

# Knowledge Base Synchronization in Solution Sales Configurator

<sup>1</sup>Ekata R.Mishra, <sup>2</sup>Sachin Bojewar

<sup>1</sup>Post Graduate Student, <sup>2</sup>Assistant Professor

<sup>1</sup>Computer Engineering,

<sup>1</sup> Vidyalankar Institute of Technology, University of Mumbai,, Mumbai, India

**Abstract**—Today’s high-tech and industrial companies want comprehensive solutions comprised of hardware, software, and services designed to meet their unique requirements. To deliver on this demand, the sales force needs a faster, simpler way to bundle components that make up complex solution offerings. However each component in a bundle is a standalone product with its own exhaustive set of features. In such cases, a methodology is required that can help automate and streamline solution selling and quickly deliver targeted solutions that precisely meet customer requirements by only picking those features of the individual standalone products that are uniquely identified and essential to be a part of the bundle. In this paper, we will learn how solution sales configuration can help transform a company into a provider of complex solutions that produce measurable value for customers by encapsulating only those features from the standalone products that are necessary and compatible with the solution bundle as a whole.

**Index Terms**—Solution bundle, orchestration, product modeling, knowledge base, configuration, pfunction, synchronize instances

## I. INTRODUCTION

Manually configuring complex solutions and creating quotes for product bundles can be time consuming and error prone. There is a better way - Solution sales configuration helps sales teams, customers, and channel partners to configure, price, quote, and order complex solutions – accurately and efficiently. This solution can enable your company to:

- Bundle hardware, software, services, and manufactured and financial products.
- Help ensure technical feasibility and commercial efficiency
- Increase customer dependence and loyalty
- Maintain data for complex configuration models and support graphical model definitions
- Optimize system scalability and performance
- Keep pace with the inherent complexity of a solution business

As the product bundle is nothing but a combination of multiple standalone products, one cannot overlook the fact that each of the standalone products configured as part of the bundle would come with its full-fledged features. However, the challenge arises when the customers demand to highly “customize” this bundle by demanding only those features that are essential for their business.

## II. PROBLEM STATEMENT

A bundle comprises of ‘n’ number of individual products depending on the application and the industrial deployment. These individual products are themselves quite complex and extensive. Therefore the features that these individual products offer when they are sold as standalone would be numerous [1] [3]. However, when these individual products are configured with other products as part of a Solution Bundle, it would come with all the features and capabilities it can offer. But customers may not need the products to come with all the features when they are bought with a bundle, as the solution may be specific to a particular business application. Moreover, these extra features or models that get added in the bundle may lead to incorrect pricing that may deter the customer from buying the solution [2]. Therefore, a need arises to enable display and offering only those characteristics of the individual products that are compatible with other products when configured as part of a solution. Nevertheless, the individual product would show up all its features when it is configured standalone. For example, Microsoft word when sold as standalone comes with comprehensive set of features as shown in Fig. 1.

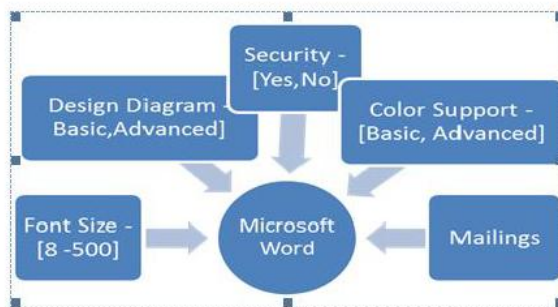


Fig. 1 Microsoft Word Package sold standalone

However, when Microsoft Word is sold as part of personal or a professional suite, the features of Microsoft Word in each of these suites gets tailored as shown in Fig 2

