Personalized Configuration of Immaterial Products

Tatiana Levashova*†, Michael Pashkin*
*St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
St. Petersburg, Russian Federation
†ITMO University, St. Petersburg, Russian Federation
tatiana.levashova@iias.spb.su, michaelpashkin@mail.ru

Abstract—**Product configuration deals with design of a new product from existing components. Recently, research on product configuration has shifted to the stage of conceptual modelling. Conceptual product models do not depend on the modelling purpose and therefore can be tailored to the current customer needs. The paper proposes an ontology-based scenario for configuration of immaterial products. The scenario suggests three product configuration operations: removal, supplement, and change. Product customization is supported by involvement of the customer in the process of configuration and by using information from the customer profile. The scenario execution is demonstrated by a particular case of configuration of a mobile operator product in the form of supplement this product with a service. An ontology for mobile product operator is proposed. OWL and SPARQL are used for ontology specification and querying.**

I. Introduction

Product configuration is an activity of customising a product to meet the needs of a particular customer. Informally, configuration can be defined as a "special case of design activity, where the object being configured is assembled from instances of a fixed set of well-defined component types which can be composed conforming to a set of constraints" [1]. The solution for the configuration task is a set of instances and in some cases also connections between these instances.

To date, numerous configuration systems have been developed [2]. Such systems are one of the most successfully applied artificial intelligence technologies. Whereas initial research effort focused mainly on the actual configuration process for solving product configuration problems, such as the rule-based approach [3] and the constraint satisfaction problem approach [4] – [6], recently, attention has been directed towards conceptual modelling of customizable products [7] – [11]. Started from configuring material products such as automobiles, computers etc., in time, immaterial products such as software [12], [13], Web-services [14], [15], insurance products [16], etc. became a point of interest. The immaterial products are characterized by their immaterial nature, strong dependency supply on demand, incorporation of production processes into consumption processes, and fast product changeability.

Conceptual modelling underlies the ontology-based approaches to product configuration (e.g., [9], [17] – [21]). Such approaches deal with problems of domain ontology creation, matching vocabularies of customers and domain ontologies, and development of product configuration systems. Most modern approaches use OWL (Web Ontology Language) [22] to ontology modelling, and SWRL (a Semantic Web Rule Language Combining OWL and RuleML) for specification of configuration rules. Ontologies are queried by means of SPARQL (SPARQL Protocol and RDF Query Language) [23]. The configuration systems are implemented in Java, mainly. Currently, OWL and SPARQL are the official Recommendations of W3C [24].

The present research proposes an ontology-based approach for immaterial product configuration. Configuration of mobile operator products in the form of services is used as an application domain of the approach. The mobile operator products are developing rapidly and constantly being improved. Some components are outdated, some appear. Customers of such products want to these products would meet up-to-date requirements and needs. Therefore, development of approaches enabling to configure mobile operator products in a personalised manner is an important problem. The proposed approach is implemented using OWL and SPARQL for ontology specification and querying.

The rest of the paper is structured as follows. Basic scenarios of product configuration and the proposed ontology-based scenario are discussed in Section 2. Section 3 proposes an application of the proposed scenario for reconfiguration of a mobile operator product through expansion of this product with a service. Main concluding remarks are summarised in the Conclusion.

II. Ontology-based product configuration

The purpose of product configuration is design of a new product from existing components. Generally, two approaches to product configuration can be distinguished: configuration from scratch and reconfiguration of an existing configuration. Bellow, common scenarios these approached follow are described.

A. Configuration from scratch

The scenario of configuration from scratch starts with capturing customer requirements to the product. The set of these requirements comprises all the customer requests to the product, product options, components, functions, etc. without any restrictions. Based on the set of customer requirements an initial product configuration is built. This configuration as much as possible corresponds to the customer needs. Then the initial configuration is modified according to the customer restrictions. An elementary example of the customer restrictions is the price this customer is ready to pay for the product. Besides this typical restriction, types of such

ISSN 2305-7254

restrictions depend on the sort of configurable product. For instance, if immaterial product is a guided tour the customer may request for a guide with certain language skills; if such product is vehicle insurance services the restrictions on these services can be imposed by the vehicle (e.g., the kind of vehicle alarm system installed may increase or decrease the policy price), etc.

B. Reconfiguration

The reconfiguration scenario supposes existence of product configurations that were built previously for this product. The scenario involves some modification of an existing product through supplementing it with components, removing components, or changing product properties.

