ONTOLOGICAL CONFIGURATOR
A Novel Approach

F. Clarizia, F. Colace and M. de Santo
DIIIE-Università degli Studi di Salerno, Via Ponte Don Melillo, 1 84084 Fisciano (Sa), Italy

Keywords: Ontology, Product Configurator.

Abstract: The ability to create customized product configurations which satisfies a user' needs is one of the major aims desired by companies. In particular, the design and implementation of web-based services that allows users to create and customize in a simple and intuitive way the desired product. One research area that has lacked progress is the definition of a common vocabulary that enables consumer-to-manufacturer and manufacturer-to-manufacturer communication. Enabling this communication opens possibilities such as the ability to express customer requirements correctly and to exchange knowledge between manufacturers. A popular approach to express knowledge uses ontological formalisms. Using ontology, a vocabulary can be defined that allows interested parties to specify and share common knowledge and serves to define a framework for the representation of the knowledge. With this aim, this paper presents an ontology-based configurator system that finds the configuration that maximizes the user's needs starting from the desired requirements, the available components, the context information and previous similar configurations. The process by which the system finds the candidate configuration follows an approach known as "Slow Intelligence". This paper presents in detail the proposed approach and presents its first application.

1 INTRODUCTION

In the past, companies relied on standard configurations that reduced production costs while increasing profits. However, as new manufactures join the market, existing ones must find ways to attract new or retain existing customers. One method that is gaining popularity is the ability to personalize products to a customer's need. When users are able to obtain a product that meet their needs as opposed to most closely meeting their needs, their perception of the manufacturer increases. This increase in perception allows manufacturers to gain repeat customers resulting in an increase of their profits. However, the ability to generate a customized product that meets a user's needs is still challenging for most companies. Customers ask for personalized products prompting companies to consider mass customization. Mass customization brings a change in how product design is organized affecting the cost-efficiency of mass production. Many researchers believe that product configuration is an effective answer to this problem (Ding, 2008). In this scenario, a customer is able to experiment any kind of product configuration, which is controlled from the point of view of its technical feasibility and cost. This approach is effective not only for products but also in the world of services: in fact, customer often requests a complex service starting from a portfolio of base services (insurance services, travel services and so on). Therefore, the product configurator is becoming the killer application for the management of companies' business processes: In fact, the customer's order drives the full chain of processes (Felfernig, 2002). The configurator, however, has also to furnish the best solution for the customer and check the actual feasibility of the product. In other words, a good configurator aims to be the perfect seller being able to satisfy customers' needs 24 hours per day. Product development based on customer preferences is key to obtaining larger market share and faster sales growth for organizations. The Internet has fostered the creation of e-businesses and the building of a real, interesting and distributed market. Many corporations have introduced on their websites an interactive product configurator allowing customer to experiment with options and be supported in the product selection process (Franke, 2003). Even though this issue represents a very challenging and interesting research topic, there is a lack of research activities. Current configurator implementations include

Clarizia F., Colace F. and de Santo M. (2010).
ONTOLOGICAL CONFIGURATOR - A Novel Approach.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 515-520
DOI: 10.5220/0002976505150520

simple rule-based system, a hierarchical questionnaire or a passive system that collects with no logic, the preferences of the customer. In these cases, the product configurator is just a product viewer for the customer and does not implement any reasoning logic or user adaptive approach. Customers' requests are passively received without further reasoning requiring the customer to be familiar with both the product structure and its functions and so often the final configuration is not the best for him. Recently, there have been some proposals in literature (Ding, 2008) (Park, 2008) (Youliang, 2007) for the introduction of an intelligent configurator. They propose an ontology based approach which seems to be an effective methodology for the improvement of the actual configurator (Yang, 2008). In this paper, a framework for the assisted product configuration, based on the use of ontology formalism and methodologies, is proposed. It works mainly by the use of four different ontologies: the functionality ontology obtained by the analysis of the user's request, the component's ontology obtained by the support of an expert, the customer's context approach and the ontology of previous configuration. In particular, the proposed configurator follows a Slow Intelligent System's model. These models are general-purpose systems characterized by being able to improve their knowledge over time using the working context, expert knowledge and task methodologies. A Slow Intelligent System is one that given a particular task is able to reason and provide an answer after completing a process of enumeration, elimination and concentration and continuously learns, searches for new knowledge and shares experience with other peers. In a Slow Intelligent System, the information is represented by the use of ontology formalism. In the following paragraphs, this approach will be detailed. This paper follows this structure: the next section describes the Slow Intelligent model approach. The third section explains in details the proposed configurator and a working example is furnished. Finally, conclusions and future works are described.