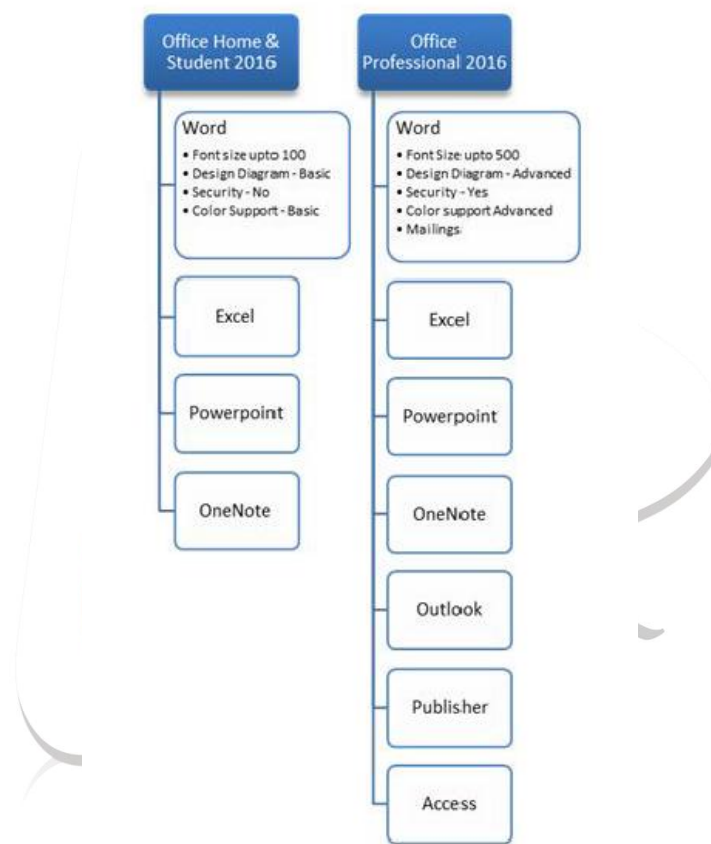


Fig. 2 Personal and Professional Microsoft Office Suite

It is also noteworthy that the Office Home and Student 2016 suite contains only four products viz Word, Excel, PowerPoint and OneNote in contrast to Office Professional 2016 which has an additional three products like Outlook, Publisher, Access. Our task is to determine an efficient method to build such customized bundles that would be beneficial to the end user during configuration and to the selling organization from the configuration management side.

**III. PROBLEM DEFINITION**

In order to build customized configurable packages, there are two ways that the organization “cuts-down” the extra features from the individual products that are not required in the bundle –

- **Allow the extra models and features to be shown or added in the configuration.**  
 This method would allow the end user to add or see the extra or incompatible features. Thus the end user post submitting the quote would see a ‘virtual’ price that would be updated at a later point of time and once the quote is processed downstream, the sales representative would manually remove the incompatible models from the quote and adjust the pricing accordingly [3]. This method is confusing from the end user’s perspective as he is allowed to add incorrect models in the quote, and from the sales and pricing perspective – it is very cumbersome to manually remove the extras.

- **Create different models for the product when configured as standalone and when configured as part of the bundle.** Although this method looks more organized from the customer's perspective, but involves dual development and maintenance of products, inventory, pricing, etc. and cannot be regarded as economical to the company [4].

Therefore, a new method is suggested wherein using few of the development capabilities – we introduce an enhancement in the product modeling stage itself, which can help us to “chop off” the extra features from the individual products when they are sold as part of a bundle and retaining all the exhaustive offerings when they are sold standalone.

#### IV. PROPOSED WORK

In order to overcome the shortcomings of the above existing methods of selling customized products, we have come up with a new approach wherein the logic for customization can be driven at the modeling stage itself. This would involve writing a code in any object oriented language that would enable the individual product to “know” if it is being configured as part of a bundle or standalone. As a result the individual product would demonstrate only the limited essential features when sold with a bundle, however when configured as a standalone, all its available features would be visible. The schema in Fig. 3 explains the approach