Just like the scenario above, the reconfiguration scenario starts with capturing customer requirements to the product. The set of requirements is the basis to select an initial configuration from the set of existing ones. It is requested that the configuration would meet as much customer requirements as possible or satisfy most of the customer restrictions. Then, the initial configuration is modified according to the customer requirements taking into account the customer restrictions.

If a configuration with a maximum set of components is selected but the customer needs a reduced configuration then the extra components are excluded from the initial configuration. On the contrary, if the selected configuration is insufficient to the customer then this configuration is supplemented with lacking components. In the both cases the price of the final product is recalculated accordingly. Configuration supplement may considerably enhance the potential product functionality. On the customer request the set of functions can be reduced. This reflects in the price of the product correspondingly.

C. Ontology-based scenario for product configuration

The scenarios of configuration from scratch and reconfiguration have much in common. Particularly, they both deal with some modification of an initial configuration according to the customer requirements and customer restrictions. Configuration through modification is the main focus of the proposed here ontology-based scenario.

Behind the scenario a product ontology lies. Such ontology is a conceptual model of the product, which represents the product's components, their properties, and the relationships between these components. The ontology model (O) is formalised as $O=(C,I,R)$, where C – a set of concepts (classes in OWL) describing the product; I – a set of instances (individuals in OWL); R – a set of unary and binary relationships (roles in OWL).

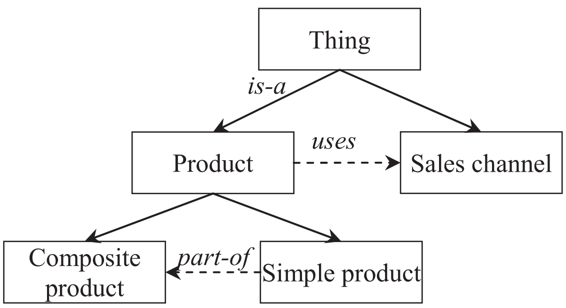

Fig. 1. Product ontology: upper level

Fig. 1 proposes a possible upper level for the product ontology. *Product* can be simple or composite. *Simple product* is a self-contained product, which does not have any products as its parts. Simple products may be components of composite products. *Composite product* is a result of union of other products which, in turn, can be simple or composite. Products are sold through sales channels. *Sales channel* is a way of bringing products to market so that they can be purchased by consumers.

Like the two scenarios above, the ontology-based scenario for product configuration (Fig. 2) starts with capturing customer requirements to the desirable product and their specification. The customer requirement specification is in the form of a list of terms representing a) product components to be subjected by configuration modification operations, and b) operation types. In terms of the product ontology, the specified product components correspond to concepts or instances defined in the ontology. It is supposed here that the customer uses the ontology vocabulary to formulate the requirements. Otherwise, the procedure for translation of the customer vocabulary into the ontology's one is needed.

Formally the customer requirements specification (S_a) is represented as $S_a=(C_a,OP_a,R_a)$, where C_a – a set of terms representing customer requirements; OP_a – a set of configuration modification operations; R_a – a set of relations over the sets of terms and operations ($R_a \subseteq C_a \times OP_a$).

Three modification operations are provided for: *supplement*, *removal*, *change*. The supplement operation is intended for extension of the product configuration with components. The removal operation is intended for some configuration reduction. The change operation is used to modify components' properties.

If the customer intends to supplement the existing configuration, the customer requirement specification is matched against the product ontology. The result of matching is a set of mappings M between the specification concepts and ontology elements $M: C_a \to C \bigcup I$. Further scenario execution depends on what kinds of ontology elements (concepts or instances) have found the mappings. For the mappings $M_1: C_{a_1} \to C, M_1 \subset M, C_{a_1} \subset C_a$ instances of the classes C are searched for in the product ontology. If for a class $c \in C$ the ontology does not specify direct instances then instances of the subclasses of the class c are selected. In the result, for the mappings M_1 a set of instances I_1 $(I_1 \subset I)$ is defined. For the mappings $M_2: C_{a_2} \to I, M_2 \subset M, C_{a_2} \subset C_a$ instances having relationships with instances from I are searched for. The result is a set of instances I_2 $(I_2 \subset I)$. It is supposed that one specification term is mapped to one and only one ontology element, i.e. $M_1 \bigcap M_2 = \varnothing$ and $C_{a_1} \bigcap C_{a_2} = \varnothing$.

