Generic Modeling and Management of Price Plans in the Internet of Services

Tom Kiemes, Daniel Oberle
SAP Research Center CEC Karlsruhe
Vincenz-Priessnitz-Strasse 1, 76131 Karlsruhe
tom.kiemes@sap.com, d.oberle@sap.com

Abstract: In order to enable trading services on the Internet and sensible cost comparisons, generic means for capturing price plans have to be established. In this paper, we present a generic pricing model which draws from established literature in business economics. We care for the context-dependency of price plans by contributing a management infrastructure for price determination.

1 Introduction

The Internet of Services envisions that services will become tradable on the Internet, be composed of services of different providers, be offered, delivered, executed, and supported by IT. New kinds of service trading platforms are arising that allow combining services as well as their integration with established enterprise applications. Thus emerge novel value webs and business models on a global scale. [BHR09]

However, in order to support this vision on a global scale, normative means for service descriptions have to be established. Efforts are ongoing to advance a Unified Service Description Language (USDL)[1] and a Service Ontology [OBBN09] to respond to this need. Such service description approaches build on top of existing WS-* specifications and allow capturing economic aspects such as information about availability, service levels, or general terms and conditions. Of particular importance are description means for capturing price plans in a generic way in order to allow cost comparisons on trading platforms. So far, the structure of generic price plans has either been discussed informally on a business economic level or has been buried in existing applications.

Therefore, this paper contributes a generic price plan model to meet this requirement. We present an ontology for capturing the static and context-dependent nature of price plans in their most common way. We evaluate the ontology in a car insurance scenario. In fact, our German lighthouse Internet of Services project THESEUS/TEXO[2] currently evaluates opening the platform for players in the German car insurance market for service trading. Unlike the current situation, players are then enabled to offer their car repair, car rental, or towing services on a trading platform where insurances can discover and combine the best fitting services for an individual damage event.

[1] http://www.internet-of-services.com
[2] http://www.theseus-programm.de/en-US/home

We realized soon, that a discovery of a service on the basis of costs requires the simulation of usage data. Consider the discovery of car rental services cheaper than 200 Euros for a specific time span. In order to obtain the actual charge, we need to indicate this specific time span, as well as add-ons, such as whether a navigation system or comprehensive coverage are desired. Only with this usage data the actual charge can be computed on the basis of the service provider's price plan. In order to address this problem, we also contribute a management infrastructure that makes use of an ontology reasoner to infer the actual charges.

The paper is structured as follows: In Section 2 we draw our attention to the modeling of price plans. The modeling can be split in two parts: the static and the context-dependent information of a price plan. Section 3 continues with highlighting the corresponding management infrastructure for simulating usage data. Finally, we point out related work and conclude in Section 4 and 5, respectively.

2 Modeling of Price Plans

In the following, we discuss our generic price plan model which consists of a static and a dynamic part. The static part requires an expressive conceptual modeling language for capturing the main concepts and relations such as “price plan” or “price component”. The dynamic part requires formal rules to express context-dependency, such as discounts for a particular user group. The combination of OWL[3] and SWRL[4] meet both requirements, share a consistent formal underpinning, and a common infrastructure (API and reasoner). Due to the lack of space we concentrate on the main aspects. The interested reader may refer to [Ki10] for a more in-depth discussion.

2.1 Modeling of the Static Information

The ontology capturing the static part of our model was created as follows. Since there is hardly any formalized generic pricing model, the first step consisted of investigating the main business literature on pricing. The main ideas are taken from Sonja Lehmann's work [Le09] which is a consolidation of existing literature and [NH05] which is one of the most cited books in the pricing domain. Both analyze the pricing aspects from a business point of view. In addition, we analyzed software implementations of large enterprise applications such as SAP CRM pricing [SA09] and Oracle pricing [Si03].[5]

[3] W3C Web Ontology Language, cf. http://www.w3.org/TR/owl-guide/
[4] Semantic Web Rule Language, http://www.w3.org/Submission/SWRL/
[5] In contrast to product pricing which can be done with existing XML-based standards, such as RosettaNet or BMEcat, we focus on service pricing which often is more flexible and complex. Thus in general it is possible to describe product pricing with our model with respect to the identified pricing aspects of [KLS02].

In a second step, we wrote a glossary which defines the concepts that are essential for defining pricing models based on the aforementioned information sources. The glossary was then formalized in OWL whereby, in general, each concept of the glossary is represented by an owl:Class in the ontology. Relevant verbs in the glossary entries, which build relations between different classes, are represented by owl:ObjectProperties. Attributes of a class are represented by owl:DatatypeProperties as well as verbs which do not refer to another class.

