CHAPTER – VII

MODULARIZATION IN LEAN PRODUCT DESIGN

7.1. INTRODUCTION

The aim of this chapter is to demonstrate the concepts of mathematical modeling as applied to modularization product architecture, which is used in Lean Product development for reduction of Lead-time and wastages. The degree of modularization in a given product architecture, which contains the physical components, helps the manufacturer to know the various needs of customers. Physical Modularization refers to the opportunity for mixing-and-matching of components in a modular product design in which the standard interfaces between components are specified to allow for a range of variation in components to be substituted in product architecture. It is through mixing-and-matching of these components, and how these components interface with one another, so that new systems are created. Consequently, the degree of modularization inherent in a system is highly dependent upon the number of components, and these interface constraints shared among the components, modules, and subsystems.

An excel spreadsheet is designed to compute all the above modularization parameters for fast computing and view easier manner than other methods of computation. It is easy to understand and one can explain to customer in “teach by showing method” of various components and interfaces between them.

The organization of this chapter is divided into four sections. Introduction and importance of study of modularization in Lean Product design and relevant literature is presented in section 7.1. Section 7.2 discusses about Levels of modularization and mathematical model of modularization. Section 7.3 of this chapter presents design of excel spreadsheet for the current study with hypothetical case study of product design. Section 7.4 presents brief summary and conclusions of role of modularization with excel spreadsheet in Lean Product development.

7.2 IMPORTANCE OF MODULARITY IN LEAN PRODUCT DESIGN

In New Product Development literature, ‘architecture’ focuses on physical components and their linkages and interactions to other components. For instance, Ulrich (1995) defined product architecture as the scheme by which the function of the
product is allocated to physical components, that is, the arrangement of functional elements, the mapping from functional elements to physical components, and the specification of the interfaces among interacting physical components. Modularity (or modularization) is an approach for organizing complex products and processes efficiently (Baldwin and Clark, 2000), by decomposing complex tasks into simpler portions so they can be managed independently and yet operate together as a whole. A motivation behind decomposition of tasks is to gain flexibility and cost savings through economies of scale. Each product can also be viewed as a system, comprising sub-components whose relationships to each other are also defined by design architecture. Moreover, (Hsuan, 1999) makes the distinction between open- versus close-architecture components. Open-architecture components are not self-contained, but a function of networked parts working together where technological interdependencies shared among these components is crucial (Bi and Zhang, 2001). Close-architecture components are usually supported by a set of open-architecture components in order to achieve full functionality and performance (Juliana, 2001).

From a system’s perspective, modularity can be viewed as a continuum describing the degree to which a system’s components can be separated and recombined, and it refers both to the tightness of coupling between components and the degree to which the “rules” of the system architecture enable (or prohibit) the mixing-and-matching of components (Schilling, 2000). Modularity permits components to be produced separately, or ‘loosely coupled’ (Orton and Weick, 1990; Sanchez and Mahoney, 1996) and used interchangeably in different configurations without compromising system integrity (Garud and Kumaraswamy, 1993). Product architecture is the arrangement of the functional elements of a product into several physical building blocks, including the mapping from functional elements to physical components and the specification of the interfaces among interacting physical components. It is important to note however, that because each component has a specific function modularization at the detailed product architecture level, is not only about mixing-and-matching of these components but also how these components are configured with other components in order to arrive at the desired performance and functionality. The subsequent generation of product variants from common product architecture is inherent in the architecture of the system manifested by the complex linkages among components. In other words, the number of components and interfaces shared among these components determine the elementary complexity of
product architectures. Moreover, if interfaces between components comprising a system become standardized then the complexity of components can be reduced.

In the Product development literature there are suggestions that a link exists between modular product design and improved lead-time. Cusumano (1997) has pointed out that a modular product development strategy makes it possible to decrease lead time since modular product architecture reduces the need for iteration between tasks. Moreover, modular products enable a greater use of parallel product developments. This indicates that the link between modular products and lead-time in product development which is to be found in the differences in the definition and organization of tasks relative to firms to pursue an integral product development strategy. The modular design strategy is used mainly for upgrading the existing product with minimum customer preferences evaluation (Zha, X.F. et al., 2004).

The current market place is unpredictable, where the customers are constantly changing demands. In the current market place, customers can no longer be lumped into a homogeneous group; they expect to be treated as individuals with different needs (Bimal Nepal et al., 2005). The need for mass customization does not only emerge to increase product variety but also provides means for decreasing product development time, which is essential for survival in the current market place (Yuval Sered et al., 2006).

