Knowledge-Assisted Product Requirements Configurator

Preethu Rose Anish, Shashi Kant Sharma, Manish Motwani, Smita Ghaisas
Tata Research, Development and Design Center,
54 Hadapsar Industrial Estates, Pune, 411013,
India
{preethu.rose, sharma.shashi, manish.motwani, smita.ghaisas}@tcs.com

***Abstract*—Producing timely and customer oriented products is a key determinant for the success of any product-based business. Product requirement elicitation and configuration practices therefore play a major role in taking products to market efficiently. Knowledge of the existing generic product is crucially important while creating its variants. In this paper, we discuss an ontological representation of product primitives for a knowledge-assisted requirements configurator and illustrate its use for a financial product suite.**

***Index Terms*—Requirements Configurator, Ontology, Product Line, Financial Product Suite, Configuring products, Knowledge-assisted configuration.**

I. Introduction

Product configuration is the process of generating a variant from a previously defined product family model and additional product specifications for a given variant [10]. Product knowledge plays an important role in identifying the requirements of the desired variant and configuring the existing product [1] [2] [9]. The success of a product based business depends to a great extent on how efficiently the existing product knowledge is utilized for customization needs. The knowledge is often not represented in a form that allows an easy reuse however. The gap analysis done in the absence of adequate product knowledge is prone to errors and omissions. Also, since the knowledge resides tacitly with only a few experts; teams depend heavily on these people. Not only does this present a big risk associated with loss of knowledge if these people leave the organization; it also makes the whole exercise of requirements gap analysis subjective. In the absence of mechanisms that make product knowledge explicit and accessible, we may have a suboptimal reuse of existing product features. This can result in an unnecessary effort overhead of redeveloping features or parts thereof. These concerns are representative of today's business scenario wherein there is a progressive shift from the traditional one-size-fits-all products to more customer-specific products [5-8]. To identify the requirements for the variants accurately, we need to have a thorough knowledge of the existing generic product features. We can then use this knowledge to compare the product features with customer-specific requirements, perform a gap analysis and use the analysis in customization of the product.

In this paper, we discuss an ontology-based framework to represent product knowledge and reuse it in configuration exercises for requirements gap analysis. Our previous work [3, 4] on knowledge-assisted requirements framework facilitates reuse of requirements knowledge in typical service-oriented developments. We build upon this work to bring in the products perspective and to cater to the needs of product-oriented business.

The rest of the paper is organized as follows: section II details the solution approach and usage illustration. Section III presents discussion and conclusion.

II. Ontology Based Product Requirements Configurator

In this section we detail the ontology-based product requirements configurator and illustrate its usage in a large financial product suite.

A. The Ontological Model

The ontologies are organized along three distinct contexts: (1) Product Knowledge (2) Environment (3) Business domain. The Environmental Context Ontology and the Business Domain Ontology are reused from our previous work [3, 4] whereas the Product Knowledge Ontology is a new addition.

The product knowledge ontology represents how a financial product is organized. It captures the product primitives in terms of concepts such as Features, Risks and Covers, Attributes, and Rules. The product knowledge is expressed and organized in terms of the instances of these concepts and their inter-relations.

The environmental context captures the environmental parameters such as Domain (e.g. *Insurance*), Business (e.g. *Non-life*), Line of Business (e.g. *Auto*), Product Line (e.g. *Personal*), Country (e.g. *India*) and Company (e.g. *ABC Inc*.). These are abstractions that let one organize knowledge of the environment in which the product is to be deployed. This is important as customers may need to conform to different laws of land and may have different organizational policies. Thus, selection of the environmental parameters ensures that only the relevant product specification is made available to the business analyst. This additionally ensures that the analyst's view of product specification is not cluttered with any irrelevant information.

The Business domain context captures the essence of the domain such as Insurance and therefore contains business concepts, their relationships and constraints. For example,