Finally, the resulting ontology was embedded in the larger context of a Service Ontology [OBBN09]. In doing so, we were able to use its existing definition of service concepts which are required to link a price plan to a service description, for instance. Besides, this embedding shaped and prescribed our modeling decisions. We strived to avoid arbitrariness in modeling so well known ontology design patterns [Ga05] were applied whenever possible. Additionally, we built on the foundational ontology DOLCE [GGM+02] to ground our work on a sound modeling basis. Modeling guidelines such as naming conventions and the need for multi-language documentation are also incorporated. An overview of the resulting ontology can be seen in Figure 1 depicted as UML class diagram.

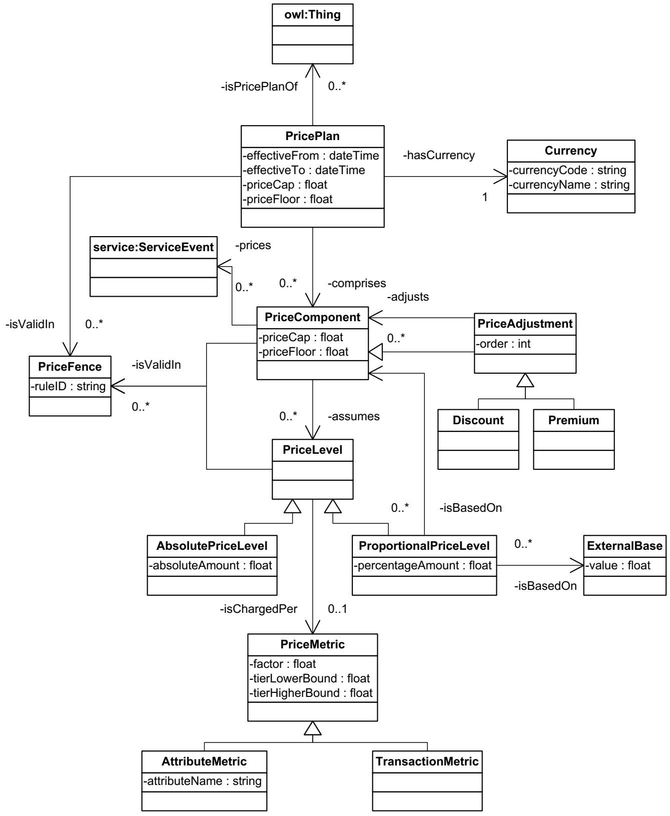

Figure 1: UML Class Diagram of Pricing Model

A service can be offered according to different *PricePlans*. In our car insurance example, a car rental service would offer different price plans for different car types. A price plan comprises several *PriceComponents.* For example, there might be price component for the number of rental days, for the optional navigation system, for the additional fully comprehensive insurance and one which charges each kilometer which is above the number of inclusive kilometers. Each price component assumes at least one *PriceLevel* which specifies the amount according to a unit. A price level for one day would be specified in case of the component concerning the rental days. The unit for the additional kilometers would be one kilometer. There are A*bsolutePriceLevels* and *ProportionalPriceLevels*. *PriceAdjustments* can be added to a price plan. For example, *Discounts* such as a weekend tariff can be granted or *Premiums* for a portable navigation system can be charged.

2.2 Modeling of the Dynamic, Context-dependent Information

The previous section did not discuss the notion of *PriceFence* because it requires a different treatment than all the other terms. Price fences represent constraints or conditions to other elements of a pricing scheme. According to [NH06], price fences are a tool for marketers to operationally segment markets. For example, car rental services can be consumed with different insurance types such as a fully comprehensive insurance with co-payment of 350€ or 100€. Different amounts will be paid for each option.

With this definition and examples in mind one can see that price fences depend on context information, such as data about the service consumer or contractual choices. This context information is dynamic, i.e., it changes in time and situation. Therefore, price fences cannot be modeled as classes or relations in an ontology. Rather we need a language to represent the rule-like nature of price fences. SWRL provides such a language and nicely blends with static information modeled in OWL. Context-dependency affects the following four object properties in our ontology:

- hasPricePlan(?x, ?y) where ?x is a ServiceDescription and ?y is a PricePlan
- comprises(?x, ?y) where ?x is a PricePlan and ?y is a PriceComponent
- assumes(?x, ?y) where ?x is a PriceComponent and ?y is a PriceLevel
- adjusts(?x, ?y) where ?x is a PriceAdjustment and ?y is a PriceComponent

That means the object properties have to be defined by one or more SWRL rules. As an example consider the rules below which state the above mentioned example regarding the different insurance types. A specific price plan comprises the price component holding the charges for fully comprehensive insurance with co-payment of 350€ or 100€ depending on what is expressed in the contract:

comprises(?x, ?y) :- PricePlan(?x), PriceComponent(?y), Contract(?c), insuranceType(?c, FullyComprehensive350)

comprises(?x, ?z) :- PricePlan(?x), PriceComponent(?z), Contract(?c), insuranceType(?c, FullyComprehensive100)

The contract changes the price plan regarding its associated price components. We introduced an additional object property called *dependsOn* to capture this information. A specific contract defines the context in this case. The adjusted rule looks as follows:

comprises(?x, ?y) :- PricePlan(?x), PriceComponent(?y), dependsOn(?x, ?c), Contract(?c), insuranceType(?c, FullyComprehensive350)

3 Management of Price Plans

In the previous section we learned how to represent static and context-dependent pricing information via an OWL ontology and SWRL rules. In this section, we discuss the management infrastructure to compare costs and generate invoices on the basis of this information.