In a competitive environment that is global, intense and dynamic, the development of new products and processes increasingly is a focal point of competition. Firms that get to market faster and more efficiently, create significant competitive leverage. Firms that are slow to market with products to see their market positions erode and financial performance falter (Wheelwright and Clark, 1992).

The new shift in the current market has introduced the concept of mass customization. Consequently, companies are being faced with the challenges of providing as much variety of products as possible for the market with little variations between products. One of the key elements in mass customization is the product platform. A product platform is a collection of the common elements, especially the underlying core technology, implemented across a range of product (Yuval Sered et al., 2006). One way to achieve mass customization is by developing the platform carefully and then using different modules to provide variety. Configuration design
involves determining which modules are in the product, what are the components in the modules, and relationships among the components and modules. A well-defined product platform is required to provide mass customization and also it is much for success in Lean Product design. Modularization is the way to minimize the many types of wastages related to product design. In the view of Lean product design, the brief essays explained about differences between conventional product design and modular product design.

**The Differences between conventional and Modular product design**

The conventional model of the product development process is based on the sequential staging of design tasks. In conventional model, after defining the product concept, design activities are typically sequenced, so that technology and component development activities with the greatest uncertainty are resolved first. As new technical knowledge is developed and technological uncertainties about components are resolved, design decisions are made, thus allowing the next stage of design activities to be implemented. This process is repeated at each stage of the product development process until all the components and their interfaces are fully specified. Although the product development process may begin with a general idea for the arrangement of components in the design, the actual product architecture, i.e., the full specification of all component interfaces, is determined at the end of design process. In essence, the product architecture is the output of the sequential design process.

As new component technology and designs leading to component interface specifications are developed sequentially, the need for changes in component design at an early stage of development may not be discovered until later stages of the design development of dependent components, are reached, as suggested by the information feedback shown in Fig. 7.1. If unexpected technical difficulties encountered in developing “downstream” components indicate a need to change “Upstream” component design, intervening component development process may also have to replaced in order to accommodate changes made in the upstream component designs, especially those affecting major components.
Feedback information flows among stages

Forward information flows among stages

Fig. 7.1 conventional Product design and development (Adapted from Takeuchi et al., 1986)

The natural time delays in this feed system and potential high costs involved in recursively redesigning dependent components when changes must be made in the upstream components may reduce the ability of a company to efficiently create and apply technical knowledge about components.

A modular product design is one in which input and output relationships between components (i.e., component interfaces) in a product have been fully specified and a "standardized". Modular product design implies that there is a new model for managing information flow and knowledge during the product development process as shown in Fig. 7.2. In contrast to the evolving information structures characteristic of the sequential product development process, a modular product design creates a complete information structure fully specified component interfaces of a modular product architecture that defines the desired outputs of development tasks before beginning processes for development and detailed design of components.
7.3 MODULARIZATION IN PRODUCT DEVELOPMENT

Modular product architectures are flexible platforms which enhance the scope for creating a large number of product variations, enabling a firm to gain cost savings through economies of scale from component commonality, inventory and logistics. Some reasons for product change include upgrade, add-ons, adaptation, wear, consumption, flexibility in use, and reuse.

Modular architectures enable firms to minimize the physical changes required to achieve a functional change. Many firms pursue modular product architecture strategies to shorten New Product Development (NPD) lead time, to introduce multiple product models quickly with new product variants at reduced costs, and to introduce many successive versions of the same product line with increased performance levels. Such architectures prove effective in products like computer, automobiles and telephones.

7.3.1 Types of Modular Architecture:

Modular architectures comprise three types: slot, bus and sectional. Each type embodies a one-to-one mapping from functional elements to chunks and well-defined interfaces.
• **Slot modular architecture**: Each of the interfaces between the chunks in slot modular architecture is of a different type from the others, so that the various chunks in the product cannot be interchanged. A automobile player is an example of a chunk in a slot modular architecture. The player implements exactly one function, but its interface is different from any of the other components in the vehicle. For example, the player and the speedometer have different type of interface to the instrument panel.

• **Bus modular architecture**: It has a common bus to which the other chunks connect via the same type of interface. Expansion cords for the personal computers, track lighting embody a bus modular architecture.