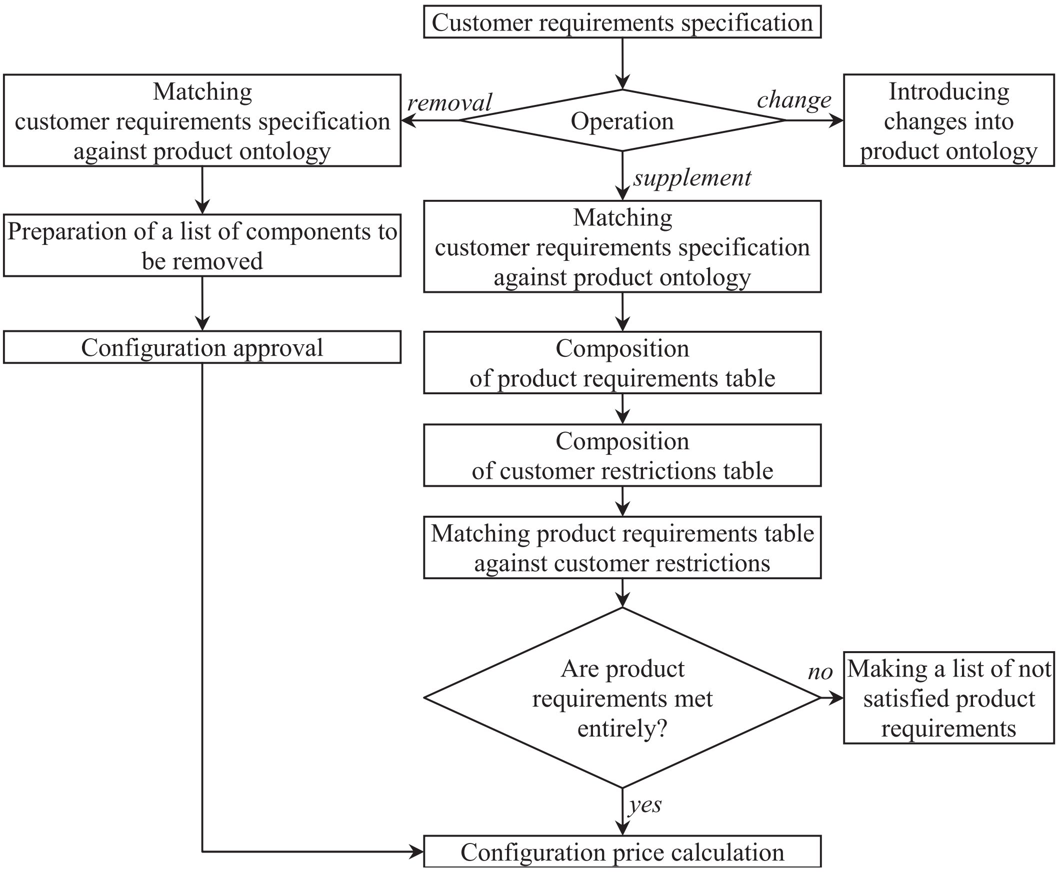

Fig. 2. Ontology-based scenario for product configuration

When searching for instances having relationships with instances I all kinds of relationships are considered except "is-a". In specific domains and scenarios the variety of relationships to be considered may be reduced by specifying the kinds of relevant relationships.

Based on the mappings M a table representing product requirements is produced (Table I). In the table, $i \in I$; the top row ($i_{1 \ldots n}$) represents the names of instances $i_r \in I_r$, $I_r = I_1 \bigcup I_2$, where n is number of instances in I_r; i_{nk} is the name of the instance k having relationship with the instance n; i_{nlp} is the name of the instance n having property named l and the value of this property is p.

In general, different columns may contain different raw number, as for the instances represented in the top row different number of relationships may be specified in the ontology.

TABLE I. PRODUCT REQUIREMENTS TABLE

i_1	i_2	…	i_n
i_{12}	i_{21}	…	i_{nk}
…	…	…	…
i_{1k}	i_{2k}	…	i_{nlp}

The product requirements table is the source to make decision if the requested product extension meets to the customer restrictions. One of the immaterial product characteristics is incorporation of production processes into consumption processes. A consequence of this is that the customer can be easily involved in the process of immaterial product configuration. The proposed scenario suggests two ways of accounting the customer restrictions. The first one is involvement of the customer into the configuration process. The second way is using of the customer's profile [25] to read information of the interest.

