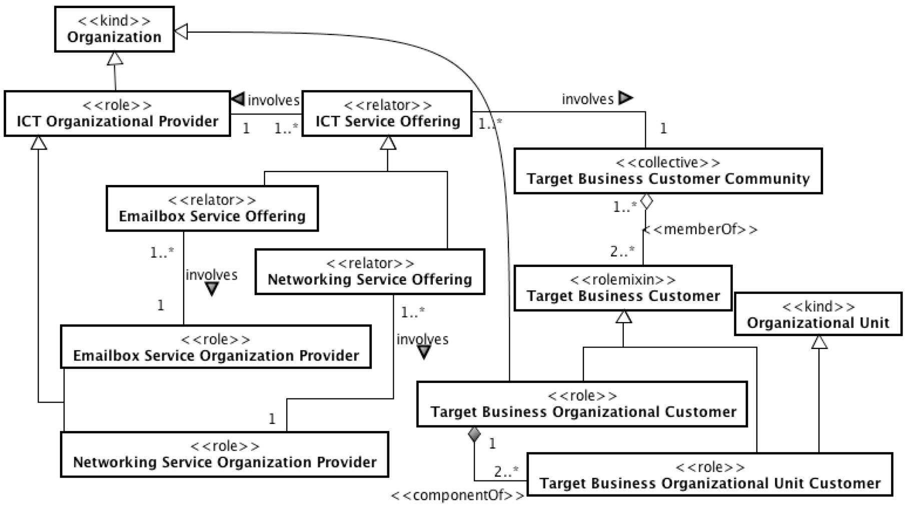
<<kind>> Organization
<<role>> ICT Organizational Provider
involves
1
1..*
<<relator>> ICT Service Offering
1..*
involves
1
<<collective>> Target Business Customer Community
<<relator>> Emailbox Service Offering
1..*
<<memberOf>>
2..*
1..*
involves
1
<<relator>> Networking Service Offering
<<rolemixin>> Target Business Customer
<<kind>> Organizational Unit
1..*
involves
<<role>> Emailbox Service Organization Provider
<<role>> Target Business Organizational Customer
1
1
2..*
<<role>> Networking Service Organization Provider
<<role>> Target Business Organizational Unit Customer
<<componentOf>>

Fig. 4. Service Offering Sub-Ontology.

The first competency question (CQ1) refers to the available types of service providers. Emailbox and networking services are offered only by organizations. Thus, the *O-Provider* pattern was used, originating the following types of service providers: *Emailbox Service Organization Provider* and *Networking Service Organization Provider*, both subtypes of *ICT Organizational Provider*.

Regarding target customers (CQ2), we used the *O-OU-Customer*, since in this specific case, offerings of ICT services are made only to organizations and organizational units (e.g., departments). Thus, *Target Business Organizational Customer* and *Target Business Organizational Unit Customer* are both *Target Business Customers*.

In order to model the service offering established between the service provider and the target customer community (CQ3), the *S-Offering* pattern was applied. Thus, we defined that the *Emailbox Service Organization Provider* is involved in the *Emailbox Service Offering*, whilst the *Networking Service Organization Provider* is involved in the *Networking Service Offering. Emailbox Service Offering* and *Networking Service Offering* are types of *ICT Service Offerings*. An *ICT Services Offering* is a relator involving an *ICT Organizational Provider* and a *Target Business Customer Community*.

Finally, for answering the fourth competency question (CQ4), we used the *TCC-Membership pattern*..By applying this pattern, we captured that *Target Business Customers* are members of a *Target Business Customer Community*.

4. Final Considerations

The main contribution of this paper is to present part of the initial version of Service Ontology Pattern Language (S-OPL). More than an additional concrete experience with the approach proposed in [7], S-OPL is a significant contribution in itself, given the importance of the service field.

To the best of our knowledge, S-OPL is the first proposal for an Ontology Pattern Language (OPL) for service modeling. Moderately related to our approach, however, we can discuss works concerning service modeling. Service Ontology of Oberle et al. [13] proposes a number of core modules that can be extended by new ontology models addressing different applications domains (e.g., healthcare and automotive). Differently from S-OPL, this approach does not propose a service modeling pattern language, with a set of high-granularity primitives (patterns) and a modeling process. There are also in the literature proposals for ontology-based modeling of services. These include the OBELIX Service Ontology and its related tools for graphical modeling of services and for knowledge-based configuration of service bundles [14]. OBELIX takes the *service as a whole* as its reuse unit, i.e., as building blocks that can be combined to form service bundles. Our proposal, in contrast, uses self-contained building blocks (organized as fine-grained patterns and guided by the S-OPL process) that address the representation of different aspects of service relations (e.g., service offering, and service agreement). These patterns are to be used in tandem for building a service ontology in a specific application domain.