• **Sectional modular architecture**: In a sectional modular architecture, all interfaces are of the same type, but there is no single element to which all the other chunks attached. The assembly is built up by connecting chunks to each other via identical interfaces. Piping systems, sectional sofas and office partitions adhere to a sectional modular architecture.

### 7.3.2 Levels of Modularization

The process of modularization in new product development can take place at many different levels: Component Level, Module Level, Sub-System Level, and System Level as shown in Fig. 7.3.

**Component Level**

This is considered the lowest level of modularization, represented by standard, off-the-shelf parts. Specifications of these parts are generally well defined and are accepted as industry standards. Parts suppliers offer a variety of products to many industries, often serving no particular industry. The supplier-buyer partnership at this level often portrays the characteristics of a durable arm’s length relationship.
Module Level

Modules are created by a combination of different parts from the Component Level. The design and manufacturing of modules must keep up with the technological innovation, demands and specification compliance of a particular system. Often the existing modules sold in the market are designed to satisfy some specific sub-system or system specifications, and are rarely universal in nature because they cannot satisfy the technical requirements and demands of all systems, even if they serve the same applications.

Sub system Level

Sub-Systems are often highly customized. There are numerous suppliers producing unique sub-systems dedicated to a particular line of cars. For example, gearboxes installed in buses or passenger cars are incompatible with the ones installed in racing cars or electric vehicles. This implies that interface and protocol compatibility between modules and sub-systems are essential for a system to function.

Sometimes Sub-System Level modularization is bypassed or non-existent where modules are brought together and assembled at the System Level where the degree of supplier-buyer interdependency and interface constraints are much higher.
System Level

Systems are enclosed by subsystems with clear boundaries, and the individual subsystems must be linked together via interface and linkage technologies. Examples of systems include automobiles, airplanes, watches, and televisions. As new components are created at each level, modularization becomes more restricted at the System Level. Opportunities for modularization are significantly reduced as the interface compatibility effects increase. Degree of component customization, degree of value inputs, and degree of supplier-buyer interdependency also tend to be the highest at the system level.

7.3.3 Mathematical Model

Assumptions of Modularization model

1. New Product Design of black box is used, implying that the product’s functional specifications, including interface specifications, do not change over a period of time. This assumption allows the evaluation of the architecture’s configuration and components composition independently from other sub systems.

2. A given product architecture is comprised of a combination of standard components and New to Firm Components.

Notations

\( n_c = \text{Number of Components} \)

\( n_m = \text{Number of Modules} \)

\( n_{ss} = \text{Number of Subsystems} \)

\( k_c = \text{Number of interfaces at Component level} \)

\( k_m = \text{Number of interfaces at Module level} \)

\( k_{ss} = \text{Number of interfaces at Subsystem level} \)

\( w_c = \text{Interface weight at component level} \)

\( w_m = \text{Interface weight at module level} \)
\( \omega_{ss} = \text{Interface weight at subsystem level.} \)

\( \delta_c = \text{Interface constraint factor at component level} \)

\( \delta_m = \text{Interface constraint factor at module level} \)

\( \delta_{ss} = \text{Interface constraint factor at subsystem level} \)

\( \alpha = \text{Supplier development index} \)

The approach of mathematical modeling helps in modeling a given system on a predefined function and hence theoretically examine and simulate the extent to which modularization is present in the system.

Various models can be generated depending upon the function used, which in turn depends on the variables taken in to consideration. The model described here signifies modularization as a function of interface constraints characterized by the number of components and respective interfaces. It also analyses how the supplier buyer relationship influences modularization.

The modularization function, \( f(\delta) \), indicates the opportunities for modularization of a system with respect to the system's interface constraints at different levels. Specifically, the interface constraints variable, \( \delta \), is a function of the number of components and the number of interfaces shared among the components. The opportunity for modularization diminishes in a non-linear fashion from Component Level to System Level as the value of interface constraint increases.