The customer restrictions are specified in the form of a customer restrictions table. Initially, this table is a copy of Table I. Each cell of the customer restrictions table is compared to the customer information about the represented components. At first, the instances connected by binary relationships are analysed. If the customer confirms existence of the relationship between the components, the corresponding cell remains filled; otherwise this cell is marked by any special symbol. If all cells representing instance to which the customer confirmed no relationships become marked, the cell representing unary relationship for this instance is marked as consequence. After binary relationships, unary relationships are considered in a similar way. For this kind of relationships the properties' values are supposed to be substituted for the values the customer provides. This is possible if there are no conflicts between the values represented in Table I and the customer values. Otherwise the conflict resolution scenario comes into play. This scenario is not considered in the paper.

The product can be extended with the component represented by the instance i_{nk}, i.e. the configuration can be supplemented with the component represented by the instance i_{nk}, if

- for each i_{nk} of the instance i_n the relationships indicated in Table I exist in the customer restrictions table;
- the instance i_n does not have any requirements (Table I does not contain marked cells below the headline "i_n");
- the customer is satisfied by the property values of the instance i_n.

For the relationships represented in the product requirements table, but missing in the customer requirements table, a list of missing components is made out.

When the customer wants to reduce the existing configuration, i.e. to remove "extra" components, the customer requirements specification (S_a) is represented as $S_a = (C_e, \text{Remove})$, where C_e – a set of customer terms representing components to be removed; $\text{Remove} \in OP_a$. Instances related to the elements of the set C_e are searched for in the product ontology. The process of identification of the related instances follows the same principles as in the supplement operation. The components from the set C_e and related to them compose the set of components I_d ($I_d \in I$) to be removed. The set I_d is made up of triples (i_{d1}, i_{d2}, i_{d3}), where i_{d1} $(i_{d1} \in I_d)$ – component that the customer wants to remove; i_{d2} $(i_{d2} \in I_d)$ – component related to i_{d1}; i_{d3} $(i_{d3} \in I_d)$ – component related to i_{d2}.

The set I_d is analysed if all the components represented by the elements of this set can be removed. The analysis is performed to be sure that the resulting reduced configuration is correct. Two rules are proposed to prevent a failed configuration.

Rule 1. If the product ontology specifies the relation $i_{d2}\ r\ i_{d3}, r \in R$ but the customer does not agree to remove the component i_{d3} then the component i_{d2} cannot be removed.

Rule 2. If the quantity of triples (i_{d1}, i_{d2}, i_{d3}), in which i_{d1} and i_{d2} represent the same components, is more than 1, then the customer is informed that the component to be removed is used by a number of other product components, which have to be removed as well. The decision about removal is up to the customer.

The operations of supplement and removal end up with calculation of the product that has been configured.

The change operation does not concern any customer requirements. The operation deals with changes in properties of products. An example when this operation is resorted to is changes in the cost of a component. The operation is used to update the cost of this component in the product ontology. As the ontology does not represent the product configurations, the operation does not have to recalculate prices of configurations that contain this component.

In the following Section an application of the proposed scenario for configuration of mobile operator products by an example of supplement operation is described.

III. Configuration of mobile operator products

The ontology developed for mobile operator products is based on the SID (Shared Information/Data Model) [26] and TAM (Telecom Application Map) [27]. These models were proposed by the global member association for digital business – TM Forum. TM Forum unites telecommunication enterprises and their suppliers with the purpose of preparing standards, recommendations, and models for information technologies in the telecommunication domain.

A fragment of the ontology for mobile operator products is represented in Fig. 3. The ontology is implemented as OWL ontology. The upper level of the ontology is represented in Fig. 1 and described above. In Fig. 3 kinds of *Simple product* are represented by concepts *Resource*, *Service*, *Basic fees*, and *Content*. These simple products make up composite products.

Resource is a physical or logical component of mobile operator's infrastructure or its inventories. Resources can be physical and logical. *Physical resources* are physical devices and equipment (e.g., SIM-cards, mobile devices, modems, etc.). *Logical resources* are logical entities requiring materials accounting and used by equipment and information systems of mobile operator (e.g., numbering capacity, pool of IP addresses).

Service is a kind of immaterial product. Service represents an implementation of supplier's proposal from the perspective of this supplier.

Basic fees is a list of available services and rules for calculation of their costs.