2 SLOW INTELLIGENCE SYSTEM

In this section we introduce and develop a general framework named Slow Intelligence (SI) Systems (Chang, 2010). We view SI systems as general-purpose systems characterized by being able to improve their knowledge over time using the working context, expert knowledge and task methodologies. A SI System is one that given a particular task is able to reason and provide an answer after completing a process of enumeration, elimination and concentration. Such systems are able to improve their knowledge over time using expert knowledge: a SI Systems continuously learns, searches for new knowledge and shares experience with other peers. After acquiring new knowledge, the system may answer the same task in a different way and with different results. In particular, the proposed system follows two decision cycles. The first one, defined as a short decision cycle, provides an instantaneous response to the environment. The second one, a long decision cycle, tries to follow the gradual changes in the environment and analyze the information acquired by experts and past experiences. In this way, the long decision cycle can influence the short one improving the reliability of the system. Therefore, SI Systems work in synergy with the environment and are usually correct but not always fast. A SI System differs from expert systems in that the learning is not obvious. A SI System seems to be a slow learner because it analyzes the environmental changes, and carefully absorbs that into its knowledge base maintaining synergy with the environment. In general, a SI System acts according to five main phases:

- Enumeration: in this phase a SI System enumerates all the possible methodologies for the resolution of a task
- Adaptation: in this phase a SI System acquires information on the context where it is working
- Elimination: In this phase, a SI System, according to the information acquired in the previous phases, selects the best methodology to approach and solve a task. Information acquired from experts as well as learned experiences are used.
- Concentration: After the selection of the best methodology for solving a task, a SI System concentrates its resources in solving the problem.
- Communication: After the resolution of a task, a SI System updates its experience and shares the new information with other peers.

3 A PRODUCT CONFIGURATOR BASED ON THE SIS APPROACH

In this paragraph the design of the ontological configurator, according to the Slow Intelligence Approach, will be described. First of all some general definition on the configuration problem will be introduced.

Definition 1: The configuration problem (CP) is formulated as: CP = {C, P, C_r, R} where C is a set of components that may constitute a customizable product, P is a set of properties of components, C_r is a set of constraints imposed on components due to technical and economical factors and R is a set of customer requirements which are usually specified in the forms of constraints. The CP collects the user's request D_{UR}

Definition 2: The configuration solution (CS) is defined as: CS = {I, V} where I is a set of individuals which are instances of components, V a set of values which are assigned to properties of individuals. The CS is expressed by the description of the customized product D_{CP}

Definition 3: The ontology of the functionalities (O_F) defined as {C_F, A_F, H_F, R_{TF}, R_F, Ax_F} where C_F is the concept set. $c \in C_F$ expresses one concept and in each ontology there is ever a root concept marked as "Thing". The aim of this ontology is the representation of the product's functionalities requested by the user.

Definition 4: The ontology of components (O_{CP}) defined as {C_{CP}, A_{CP}, H_{CP}, R_{TCP}, R_{CP}, Ax_F} where C_{CP} is the concept set. $c \in C_{CP}$ expresses one concept and in each ontology there is ever a root concept marked as "Thing". The aim of this ontology is the representation of the product's functionalities requested by the user.

Definition 5: The ontology builder module. This module has the aim to build an ontology starting from some inputs furnished by users or obtained by the environment. In particular this module has inside a meta-ontology representing in a very general way the expected ontology. The inputs represent the nodes and the related relations that have to be considered in this meta-ontology in order to obtain the desired ontology. If some inputs are not represented in the meta-ontology the user will define the new nodes, the attributes and the relations with the other nodes of the meta-ontology.