The number of components in the system and its respective interfaces form the characteristics of the function given below. (Hsuan, 1999).

\[
    f(\delta) = e^{-a\delta^2} \quad 0 \leq a \leq 1
\]

Where 
- \( f(\delta) \) – modularization function
- \( \delta \) – interface constraints
- \( a \) – supplier development index
The rate of change of opportunities for modularization is the derivative of the modularization function with respect to the interface constraints. The following relation expresses the relation

\[ f'(\delta) = \frac{df(\delta)}{d\delta} = 2\alpha \delta e^{-\alpha \delta^2} \tag{7.2} \]

One another important aspect derived from the modularization function is the sensitivity of modularization with respect to supplier-buyer index \( \alpha \), or with respect to interface constraint \( \delta \). Such relationship is expressed by the following equations:

\[ S^{f(\delta)}_\alpha = \alpha \frac{df(\delta)}{d\alpha} = -\alpha \delta^2 \tag{7.3} \]

\[ S^{f(\delta)}_\delta = \delta \frac{df(\delta)}{d\delta} = -\alpha \delta^2 \tag{7.4} \]

**Evaluation of supplier Development index (\( \alpha \))**

The characteristic of the supplier development index is that it indicates the extent to which the index can induce a change on the interface constraints and the modularization opportunities. The value of index ranges from 0 to 1. The index with a value of 1 represents the dual arm length relationship and 0 indicates strategic partnership between the supplier and the buyer. Generally, the value of the supplier development index lies somewhere between 0 and 1.

An opportunity for modularization is greatly influenced by the value of the index which in turn indicates the relationship. The success or failure of modularization in new product development is expected to vary depending on the nature of the supplier-buyer partnerships. The durable arm’s length relationship results in standard products, low value inputs and low degree of supplier-buyer interdependence unlike the strategic partnership, which yields customized, non-standard products, high value inputs, and high degree of supplier-buyer interdependence.
Evaluation of Interface of constraint Factor (δ)

Interfaces are linkages among components, modules, subsystems of a given product architecture. The interfaces define the principles for the fundamental interactions across all components and interfaces comprising a technological system. Standardization of these interface principles results in a degree of independence between component designs. Interfaces constraints are restrictions imposed by the components and how interfaces are shared amongst the components in a given product architecture.

A system is divided into subsystems that in turn are divided into modules and then into components. The opportunities for modularization are analyzed at these different stages and each stage of analysis is associated with the interface constraint factor, which depends on the number of components, and the number of interfaces shared.

The factor for each level of analysis is the average of several values of the interface constraint factor as each level involves multiple number of components, modules and subsystems.

The interface weight \( w_c \) at the module level is the ratio of number of interfaces shared by the component to the total number of interfaces shared by all components within the module.

\[
    w_c = \left[ \frac{K_c}{\sum K_c} \right]_n
\]

Let \( \delta_m, \delta_s, \delta_i \) be the interface constraint factor at the module, subsystem and system levels. The interface constraint factor \( \delta_m \) is represented as the product of the number of components in the module and the product of all the interface weights of components in the module.

\[
    \delta_m = n_c \cdot \left[ \prod w_c \right]_m
\]

The interface weight at the sub system level is the ratio of number of interfaces shared by the module to the total number of interfaces shared by all modules in the sub system.

211
The interface constraint factor at the sub system level $\delta_m$ is the sum of interface constraints shared among the modules in addition to the constraints introduced by the components at the modules($\delta_m'$).

$$\delta_m' = \sum[\delta_m]_s$$  \hspace{1cm} (7.8)

$$\delta_{ss} = \delta_m' + n_m \star \prod w_m$$  \hspace{1cm} (7.9)

The interface weight at the system level is the ratio of the number of interfaces shared by the sub system to the total number of interfaces shared by all the subsystems in the system.

$$w_{ss} = \left[ \frac{k_{ss}}{\sum k_{ss}} \right]_s$$  \hspace{1cm} (7.10)

$$\delta_{ss}' = \sum[\delta_{ss}]_s$$  \hspace{1cm} (7.11)

$$\delta_s = \delta_{ss}' + n_{ss} \star \prod w_{ss}$$  \hspace{1cm} (7.12)

The opportunities for modularization are evaluated at all levels, the value of the interface constraints increases at each level from component to system. The average value of $\delta$ is represented as the aggregate interface constraint factor.