Content is information intended to owners of mobile devices.

Mobile operators can sale their products through *subscriber self-service* (e.g., personal account, Internet-shop) or *structures of customer service* (e.g., a special shop). Such systems and structures are kinds of *Sales channel* (not depicted in Fig. 3).

The concepts used in the ontology are characterised by sets of properties. In the figure, exemplified properties of *Form* and *Number* are given. These properties characterise the concept *SIM-card*. Instances of the concept can have values FF, 2FF, 3FF, 4FF for the *Form* property and identification card number for the *Number* property.

In the example considered in this Section, the customer would like to supplement the current product configuration

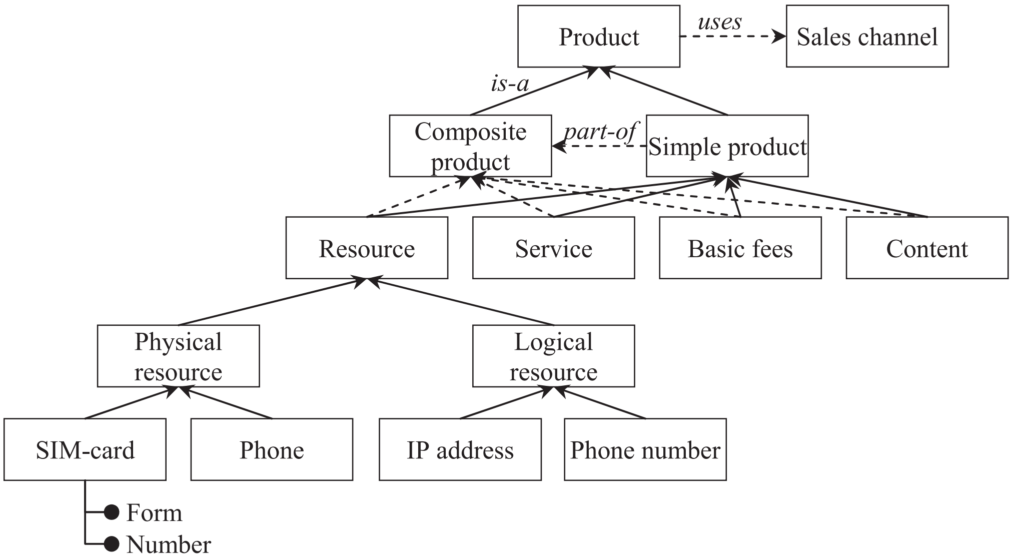

Fig. 3. Ontology of mobile operator products: fragment

with the service that allows mobile devices to locate this customer. The customer requirements specification comprises concepts *Service* and *Location*, and specifies the operation as *Supplement*.

Fig. 4 shows a specialisation of the concept *Service* in the ontology of mobile operator products. Shadowed boxes in the figure represent instances. The ontology comprises both terms contained in the user requirements specification. As can be seen from the figure the operator can propose two variants for implementation of the requested service. The *Location* service may be implemented either as *Locator* or *Satellite*. *Locator* service defines the customer location at the registration in the network. *Satellite* service defines the customer location using GPS. This service is available if in the mobile device of the customer the application *Smart Positioning* is installed. This application is designed for *Android* (a kind of *Operational system*).

Matching the customer requirements specification against the ontology of mobile operator products is implemented as a series of requests to this ontology. SPARQL 1.1 is used to requests implementation. For simplicity, class properties or unary relationships are out of the consideration in the proposed example.

At first, it is needed to find out if the ontology represents a location service. In other words, if the ontology represents concepts *Location* that is a kind of *Service*. The request (Fig. 5) returns direct and remote subclasses of the class *Service* up to the lowest class taxonomy level. Table II presents the result of request execution. It is seen form the table that class *Location* is a subclass of class *Service*, i.e. a kind of service.