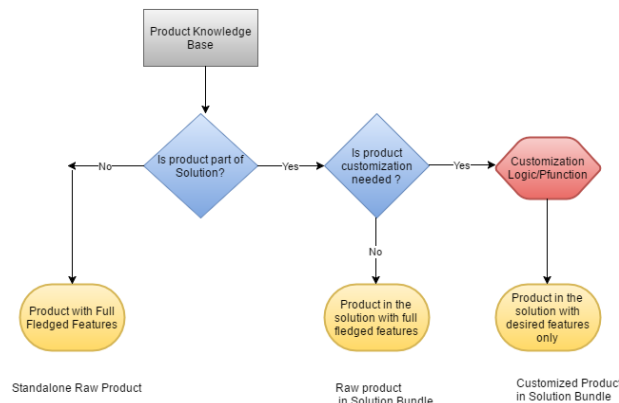


Fig. 3 Proposed Architecture

The logic written in java would drive synchronization between the instances of the Solution and the individual products and thus enable the product to know if it's a part of a bundle or not. This mechanism is termed as ‘Multi Configuration Instantiation’ since the phenomenon may involve instantiating multiple individual products at the same time when configured as part of a Solution with limited features and the tool used is ‘SAP Solution Sales Configurator’.

#### V. IMPLEMENTATION

The concept of Knowledge Base Synchronization has been implementation with the following tools [2]

- Eclipse Luna for RCP and RAP Developers.
- JDK 1.7
- Microsoft SQL Server Management Studio version 10.0.1600.22
- SQL Server 2008 Service Pack3
- Solution Modeling Environment (SME 2.3.14)

The implementation leverages the benefit of Knowledge Base Orchestration[2] fundamental of product modeling which is one of the known survivor of the artificial intelligence boom of the 1980s [4]. The individual products part of Solution can be considered as child Knowledge Bases whereas the Solution Bundle itself is modeled as the parent Knowledge Base. Now using the concept of Knowledge Base Orchestration, we can call each of the product knowledge bases from the Solution Knowledge Base. The child product knowledge bases are fully expanded by default and hence each of the product is shown with its full fledged features. This causes unnecessary selections to be visible which may not be applicable and/or compatible to the application for which the Solution Bundle is being configured. In order to tailor each of the individual product knowledge bases in accordance to the solution bundle it is essential that the solution bundle communicates about the application in which each of the individual products will be utilized. If this information is available at the product level, we can write rules at the product level that would restrict the features of the product in accordance to the application in which it will be used.

This has been achieved by synchronizing the parent and child knowledges bases using a pfunction which consists of a java code that synchronizes the instances across each of the product and solution knowledge bases. We can then model each of the individual product knowledge bases by writing appropriate restriction logic as per application and this tailor the product knowledge base to show only those features that are applicable and/or compatible to the application selected at the Solution Bundle level.

#### VI. CONCLUSION

There has been a considerable increase in the demand of customized products and the trend is rising immensely. As there products become more and more complex with numerous features and being co-sold with other products as part of a Solution Bundle, there arises a need to customize these products to suit the needs of the customer. With each individual product and the bundle being modeled in the SAP Solution Sales Configurator, an interface to Java using pfunction feature of SSC helps us to

synchronize the knowledge bases of individual products and the solution, thus enabling only those features of the products that are compatible with the parent solution. This helps appropriate pricing, easy determination of the finished materials by the sales team and correct assembly process.

## VII. ACKNOWLEDGMENT

We have great pleasure in presenting the paper on Knowledge Base Synchronization in Solution Sales Configurator. We are thankful to Prof. Sachin Deshpande who guided us for publishing the paper through his valuable inputs and mentored us for the whole process, from beginning to the end with full support whenever required. We would thank Principal Dr. Sunil A. Patekar for his moral support. Our family and friends who have given us encouragement and motivation to complete our work.

## REFERENCES

- [1] A. Haag, Product Configuration in SAP: A Retrospective, in Book: Knowledge-based Configuration- From Research to Business Cases, Elsevier/Morgan Kaufmann, pp. 319-337,2014
- [2] <http://www.sap.com/services-support/svc/custom-app-development/>
- [3] Linda Zhang, 'Product configuration: A review of the state-of-the-art and future research',International Journal of Production Research, 2014
- [4] Jiun-Yan Shiau,Nat. Kaohsiung,'Product configuration for engineering change decision',International Journal of Intelligence Science, 2013.
- [5] Kaj A. Jorgensen,'Product Configuration and Product Family Modelling',Networking, Sensing and Control, 2009. ICNSC '09. International Conference, 2009.