7.3.4 Design of Excel Spreadsheet for computing Modularization parameters

The hypothetical system generated consists of two sub-systems with one interface between them. The sub-system 1 consists of two modules and the sub-system 2 consists of three modules. The Fig. 7.4 shows the detailed layout of the interfaces between the components and modules. Excel spread sheet designed for computing various levels interface constraints and modularization values at different stages for reducing lot of repeated data entry trails with other software languages and as shown in Fig. 7.5.
Number of interfaces between components in different modules is presented in columns B to O for component C1 to C14. The module $M_C$ of sub-system 2 is taken for evaluation and it consists of four components (C6, C7, C8 and C9). Consider the component C6 which shares one interface with other components in the module. Let $k_c$ be the number of interfaces shared by the component ($k_c$=1 for C6, 3 for C7, 2 for C8 and 2 for C9) and $w_c$ be the interface weight. The interface weight of component C6 at the module level is given by

$$w_c = \frac{K_c}{\sum K_c} = \frac{1}{8} = 0.1333 \quad (From \ Equation \ 7.5)$$

$$G15=G14/\sum(SG14:J14)$$

Similarly the interface weights for all components of various modules and sub-systems are determined ($w_c$ for components C7, C8 and C9 of module 1 of sub-system 2 are 0.38, 0.25 and 0.25 respectively).

Let $\delta_m$ be the interface constraint factor for the module and $n_c$ be the number of components in the module. The interface constraint factor for module C is given by

$$\delta_m = n_c \times \prod w_c = 4 \times (0.133 \times 0.38 \times 0.25 \times 0.25) = 0.01117 \quad (From \ Equation \ 7.6)$$

We obtained the above said value from spreadsheet as follows

$$G12=COUNT(G14:J14)*PRODUCT(G15:J15)$$

Similarly the interface constraint factor values for all the modules in both the sub-systems are determined.

Let $n_m$ be the number of modules in a sub-system, $w_m$ be the interface weight of the module at the sub-system level, $k_m$ be the number of interfaces shared by the module with other modules in a sub-system. Let us consider module one (M1) of sub-system two which shares two interfaces with other modules.

$$w_m = \frac{k_m}{\sum k_m} = \frac{2}{6} = 0.3333 \quad From \ Equation \ 7.7$$
Let $\delta'_m$, $\delta_n$ be the cumulative value of interface of the module and interface constraint factor of the sub system respectively. Consider sub system 2 and $\delta_{ss}$ is given by

$$\delta'_m = \sum [\delta_m]_{ss} = 0.6117 \quad \text{From Equation 7.8}$$

$$\delta_{ss} = \delta'_m + n_m \cdot \left[ \prod w_m \right]_{ss} = 0.6117 + 3 \times (0.333 \times 0.333 \times 0.3333)$$

$$= 0.7399 \quad \text{From Equation 7.9}$$

The following formula is used for computing subsystem constraint factor

$$G7 = \frac{(\text{SUM}(G12:012) + \text{COUNT}(G10:010) \times \text{PRODUCT}(G11:011))}{\text{SUM}(G10:010)}$$

The interface constraint factor for sub system 1 is determined similarly ($\delta_{ss1} = 1.0938$). Let $w_m$, $n_m$, $k_m$, $\delta'_m$ be the interface weight at the sub system level, number of sub systems, number of interfaces shared by the sub system and cumulative equivalent of the interface constraint factors of the sub systems respectively.

$$w_{ss} = \left[ \frac{k_{ss}}{\sum k_{ss} s} \right] = \frac{1}{2} = 0.5 \quad \text{(From Equation 7.10)}$$

$$\delta'_{ss} = \sum [\delta'_m]_{ss} = 1.0938 + 0.7339 = 1.8277 \quad \text{(From Equation 7.11)}$$

$$\delta_s = \delta'_{ss} + n_{ss} \cdot \left[ \prod w_{ss} \right] = 1.8277 + 2 \times (0.5984 \times 0.4016)$$

$$= 2.3083 \quad \text{(From Equation 7.12)}$$

The following formula is used for computing system interface constraint in Spreadsheet.

$$B4 = \text{SUM} (B7:O7) + \text{COUNT} (B6:O6) \times \text{PRODUCT} (B8:O8)$$
Fig. 7.4 Hypothetical model for computing modular system

Fig. 7.5 Design of Excel spreadsheet for computing Modularization parameters
7.3.5 Evaluation of Modularization Opportunities

The interface constraint factor for the system was determined in the previous section and its value is 2.3083. The opportunity of modularization is determined using the modularization function. The modularization function is given by

\[ f(\delta) = e^{-a\delta^2} \quad 0 \leq a \leq 1 \quad (\text{From Eq.7.1}) \]

where

- \( f(\delta) \) – modularization function
- \( \delta \) – interface constraints
- \( a \) – supplier development index

The supplier development index indicates the extent to which the supplier-buyer relationship can impact on the modularization opportunities and its value ranges anywhere between 0 and 1. An index of 1 denotes dual arm’s length relationship and 0 indicates strategic partnership. Let’s consider supplier development index value is 1, then evaluate the opportunities of modularization.