Definition 6: The ontologies' comparison, simplifier and merging module. This module has the aim to manipulate the input ontologies in order to obtain as output an ontology representing them. In particular the following operations can be accomplished:

- Merging: The merging operation is so defined: $M : OxO \rightarrow O$ This operation has the aim to merge two ontologies.
- Simplifying: The simplifying operation is so defined: $S : OxO \rightarrow O$ This operation has the aim to simplify an ontology erasing nodes, relations or attributes. This operation involves the use of another ontology containing the information that will simplify the first one.
- Comparing: The comparing operation is so defined: $C : OxO \rightarrow R$ This operation has the aim to compare two ontologies giving a positive or negative grade which is function of the number of similar nodes, relations and attributes that are between the ontologies.
- Selection: The selection operation is so defined: $Sel : OxOxT \rightarrow O$ This operation has the aim to select an ontology according to the grade $t \in T$ obtained by the use of the comparing function.
- Deletion Node: The deletion node operation is so defined: $DN : OxO \rightarrow O$ The operation DN(O_A, O_B) delete all the nodes in O_A that are in O_B
- Deletion Relation: The deletion relation operation is so defined: $RN : OxR \rightarrow O$ The operation DN(O_A, R_i) delete all the relations R_i among the node of O_A
- Set Attribute: The set attribute operation is so defined: $SA : OxO \rightarrow O$ The operation SA(O_A, O_B) set the attributes for each node C_i which belongs both to O_A and O_B to the value expressed in O_B

4 THE PROPOSED APPROACH

The proposed approach is based on a four-layer modelling architecture as shown in figure 1.

User Layer	User Request
Functionality Layer	Functionalities' Ontology
Component Layer	Components' Ontology
Instance Layer	Configuration Instances

Figure 1: Four-Layer modelling architecture.

The first layer expresses the requests furnished by the user. The requests are collected by the use both of graphic user interfaces both of text areas. The second layer maps the collected requests in the functionalities that the product has to show. The product's functionalities are inferred by the analysis of the user's requests and are expressed by the use of the ontological formalism. The third layer has the aim to select the components achieving the requested functionalities and also in this case for their representation the ontological formalism is adopted. The last layer translates the components' ontology in the final configuration of the product. In this case the output is an XML file that can be easily managed in order to build the customized product. In order to implement the previous layered architecture the Slow Intelligent Approach is adopted. The framework is depicted in figure 2.

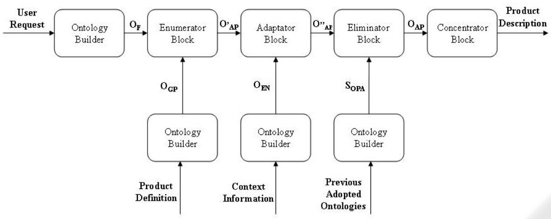

Figure 2: The System Architecture.

This system is compatible with the Slow Intelligent approach and in particular the main blocks can be so identified:

- Enumerator Block: this block has as inputs the user request and the product definition and as output the ontology O'_{AP} and has as components two ontology builders and a comparison, merging and simplifier modules. This block can be characterized by the following function: $F_{EN} : O \times O \rightarrow O$ At the end of this block the ontology representing a first rough version of the customized product is obtained
- Adaptor Block: this block has as inputs the ontology O'_{AP}, representing a first rough version of the customized product, and the information about the context and as output the ontology O''_{AP} and has as components an ontology builder and a comparison, merging and simplifier modules. This block can be characterized by the following function: $F_A : O \times O \rightarrow O$ At the end of this block the ontology representing a context adapted product is obtained
- Eliminator Block: this block has as inputs the ontology O''_{AP} and the previous product's ontologies developed both in the past both by other similar configurators that work in other part of the system. The aim of this block is the tuning of the O''_{AP} according to the previous configurations obtained in similar contexts. This block can be characterized by the following function: $F_{EL} : OxO^N \rightarrow O$ The output of this block is the ontology O_{AP} which represents the adapted product.
- Concentrator Block: this block has as inputs the ontology O_{AP} and as output the configuration of the product. The aim of this block is the mapping in an XML file of the ontology representing the customized product. his block can be characterized by the following function: $F_C : O \rightarrow D_{CP}$

In this way the configuration problem CP can be formulated in its general formulation as: $F_C(F_{EL}(F_A(F_{EN}(U_R, U_P))))$. As previously said a Slow Intelligent System, as the proposed configurator, can follow a slow and a fast process of solution's inference. So the previous formulation can be defined as the slow process, while the fast process $F_C(F_{EN}(U_R, U_P))$. In the next paragraph the detailed description of each previous introduced block will be explained.