TABLE II. CLASS *SERVICE* AND ITS DESCENDANTS

services
Service
Cost-free service
Basic
Extra
Pay service

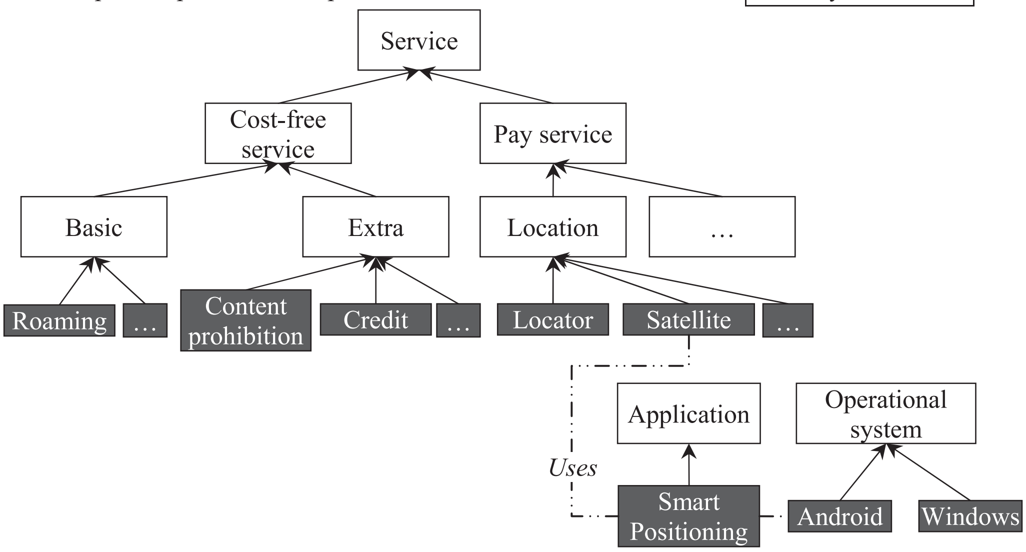

Fig. 4. Service ontology (an example)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX occ: <http://www.semanticweb.org/tanya/ontologies/2016/2/ontology-39#>

SELECT ?services
WHERE {?services rdfs:subClassOf* occ:Service. }
```

Fig. 5. Request for search for subclasses of *Service* class

Location

The request for ways how the location service can be available is used to compile a product requirements table. The request searches for instances of the class *Location* (namespaces are the same as in Fig. 5):

```
SELECT ?Location
     WHERE { ?Location rdf:type occ:Location. }
```

The result of request execution is a list of instances requested (Table III). These instances represent the ways of service implementation.

TABLE III. A SET OF INSTANCES FOR *LOCATION* SERVICE

Location
Satellite
Locator

Now, according to the ontology-based product configuration scenario, instances having relationships with *Satellite* and *Locator* should be found in the ontology of mobile operator products. The following request template is used for this:

```
SELECT *
    WHERE { occ:<Concept> ?relation ?object.
    FILTER (?relation != rdf:type)},
```

where < Concept > is the name of the instance requested.

For *Locator* the request returned nothing. This means that for this instance no relationships except "is-a" are specified. For *Satellite* the request result is presented in Table IV.

TABLE IV. RELATIONSHIPS FOR INSTANCE *SATELLITE*

relation	object
uses	SmartPositioning

The last request for relationships concerns relationships

request execution says that "SmartPositioning" uses "Android".

The product requirements table (Table V) looks as below.

TABLE V. REQUIREMENTS TABLE FOR LOCATOR SERVICE

Locator	Satellite	SmartPositioning
	SmartPositioning	Android

In the given example a customer restrictions table is composed based on the customer profile. Profile model is ontology-based. It complies with the ontology of mobile operator products. A profile comprises descriptions of the customer (personal data, preferences, etc.) and characteristics of the personal mobile device for which the product is configured. Fig. 6 demonstrates an example of the customer profile.

Another series of requests is used to built a customer restrictions table. These requests are to the customer profile for definition if in the customer device, components corresponding to the instances from Table V are available. The requests on instances substitute for confirmation of the relationships by the customer and allow avoiding involvement of the customer in the configuration process.

Based on Table V it can be concluded that the service *Locator* does not make any demands to the customer device. This service is available to any kind of mobile device. The request for search for availability of "SmartPositioning" application in the customer device looks as follows:

```
SELECT ?class
  WHERE { ?class ?relation ?object .
  FILTER (regex(str(?class), "SmartPositioning")) }
  LIMIT 1.
```

This request returns nothing. From this point the series of requests can be stopped. A request for search for operational system Android has no sense. Nevertheless, the customer restrictions table is given below to follow the scenario.

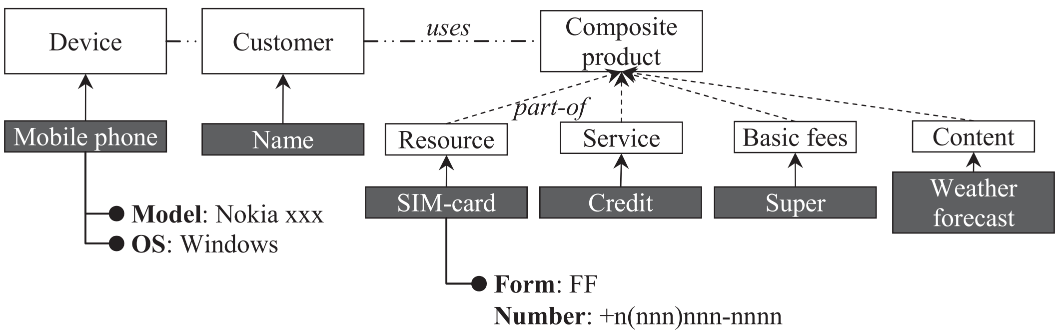

Fig. 6. Customer profile (a fragment)

TABLE VI. CUSTOMER RESTRICTIONS TABLE

Locator	Satellite	SmartPositioning
	No	No

Based on matching of Table V and Table VI the customer is offered to connect *Locator* service.

If the customer desires to know which requirements are made to his/her device in order to he/she would be able to connect the service *Satellite*, then for the series of requests to the customer profile, a list of instances representing missing product components is produced.

VII. CONCLUSION

An ontology-based scenario for configuration of immaterial products is proposed. The scenario is demonstrated by a particular case of configuration of a mobile operator product in the form of supplement this product with a service. For this case an OWL-based ontology for mobile product operator is developed. The scenario is executed through series of SPARQL-requests to the ontology.

An ontology-based product model supporting the scenario provides with the domain semantics, which facilitates understanding between the customer and the product configurator and allows the customer to think of the product and its components at the conceptual level without going into details. For the product configurators this model enables to take into account changes in the product easily. Newly appeared components simply need to be introduced in the ontology with corresponding properties revisions.

The proposed scenario reduces efforts of human configurators. They do not need to be aware of technical details how to meet various constraints and how to configure product the most suitable for the customer. The scenario supposes two ways of customer participation in the configuration process. The first one does not insists on involvement of the customer; information from the customer profile is used to customize the product. The second way intends to customer interacting. This way allows the customers to build the desired configuration interactively step-by-step due to the possibility of participation of the customer in the configuring process and the possibility of multiple reconfiguring.

The scenario is applicable to configuration of immaterial products that can be described by ontology models limited to unary and binary predicates in terms of logics.

ACKNOWLEDGMENT

The present research was partly supported by the projects funded through grants 15-07-08092, 16-07-0375, and 17-07-00248 of the Russian Foundation for Basic Research, the Project 213 within the research program I.5P of the Presidium of the Russian Academy of Sciences, and Grant 074-U01 of the Government of Russian Federation.

REFERENCES

[1] D.Sabin and R.Weigel, "Product Configuration Frameworks – A Survey", IEEE Intelligent Systems, vol. 13, no. 4, 1998, pp. 42–49.

[2] L.Hvam, N.H.Mortensen, and J.Riis Product customization. Berlin, Heidelberg: Springer-Verlag, 2008.

[3] V.E.Barker, D.E.O'Connor, J.Bachant, and E.Soloway, "Expert systems for configuration at Digital: XCON and beyond", Communications of the ACM, vol. 32, no. 3, 1989, pp. 298–318, DOI: 10.1145/62065.62067.

[4] S.Mittal and F.Frayman, "Dynamic constraint satisfaction problems", in Proc. of the eighth National Conf. on Artificial Intelligence AAAI-90, July-Aug., 1990, pp. 25–32.

[5] H.Xie, P.Henderson, and M.Kernahan, "Modelling and solving engineering product configuration problems by constraint satisfaction", International Journal of Production Research, vol. 43, no. 20, 2005, pp. 4455–4469.

[6] L.Wang, W.K.Ng, and B.Song "Constraint Satisfaction Approach on Product Configuration with Cost Estimation", Next-Generation Applied Intelligence. LNCS, vol. 5579, 2009, pp. 731–740.