The interference constraint factor at different levels such as module, subsystem and system level are presented in cells P12, P7, P4 respectively,

The modularization opportunity computed for alpha equal to one,

- System Level: \( Q12 = \exp(-(P12^2)) = 0.945 \)
- Subsystem Level: \( Q7 = \exp(-(P7^2)) = 0.434 \)
- Module Level: \( Q4 = \exp(-(P4^2)) = 0.005 \)

Similarly, the modularization opportunities at the system, sub-system and module levels can be determined for various supplier-buyer partnership indices varies from 0 to 1 at the interval of 0.1 is presented in the Table 7.1.

The rate of change of opportunities for modularization with respect to the interface constraint factor at the module, subsystem and system levels are computed
for $\alpha = 1$ by using the equation 7.2. These values are presented in column R, cells R12, R7, and R4 respectively.

For module level (R12): $f'(\delta) = -0.45866$

For sub-system level (R7): $f'(\delta) = -0.79289$

For system level (R4): $f'(\delta) = -0.02241$

The sensitivity of modularization with respect to supplier-buyer index $\alpha$ for all the levels with $\alpha = 1$ is presented in column S, cells S12, S7 and S4.

$S_a f'(\delta) = \frac{\alpha}{f(\delta)} \frac{df(\delta)}{d\alpha} = -\alpha \delta^2$ From Equation 7.3

For module level (S12): $S_a = -0.05921$

For sub-system level (S7): $S_a = -0.83511$

For system level (S4): $S_a = -5.32831$

Table 7.1 Modularization function values for various supplier-buyer partnership index.

<table>
<thead>
<tr>
<th>Supplier Buyer Partnership Index</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>0.5869</td>
<td>0.34450</td>
<td>0.20220</td>
<td>0.11870</td>
<td>0.06970</td>
<td>0.04090</td>
<td>0.02400</td>
<td>0.01410</td>
<td>0.00810</td>
<td>0.0049</td>
</tr>
<tr>
<td>Sub system</td>
<td>0.91990</td>
<td>0.84620</td>
<td>0.77840</td>
<td>0.71600</td>
<td>0.65870</td>
<td>0.60590</td>
<td>0.55730</td>
<td>0.51270</td>
<td>0.47160</td>
<td>0.4338</td>
</tr>
<tr>
<td>Module</td>
<td>0.99410</td>
<td>0.98820</td>
<td>0.98240</td>
<td>0.97660</td>
<td>0.97080</td>
<td>0.96510</td>
<td>0.95950</td>
<td>0.95370</td>
<td>0.94810</td>
<td>0.9429</td>
</tr>
</tbody>
</table>
Fig. 7.6 Modularization at various stages of Product development

The modularization opportunities computed for various values of alpha (Ranging from 0 to 1 in the interval of 0.1) and shown in Fig. 7.6. The Fig. 7.7 shows the modularization at three levels of product design and development such as module, subsystem and system levels.

Fig. 7.7 Modularization at different stages of supplier buyer partnership
7.4 CONCLUSIONS: In this chapter, the concept of modularization as applied to product design has been studied along with mathematical modeling to theoretically examine and simulate a given system. Among the various approaches of mathematical modeling available, the one based on the parameters of interface constraint factor and supplier development index has been studied.

We introduced a method for computing various modularization function values at different stages with respect to supplier buyer partnership index. Mathematical model is studied from the literature and with excel functions, an excel spreadsheet is designed for computing the modularization function values. The developed model is the non linear relationship between modularization and interface constraints which is characterized by the number of components and respective interfaces. One of the main advantages is to demonstrate that a change in the interface compatibility at one level of analysis can have significant impact at other levels in short period of computing time.

With a modular product development strategy the organization of the information exchange between concept generation activities and detailed design are also likely to differ from that pursued with an integral product development strategy. This is because with a modular product the architecture of the product is decided before the detailed design activities begin whereas with the integral product development process the architecture emerge from the solutions to the detailed design problems. The modular product development strategy therefore provides a greater opportunity for imposing decisiveness on the decisions that link concept development, system level design and detailed design activities. Finally, with a modular product development strategy a much more extended use of parallel information processing such as supplier development is likely to be an efficient way of minimizing Product development lead-time. Hence, study of modularization plays vital role in Lean Product development.

219