The Enumerator Block: this block has the following aims:

- To collect the user's request and transform it in the ontology O_F
- To collect the product's definition and transform it in the ontology O_{GP}
- to map the ontology O_F in the ontology O_{GP} in order to obtain the ontology O'_{AP} of the user's desired product

In particular the user's request is collected by the use of graphic user interface or by the introduction of sentences explaining the functionalities requested to the product. This information is managed by an ontology builder that extracts concepts and relations organizing them in an ontology which expresses the desired functionalities of the product. In the same way an ontology representing the product's components and their relations is developed. In this case the input is the general configuration of the product, the components and their attributes. The final result is an ontology O_{GP} that represents the general configuration of the product. The module Comparison Merging Simplifier has the role to match the functionalities ontology components with the components ontology. At the end of this phase an ontology O'_{AP} representing a first answer to the user request.

The Adaptor Block: this block has the following aims:

- To collect the environment information and transform them in the ontology O_{EN}
- To adapt the ontology O'_{AP} in the ontology O''_{AP} by the use of the ontology O_{EN}

The Eliminator Block: this block has the following aims:

- To compare the ontology O''_{AP} with the other ones that are previously obtained and proposed to the customers. In this phase a comparison with the results obtained by other similar modules that work in different scenarios will be conducted
- To define, after the comparisons, the ontology O_{AP}

The Concentrator Block: this block has the following aims:

- To implement the product defined by the ontology O_{AP} selecting the appropriate components

In order to show how the system works an example will be described. In particular the configuration's problem of a personal computer will be faced. The first step is the introduction of the ontology which defines the functionalities and the components related to a personal computer. This ontology, namely $O_{GP} = O_{GF} \cup O_{GC}$, is defined by experts and contains both the functionalities both the components that can realize them. The user request, as previously said collected by the use of a graphic user interface, will be transformed in an ontology O_F. This ontology expresses the requests of the customer in terms of the main functionalities that the final product has to show. At this point the enumerator block works on the previous ontologies in this way: first of all the simplifying function $S(O_{GP}, O_F)$ is applied. In this way an ontology O_{UFC} containing both the functionalities both the needed components. The ontology O'_{AP} is obtained by the use of the function DN(DR(O_{UFC}, "is_realized"), O_F). In this way the function of the block Enumerator is the following $O'_{AP} = F_{EN}(O_{GP}, O_F) =$ DN(DR(S(O_{GP}, O_F), "is_realized"), O_F). This ontology is the input of the adaptator block. In this case the O'_{AP} has to be compared with the ontology representing the information obtained by the environment O_{EN}. This ontology can be obtained or by the analysis of the environment where the customer lives or the product will be used or by the support of experts. The adaptor block works in the following way: first of all the function merging is invoked: $M(O'_{AP}, O_{EN})$. The ontology O''_{AP} is obtained by the use of SA fuction: $O''_{AP} = SA(M(O'_{AP}, O_{EN}), O_{EN})$. In this way the context adaptation of the ontology representing the product's components is obtained. So the function F_A is so defined: $O''_{AP} = F_A(O'_{AP}, O_{EN}) = SA(M(O'_{AP}, O_{EN}), O_{EN})$. The ontology O''_{AP} is the input of the Eliminator Block which has the aim to compare the obtained components' ontology with the other ones previously developed by other systems that are working in the distributed system. In particular the selection function will be used: a threshold will be fixed and $\forall O_j \in$ Previous Ontology Set $O_{AP} = Sel(O''_{AP}, O_J)$. So the function F_A is so defined: $O_{AP} = F_A(O''_{AP}$, Set_Previous_Ontology$) = Sel(O''_{AP}, O_J)$ $\forall O_j \in$ Set_Previous_Ontology. At this point the obtained ontology will be elaborated by the concentrator in order to obtain the description, in a XML format, of the components' list needed for the realization of the product. So the function F_C is a parser from the OWL language to the XML file required for the assembly of the product: $D_P = F_C(O_{AP})$. In order to better explain the proposed methodology an example will be introduced. A customer would like to buy a Personal Computer in order to play videogames and surf on internet. He knows that he needs an operative system, a browser web and an antivirus. In particular he prefers a Microsoft Windows family operative system. He is in the United States and prefers to have a desktop. He prefers, besides, cheaper components. This user by the use of a graphic user interface can easily define the ontology O_F that contains the functionalities the he needs. Starting from the general ontology O_{GP} containing all the functionalities and the components involved in a personal computer. So these ontologies, O_{GP} and O_F, can be used in the function F_E obtaining the ontology O'_{AP} .At this point the adaptation module has to be used. In particular as previously said this module aims both to introduce new concepts both to set the attributes of the O'_{AP} according to the information obtained by the environment, where the product will work, or experts. In the proposed case, for example, in the proposed configuration there is the node "keyboard" with the following attributes: US layout, Euro layout, CN layout: in this case the attribute US layout will be selected and the other one will be deleted. So after this phase the ontology O''_{AP} is obtained and the eliminator block can work.