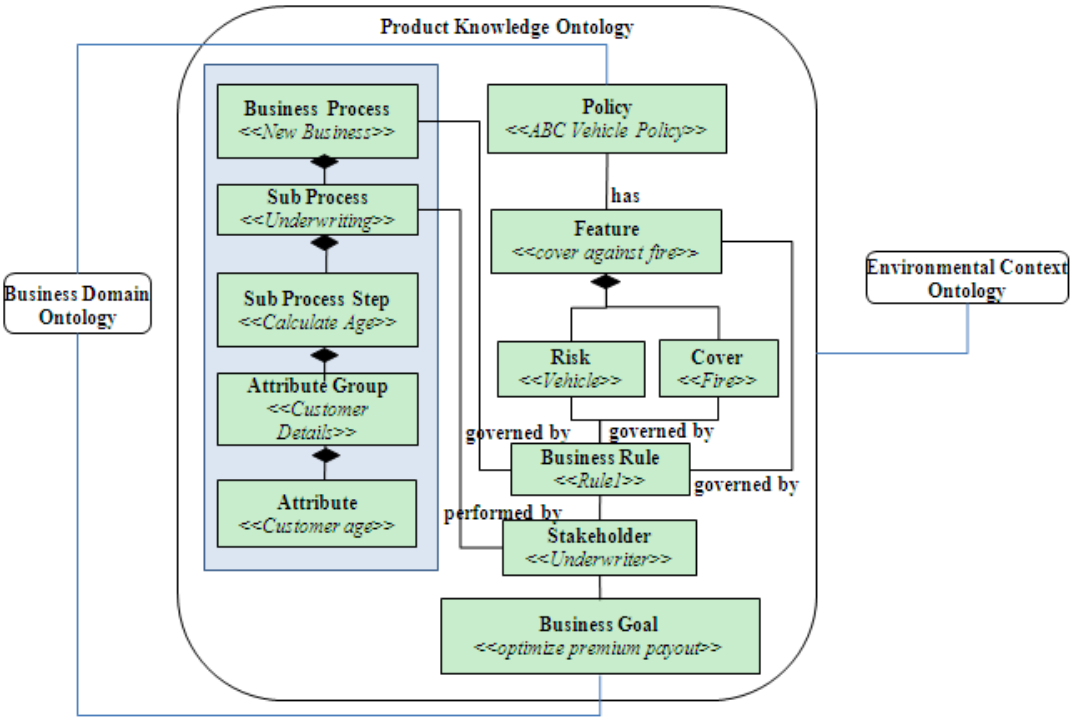

Fig. 1. Product Knowledge Ontology

consider the following scenario from insurance domain – *For vehicle claims, in the event of fire; the policyholder may submit a claim request*. The abstractions such as **BusinessEvent (***fire***)**, **Party (***Policyholder***)**, and **BusinessAction (***submit a claim request***)** are used to capture this information.

Figure 1 shows the partial Product Knowledge Ontology. The technical details pertinent to the semantic guidance and interactions amongst the ontologies are explained in detail in our previous work [3, 4]. The working of the configurator is illustrated next.

B. Knowledge –assisted Product Requirements Configuration and Gap-analysis –an Illustration

The first step in any product customization exercise is to capture and understand the customer requirements. This is done in one of the following ways: (1) Show and tell technique wherein the product vendor shows the existing features in the product suite to the customer. The customer then suggests the customizations needed (if any) (2) the customer creates their own requirements documents and shares it with the vendor's business analyst team. (3) The customer representative(s) verbally shares the requirements and the business analysts create the requirements specification document accordingly.

We illustrate the first approach (show and tell technique) in this paper. For the purpose of illustration, we use an example from auto insurance. It should be noted that ontology based repository for product knowledge and domain knowledge is created and validated with experts. In this paper we do not discuss the process of creating the knowledge repository using the ontological framework. The reader is referred to our previous work [3, 4] for details of the process.

Step 1: The requirement analyst captures the customer-specific parameters such as ***Domain*** (e.g. *Insurance*), ***Business*** (e.g. *Non-life*), ***Line of Business*** (e.g. *Auto*), ***Product Line*** (e.g. *Personal*), ***Country*** (e.g. *India*) and ***Company*** (e.g. *ABC Inc*.).

Step 2: The product configurator presents a suitable variant based on the parameters, if already available in the repository from previously executed projects that match the selected parameters. Else, a generic specification is presented. The customer representative and the analyst review this together. Table 1 shows a (partial) generic product specification.

Step 3: The customer suggests additions / modifications as per their needs.

Step4: The analyst begins the process of configuring the generic product specification in the product suite to incorporate the customer's inputs from *Step 3*. At this stage, the configurator creates a copy of the generic product specification and makes it available in the workspace designated for the requirement analyst. The original generic specification in the knowledge repository is not altered or overwritten.

Step5: The configurator provides recommendations to the business analyst as she incorporates customer's inputs to create a variant of the product. The recommendations are facilitated by the underlying ontologies and the semantic rules that operate on it.