From the case study, we noticed that the use of S-OPL, whose patterns were extracted from a well-founded core ontology (UFO-S), tends to bring the following benefits: (i) the resulting ontology tends to contain fewer inconsistency problems given that many of the potentially recurring source of inconsistencies tend to be solved by the basic patterns of the core ontology; (ii) the development process of the derived domain-specific service ontologies tends to be accelerated by the massive reuse of modeling fragments; and (iii) S-OPL guides pattern selection, also facilitating their combination. Despite that, some limitations were identified, such as: (a) the presence of only one entry point for the OPL and (b) the insufficient account of resources, which constitute a strategic aspect for some services (e.g., in cloud services dynamic resource allocation is a key aspect) and (c) the lack of distinction between contractual relationship and factual relationship, as a way to consider the relationships both at the contractual layer and at the operative layer. Moreover, (d) there is the need to extend the service lifecycle in order to account for service discovery and service dismission. Finally, (e) the concept of core action, as defined in [15], is missing.

The latter point gains importance in the case of instrumental services. For instrumental services, the offer of the provider does not consist in an action (e.g., cutting your hair), but rather in allowing the possibility for the user to perform a given action, which constitute the *core action* of the service. For instance, in our case study, the providers offer the Internet connection and the mailbox application, with whom they guarantee to the users that they can send/receive or manage emails. In this frame, the provider performs *supporting actions* apt at enabling the core service consumption [16]. In other words, although the actions are guaranteed by the provider, they are executed by the user.

As ongoing future works, we intend to enlarge S-OPL by adding new patterns (e.g., patterns that deal with relationships between services and business goals) and to define new entry points in S-OPL process for making it more flexible. Moreover, new applications and evaluations of S-OPL are intended to be conducted to provide relevant feedback for further improvements.

Acknowledgments. This research is funded by the Brazilian Research Funding Agency CNPq (Processes 485368/2013-7 and 461777/2014-2).

5. References

1. Mora, M., Raisinghani, M., Gelman, O., Sicilia, M.A.: Onto-ServSys: A Service System Ontology. In: Demirkan, H., Spohrer, J.C., and Krishna, V. (eds.) The Science of Service Systems. pp. 151–173. Springer (2011).
2. Nardi, J.C., Falbo, R. de A., Almeida, J.P.A., Guizzardi, G., Pires, L.F., Sinderen, M. Van, Guarino, N.: Towards a Commitment-based Reference Ontology for Services. 17th IEEE International Enterprise Distributed Object Computing Conference (EDOC). pp. 175–184 (2013).
3. Milosevic, Z., Bojicic, S., Rossman, M., Walker, M.: Healthcare SOA Ontology (Release 1). (2013).
4. Case, G.: ITIL v3. Continual Service Improvement. (2007).
5. Buschmann, F., Henney, K., Schimdt, D.: Pattern-oriented Software Architecture: On Patterns and Pattern Language. John Wiley & Sons Ltd. (2007).
6. Falbo, R. de A., Guizzardi, G., Gangemi, A., Presutti, V.: Ontology Patterns: Clarifying Concepts and Terminology. Proceedings of the 4th Workshop on Ontology and Semantic Web Patterns. , Sydney, Australia (2013).
7. Falbo, R. de A., Barcellos, M.P., Nardi, J.C., Guizzardi, G.: Organizing ontology design patterns as ontology pattern language. Proceedings of the 10th Extended Semantic Web Conference - ESWC 2013. , Montpellier, France (2013).
8. Scherp, A., Saathoff, C., Franz, T., Staab, S.: Designing core ontologies. Appl. Ontol. 6, 177–221 (2011).
9. Ruy, F.B., Reginato, C.C., Santos, V.A., Falbo, R. de A., Guizzardi, G.: Ontology Engineering by Combining Ontology Patterns. 34th International Conference on Conceptual Modeling (ER'2015). , Stockholm, Sweden (2015).
10. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models, Universal Press., (2005).
11. Guizzardi, G.: Ontological Patterns, Anti-Patterns and Pattern Languages for Next-Generation Conceptual Modeling. 33rd International Conference on Conceptual Modeling (ER 2014). pp. 13–27. , Atlanta, USA (2014).
12. Guizzardi, G., V. Zamborlini: Using a Trope-Based Foundational Ontology for Bridging different areas of concern in Ontology-Driven Conceptual Modeling. Sci. Comput. Program. Elsevier. (2014).
13. Oberle, D., Bhatti, N., Brockmans, S., Janiesch, C.: Countering service information challenges in the internet of services. J. Bus. Inf. Syst. Eng. 5, (2009).
14. Akkermans, H., Baida, Z., Gordijn, J., Pena, N., Altuna, A., Laresgoiti, I.: Value Webs: Using Ontologies to Bundle Real-World Services. IEEE Comput. Soc. 19, 57–66 (2004).
15. Ferrario, R., Guarino, N.: Commitment-based Modeling of Service Systems. Third International Conference, IESS. pp. 170–185. Springer Berlin Heidelberg, Geneva (2012).
16. Ferrario, R., Guarino, N., Fernández-Barrera, M.: Towards an ontological foundation for services science: The legal perspective. Springer Netherlands,. 235–258 (2011).