[7] T.Soininen, J.Tiihonen, T.Männistö, and R.Sulonen, "Towards a general ontology of configuration", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 1998, vol. 12, no. 4, pp. 357–372, DOI: 10.1017/S0890060498124083.

[8] A.Felfernig, G.Friedrich, and D.Jannach, "Conceptual modeling for configuration of mass-customizable products", Artificial Intelligence in Engineering, 2001, vol. 15, no. 2, pp. 165–176.

[9] D.Yang, M.Dong, and R.Miao, "Development of a product configuration system with an ontology-based approach", Computer-Aided Design, 2008, vol. 40, is. 8, pp. 863–878, DOI: 10.1016/j.cad.2008.05.004.

[10] H.Lee, "Product configuration strategy based on product family similarity", International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering, 2013, vol.7, no.8, pp. 1709–1713.

[11] S. Xuanyuan, Y. Li, L. Patil, and Z. Jiang, "Configuration semantics representation: A rule-based ontology for product configuration", in Proc. 2016 SAI Computing Conf., IEEE, 2016, pp. 734–741, DOI: 10.1109/SAI.2016.7556062.

[12] B.Aiello, and L.Sachs, Configuration Management Best Practices, Addison-Wesley, 2011.

[13] J.Tiihonen, M.Raatikainen, V.Myllärniemi, and T.Männistö, "Carrying ideas from knowledge-based configuration to software product lines", in Software Reuse: Bridging with Social Awareness; eds. by M.G.Kapitsaki and E. de Almeida, Cham, Switzerland: Springer International Publishing, 2016, pp. 55–62, DOI: 10.1007/978-3-319-35122-3_4.

[14] Y.Zhang, S.-S.Zhang, and S.-Q.Han "Adaptive service configuration approach for quality of service management in ubiquitous computing environments", Journal of Zhejiang University SCIENCE A, 2009, vol. 9, is. 7, pp. 964–975.

[15] J.A.Lakunza, J.C.Astiazaran, and M.Elejoste, "Service model for the service configuration", in Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services; eds. by C.Emmanouilidis, M.Taisch, and D.Kiritsis, Berlin, Heidelberg: Springer, 2013, pp. 430–437, DOI: 10.1007/978-3-642-40361-3_55.

[16] Duck Creek technologies official website, On Demand, 2017, Web: http://www.duckcreek.com/wp-content/uploads/2017/01/Brochure_On-Demand_Jan2017.pdf

[17] W.P.Shao, Y.P.Hao, P.F.Zeng, and Y.X.Liu, "Research on product configuration knowledge representation and sharing based on ontology", in Proc. 4th Int. Conf. Wireless Communications, Networking and Mobile Computing, IEEE, 2008, pp. 1–5, DOI: 10.1109/WiCom.2008.2577.

[18] F.Colace, M.Santo, and P.Napoletano, "Product configurator: an ontological approach", in Proc. Ninth Int. Conf. Intelligent Systems Design and Applications, IEEE, 2009, pp. 908–912, DOI: 10.1109/ISDA.2009.236.

[19] F.Colace, M.Santo, and L.Greco, "An adaptive product configurator based on slow intelligence approach, International Journal of Metadata, Semantics and Ontologies, 2014, vol. 9, no. 2, pp. 128–137, DOI: 10.1504/IJMSO.2014.060340.

[20] H.Wicaksono , V.Schubert , S.Rogalski , Y.Ait Laydi , and J.Ovtcharova, "Ontology-driven requirements elicitation in product configuration systems", in Enabling Manufacturing Competitiveness and Economic Sustainability; ed. by H.A.EIMaraghy, Springer, 2011, pp. 63–67.

[21] M.Dong, D.Yang, and L.Su, "Ontology-based service product configuration system modeling and development", Expert Systems with Applications, 2011, vol. 38. no. 9, pp. 11770–11786.

[22] OWL Web Ontology Language Overview; eds. by D.L. McGuinness, Frank van Harmelen, W3C Recommendation, 10 Feb. 2004, URL: https://www.w3.org/TR/owl-features/.

[23] SPARQL 1.1 Overview; ed. by W3C SPARQL Working Group, W3C Recommendation, 21 Mar. 2013, URL: https://www.w3.org/TR/sparql11-overview/.

[24] W3C official site, URL: https://www.w3.org/Consortium/.

[25] T.Leckner, M.Lacher, "Simplifying configuration through customer oriented product models", in Proc. ICED 03 Conf., Aug. 2003, pp. 237–238.

[26] Information Framework (SID), TM Forum site, URL: https://www.tmforum.org/information-framework-sid/.

[27] Application Framework (TAM). TM Forum site, URL: https://www.tmforum.org/application-framework/