For instance, when the representative adds a new ***Cover*** (*Third Party Liability)* upon suggestion by the customer, the configurator alerts her that the feature that she has added already exists in some other configured product (e.g. *Product A*) in the product suite. She is recommended to look up the associated artifacts with this Cover (*Third Party Liability)* in *Product A* such as the associated ***Rules*** (*A claim against*

vehicle fire should be intimated within 1 week of its occurrence), ***Processes*** (*Fire Claim Process*), ***Sub processes*** (*Fire claim Registration*), ***Attributes*** (*Policy Id, Vehicle no.)* etc. When she selects the suggested rule (*A claim against vehicle fire should be intimated within 1 week of its occurrence*), she receives an alert to verify if this rule contradicts with some other rule (*all claims must be intimated within 30 days of its occurrence)* that already exists in the generic product specification**.** Similarly, if she modifies a ***Sub Process (***e.g*. Fire claim Registration),* she is alerted about the impacted artifacts such as ***Sub process step*** (*e.g. capture claim details*), ***Attribute Group*** (e.g. *details for Claim intimation*), ***Attribute*** (e.g. *cause of fire*)

TABLE 1: PARTIAL GENERIC PRODUCT SPECIFICATION

Generic Product Specification (Partial)

Covers

Bodily injury/ Property Damage Liability

Attributes

For Fire claim registration sub process

1. Vehicle Colour
2. Claim Details
3. Damage details
4. Injury Details

Attribute Format for Policy Number: free text

Process

Premium Collection

1. Accepts premium in local currency.
2. The premium depends on the following factors of the vehicle
    1. The make, model, year of manufacture, cubic capacity

Policy Transfer

Policy transfer can be done only in case of

1. Death of the insured

Payment:

a. Payment mode can be through
    1. Check
    2. Cash

.

Step 6: Upon completing the customizations as detailed above, the analyst can produce a gap analysis report from the configurator. The report highlights the 'evolution' of the product specification from its generic version to its specific variant. The configurator also stores the country specific version in its knowledge repository.

Table 2 and 3 show representative (partial) variants for *India* and *Middle East* respectively. These variants are derived from the generic product specification (Table 1). For a better readability, we have highlighted the variations as follows: the new additions in the variants are highlighted in 'bold' and the details that are not reused in the variants from the generic specification are 'stricken off".

TABLE 2: PARTIAL SPECIFICATION OF A PRODUCT VARIANT FOR INDIA

India Specific Variant of Product Specification (Partial)

Covers

1. Bodily injury/ Property Damage Liability
2. **Third Party Liability**

Attributes

For Fire claim registration sub process

1. ~~*Vehicle Colour*~~
2. Claim Details
3. Damage details
4. Injury Details

Attribute Format for Policy Number: **Office Code (6 digit) + Dept Code (2 digit) + Year (2 digit) + 8 digit Sequence No.**

Process

Premium Collection

1. Accepts premium in local currency **of India - Rupees.**
2. The premium depends on the following factors of the vehicle
    1. **IDV**
    2. **GVW/Cubic Capacity as applicable**

Policy Transfer

Policy transfer can be done in case of

1. Death of the insured.
2. **Vehicle sale**

Payment

a. Payment mode can be through
    1. Check
    2. Cash
    3. **Bank transfer**

TABLE 3: PARTIAL SPECIFICATION OF A PRODUCT VARIANT FOR A COUNTRY IN THE MIDDLE EAST

Country in Middle East Specific Variant of Product Specification (Partial)

Covers

1. Bodily injury/ Property Damage Liability
2. **Third Party Liability**
3. **Riot, Strike, Storm and flood**

Attributes

For Fire claim registration sub process

1. Vehicle Colour
2. Claim Details
3. Damage details
4. Injury Details

Attribute Format for Policy Number: **Policy number prefixed with HPM followed by the year in 'YY' Format followed by seq. no**.

Process

Premium Collection

1. Accepts premium in **any currency and converts them to BHD**
2. The premium depends on the following factors of the vehicle
    1.The make, model, year of manufacture
    2.Driving record, claims experience of the Insured

Policy Transfer

Policy transfer can be done in case of

1. Death of the insured.
2. **Vehicle sale except for fleet policy**

Payment

a. Payment mode can be through
    1. Check
    2. Cash
    3. **Credit Card**

III. Discussion and Conclusion

Work in the area of ontology-based frameworks for capturing and analyzing customer requirements has been reported in literature [1, 11- 14]. The ontological models detailed in these works are specific to the *Manufacturing domain* or the *Production domain*. The product primitives of the Finance domain are however very different from those in a Manufacturing or the Production domain. To the best of our knowledge, no work has been reported on ontology-based knowledge-assisted product requirements configuration in the Finance domain.