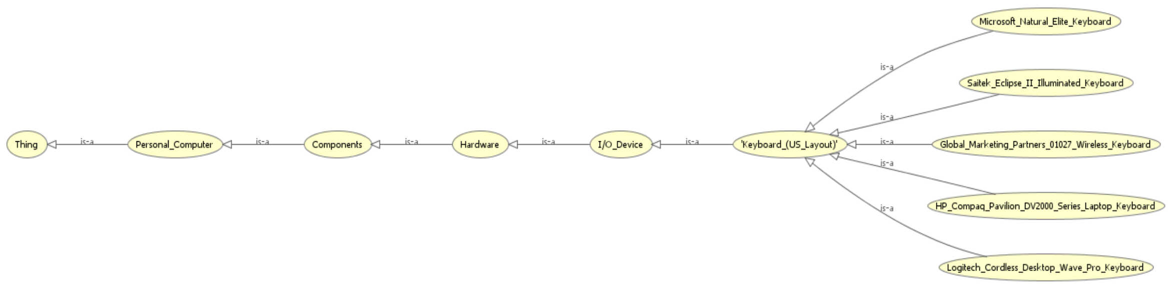

Figure 3: Particular of the OAP ontology related to the component keyboard.

As previously said this module has the aim, by the use of selector function, to compare the ontology O''_{AP} with the other ones developed by other configurators that work in the system or by the same configurator in the past. If for example a previous developed configuration has the same configuration of O''_{AP} but introduces a node representing a component, for example a particular kind of peripheral as a Joystick, for the improvement in the video game playing experience, it will be the new ontology O_{AP}. At the end of this module the concentrator block will define the real configuration by the selection of the real components that are leaves in the ontology O_{AP} (fig. 4). Each node, representing a real component, contains attributes on the main characteristics of the products (price, colour, connection, …)

The selection task will be accomplished according to the information furnished by the user (i.e. select the cheaper components), by the comparison of previous configuration furnished in the past and by the rules codified in the ontology (i.e. an Intel processor needs an Intel motherboard). So at the end the process an xml file containing the final configuration of the personal computer. In this case this file will be showed to the customer by the use of a browser web, so in this way he can check the correctness of the configuration. The same XML file can be used for the real configuration of the product.

5 CONCLUSIONS

In this paper we introduced a Slow Intelligent System approach for the design of a product configurator based on the ontological formalism. In particular first designs of the framework and a first prototype have been developed. In the future a more detailed experimentation of the system will be developed.

REFERENCES

A. Felfernig, G. Friederich, D. Jannach, M. Zanker, “Semantic Configuration Web Services in the CAWICOMS Project”, *6th IEEE International Symposium on Wearable Computers, ISWC 2002*, LNCS 2342, pp. 192-205, 2002

X. Ding, “Product Configuration on the Semantic Web Using Multi-Agent”, *Networking, Sensing and Control, 2008. ICNSC 2008*. IEEE International Conference on, pp. 304 – 309, 2008

N. Franke, F. T. Piller, “Configuration Toolkits for Mass Customization: Setting a Research Agenda”, *International Journal of Entrepeneurship and Innovation Management*, No. 1, 2003

J.M. Park, J.H.Nam, Q.P. Hu, H.W. Suh, “Product Ontology Construction from Engineering Documents”, *Smart Manufacturing Application, 2008. ICSMA 2008*. International Conference on, pp 305 – 310, 2008

H. Youliang, Ng Wee Keong, L. Haifeng, L. Wenfeng, S. Bin, L. Xiang, “Semantic Modeling and Extraction for Cross-family Product Configuration, Service Systems and Service Management, *2007 International Conference on*, pp 1-6, 2007

Dong Yang , Ming Donga, Rui Miaoa, “Development of a product configuration system with an ontology-based approach”, *Computer-Aided Design*, Volume 40, Issue 8, August 2008, Pages 863-878

S.K. Chang, "A general framework for slow intelligence system", *International Journal of Software Engineering and Knowledge Engineering*, 20-1, 2010