Both the concrete prices for cost comparisons and the prices of an invoice can only be calculated when usage data is available such as the actual number of rental days, the actual driven kilometers or the utilization of a navigation system. In the case of cost comparisons for discovering services such usage data has to be simulated. Think of current car rental web sites, e.g., www.sixt.de, where the potential customer has to specify assumed usage data such as the intended number of rental days as well as the need for a navigation system. Only after the input of assumed usage data a price can be calculated the same way it would be calculated when real usage data is collected by any kind of monitoring done by IT systems or human data input.

Figure 2 sketches our management infrastructure (more info can be found in [Ki10]). Monitoring or Simulation components yield the usage data which are input to a reasoner and conform to the ontology. The usage data is represented as instances according to the Service Ontology. The reasoner, in our case KAON2,[6] automatically infers a specific price on the basis of the ontologies, rules for price fences, and concrete price plans. Before the reasoner executes a query the dependsOn for the query context has to be inserted into the ontology. In our example this would be a concrete contract. The reasoner can execute the rule only for this individual. No other dependsOn which represents a dependency for the corresponding price plan is contained in the ontology at that moment. After the query, the dependsOn has to be removed again so that for a new query, considering another contract, an appropriate dependsOn could be inserted again. Multiple dependencies to different entities are also possible.

[6] http://kaon2.semanticweb.org

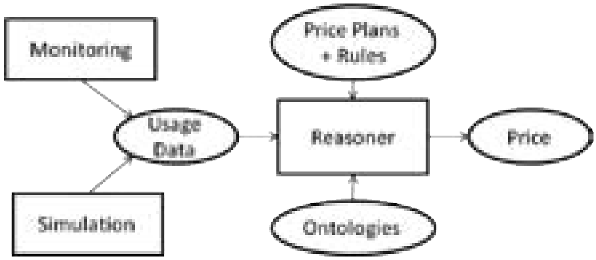

Figure 2: Dataflow of Pricing Information

4 Conclusion

In this paper we elaborated on how to build a generic pricing model for services by means of an OWL ontology. We learned that cost comparisons and calculations require the representation of context-dependent information, such as data about the service consumer that can change in time and space. In order to represent this information we applied the Semantic Web Rule Language which nicely integrates with OWL. We also introduced a machinery that applies a reasoner to finally determine a specific price based on the ontology, the rules, and (simulated) usage data.

References

[BHR09] Buxmann, Peter ; Hess, Thomas ; Ruggaber, Rainer: Internet der Dienste. In: Wirtschaftsinfomatik 5 (2009), S. 341,342

[Ga05] Gangemi, Aldo: Ontology Design Patterns for Semantic Web Content. In: International Semantic Web Conference, Bd. 3729, Springer, 2005, S. 262–276

[GGM+02] Gangemi, Aldo; Guarino, Nicola; Masolo, Claudio; Oltramari, Alessandro; Schneider, Luc: Sweetening Ontologies with DOLCE. In: Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web Bd. 2473, Springer, 2002, S. 166–181

[KLS02] Kelkar, O.; Leukel, J.; Schmitz, V.: Price modeling in standards for electronic product catalogs based on XML, Proc. of the 11th Int. WWW Conference, 2002, S. 366–375

[Ki10] Kiemes, Tom: Context-aware Modeling and Management of Service Pricing Strategies, Masterthesis, Hochschule Karlsruhe – Technik und Wirtschaft, April 2010

[Le09] Lehmann, Sonja: Pricing Strategies of Software Vendors. In: Wirtschaftsinformatik 6 (2009), S. 1–10

[NH05] Nagle, Thomas T.; Hogan, John E.: The Strategies and Tactics of Pricing. 4. Prentice Hall, 2005

[NH06] Nagle, Thomas T.; Hogan, John E.: Segmented Pricing: Using Price Fences to Segment Markets and Capture Value. In: SPG Insights (2006)

[OBBN09] Oberle, Daniel; Bhatti, Nadeem; Brockmans, Saartje; Niemann, Michael: Countering Service Information Challenges in the Internet of Services. In: Wirtschaftsinformatik 5 (2009), S. 370–390

[SA09] SAP AG: SAP CRM Pricing Documentation. http://help.sap.com/saphelp_crm70 /helpdata/EN/77/96ac30ac1c7644ad9995e6af688ad8/frameset.htm, 2009.

[Si03] Siebel Systems, Inc: Pricing Administration Guide. http://download.oracle.com/docs/cd/ E05554_01/books/PDF/PriceAdm.pdf, 2003. – Last Accessed: 2010-05-12