We present an approach for a knowledge-assisted configuration of product requirements for Finance domain. We capture the product primitives in the form of an ontological model. We associate various concepts in the Business Domain and Environmental Context ontologies with those in the Product Knowledge Ontology and build inference mechanisms based on this foundation. The inference mechanisms are used to provide knowledge assistance while configuring a product requirement set. Every configuration exercise involves creating an instance of the ontological model comprising of business domain, environment and product requirements primitives using *just-in-time* alerts. We illustrate the knowledge-assisted requirements configuration using an example from a large financial product suite**.**

The ready- to- review documents such as the product variant specification and gap analysis report generated from the configurator save time and effort on review cycles and also make it more comprehensive. In contrast, reviewing product features manually is not only inefficient and cumbersome but also error-prone. The configurator assists business analysts in correctly analyzing the gaps that exist between the customer requirements and the generic product. This reduces the chances of redeveloping already existing features. Also**,** by making the tacit knowledge explicit, the configurator reduces the excessive dependence on a small number of experts. The variant–specific models can be exported into model-based tools for the downstream activities in forward engineering (e.g. model based code-generation). The approach makes knowledge explicit and helps preserve it within the organization, so that it is available and scalable; unlike a handful of experts who may not be available for every product configuration exercise in projects across the organization.

An inherent and obvious limitation of the approach is that it requires an upfront creation of ontology-based knowledge repository. This can be a potential hindrance to its adoption. The configurator is presently in use by the financial products team. We are currently gathering feedback from the users and we would publish our evaluation results in a suitable forum in the near future.

References

[1] Wicaksono H,Schubert V, Rogalski S, Ait Laydi Y, Ovtcharova J, *Ontology-driven requirements elicitation in product configuration systems*, 4th International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2011), Montreal, Canada 2011

[2] J R Jiao, C.H.Chen, *Customer requirement management in product development: A review of research issues*, Concurrent Engineering: Research and Applications, Vol.14, No. 3, 2006

[3] S. Ghaisas and N. Ajmeri, *Knowledge assisted Requirements Evolution*, Accepted in Managing requirements Knowledge, (MaRK) Ed., Springer

[4] Ajmeri, N., Sejpal R., Ghaisas, S.: *A semantic and collaborative platform for agile requirements evolution*, In: Third International Workshop on Managing Requirements Knowledge, pp. 32--40. IEEE Press, Sydney (2010)

[5] A Falkner,I Feinerer, G Salzer, G Schenner, *Solving practical configuration problems using UML*, Workshop on Configuration Systems, ECAI 2008

[6] O Djebbi, C Salinesi, *Towards an automatic PL requirements configuration through constraints reasoning*, Int. Workshop on Variability Modeling of Software Intensive Systems (VaMoS), Essen, Germany, January 2008

[7] R Rabiser, D Dhungana, *Integrated support for product configuration and requirements engineering in product derivation*, 33rd EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2007)

[8] S. Schwarze, *Specification mapping - the integration of customer requirements within a product configuration*, Institute of Industrial Engineering and Management (BWI), Switzerland, Swiss Federal Institute of Technology (ETHZ), 1993.

[9] S Mesihovic, J Malmqvist, *Product data management (PDM) system support for the engineering configuration process*,14th European Conference on Artificial Intelligence ECAI 2000 Configuration Workshop August 20-25, 2000, Berlin, Germany.

[10] Karsten Schierholt (2001). Process configuration: *Combining the principles of product configuration and process planning*. AI EDAM, 15, pp 411-424

[11] Yang, D., Dong, M., Miao, R., 2008, *Development of a product configuration system with an ontology-based approach* Computer-Aided Design, 40:863-878

[12] Kim KY, Manley DG, Yang H. *Ontology-based assembly design and information sharing for collaborative product development*. Computer-Aided Design 2006; 38(12):1233–50

[13] Zhang JS, Wang QF, Wan L, Zhong YF. *Product configuration modeling based on ontology*. Computer Integrated Manufacturing Systems 2003;9(5):344–50

[14] Soininen T, Tiihonen J, Mannisto T, Sulonen R. *Towards a general ontology of configuration*. Artificial Intelligence for Engineering, Design, Analysis and Manufacturing (AI EDAM)1998;12(4):357–72