東　　海　　大　　學

工業工程與經營資訊研究所

碩士論文

協同設計之開放架構

研 究 生：楊惇婷
指導教授：黃欽印 教授

中 華 民 國 九 十 七 年 六 月

Open Architecture for Collaborative Design

By
Tun-Ting Yang

Advisor: Prof. Chin-Yin Huang

A Thesis
Submitted to the Institute of Industrial Engineering and Enterprise Information at Tunghai University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in
Industrial Engineering and Enterprise Information

June 2008
Taichung , Taiwan , Republic of China

ACKNOWLEDGEMENTS

在東海的這兩年，是我生命中最寶貝的一段時光。在東海，我學習到不同於過往的專業知識與經驗；在工工，我交到了我相信會是一輩子的好朋友，以及不斷在旁給我指引與鼓勵的師長們，這過往兩年滿滿的回憶，不僅僅會是我求學生涯中的一塊不可替代的足跡，更會是我未來在人生旅途中可以咀嚼回味，做為前進一塊不可或缺的動力。

謝謝我的指導教授黃欽印老師，在我研究生涯的兩年中，不論是在做學問上、出國進修、生活中的困惑等，皆給予惇婷極大的支持與鼓勵，很榮幸能在黃老師的指導之下，完成此研究生活的兩年，除了學識修養上是同學們有目共睹，在待人處事上更是給予惇婷許多的啟示與成長，心中真是有說不完的感激，謝謝黃老師一路以來的引領與支持，這對於惇婷真的有很大的意義與影響，謝謝老師。

工工所中，另外需要感謝的是姚銘忠教授，姚老師是我十分敬佩與景仰的一位教授，雖然沒有榮幸成為他的研究生，但不論在課業上、學習中，或是生活上所遇到的困難，姚老師就像父親一樣，總會在我背後支持我，給予我力量，真的很謝謝姚老師。

感謝 Auto 和 IIM 的朋友們，這兩年有你們的陪伴，真的讓兩年看似枯燥的研究生活有了一注活水，很感謝您們在我需要力量的時候，給予我莫大的支持與鼓舞，能夠進入東海工工，成為你們的朋友，真的是我非常榮幸，並引以為傲的一件事情。謝謝韋珺、啟偉、政穎，那半年的時光真的是我非常珍惜的日子，尤其韋珺，很開心能成為你的好朋友，我只想和你說: 有妳在真好!!謝謝一路上，當我躊躇、煩躁的時候，妳願意陪同我，聽我傾訴、提供我建議與協助。

當然我要感謝我最重要的兩個人，我的母親以及男朋友冠宇，感謝母親給我絕對的自主權以及完全的支持，在無數個等待我回台北的周末，我卻總是有忙不完的事情，而無法長伴您身旁，時常也是火氣很大，謝謝您給女兒的體諒、關心與鼓舞。

謝謝冠宇讓我認識了工工，更讓我學習許多我在政大四年都沒有機會接觸到的人事物，這兩年的時光，你給我很大的支持，不論是實質上或是精神上，是你讓我有今天的成長與學習，謝謝你，希望我們都能完成自己心中的那個夢。另外，也要感謝冠宇的家人這兩年來對我的照顧，謝謝皮皮和麻糬帶給我多個愉快的周末。

謹以此論文獻給以上所有的人，感謝有你們，才有惇婷享有豐厚果實的今天，謝謝你們，願平安喜樂。

Open Architecture for Collaborative Design

Student: Tun-Ting Yang Advisor: Prof. Chin-Yin Huang

Department of Industrial Engineering and Enterprise Information

Tunghai University

ABSTRACT

Collaborative design (CD) is no more a new topic in these day and emerged around the 90s. Related researches have accomplished quite a view contribution so far to those aspects such as collaboration platform formation, CD environment construction, or even discussing about the communication mechanism across multiple-disciplines participants or protocols agents shared with each other.

However, there is so little reference related to CD on the way to demonstrate how a CD project should proceed and when deciding in executing a CD activity, what are the crucial dimensions should decision-makers or system-developers should take into considerations. Therefore, we try to give a broad map or guidelines of CD in view of collecting different perspectives need to be considered when implementing design collaboration in this thesis.

The CD cube framework provides people a general direction and main dimensions coping with CD. The framework directs people to make sure what aspects should they take into accounts, what stages are they try to participate in, and to what extent they want to collaborate with. The CD cube help managers to plan or analyze a CD project in a more complete and sound way for making plans or decisions under the organized and structured dimensions.

Keywords: Open Architecture, Collaborative design, Design Engineering, Information/Communication Framework, CD Elements

協同設計開放架構

學生：楊惇婷　　　　　　　　　　　　　　　指導教授：黃欽印 博士

東海大學工業工程與經營資訊研究所

摘要

協同設計從九零年代發跡，截至目前已有十年左右的發展，可以算是一門顯學。協同設計相關的研究著作相當充沛，舉凡協同設計平台建置、協同環境建構與探討、協同績效與影響的要因、協同活動下的資訊溝通架構等等，皆已有相當的發展成果。

不僅僅在學術界，在實務界中協同設計的應用與發展也有相當的成效，以台灣產業為例，實施協同設計的產業就跨及多種領域，包含 IT 系統（神達、華碩等）、光電產業（科橋、全台晶像等）、散熱零組件（建準、雙鴻等）、紡織（寶成工業、聚陽、佰龍機械、宏遠興業等）、機械電機（力山工業、上銀科技、福裕、三龍產業等）、工業設計（浩漢、世訊科技等）、車輛（光陽工業、三陽工業、巨大機械）、通訊（奇美通訊、正文科技、威寶電信）、模具（綠點高、雙葉開發等），此役可以展現出協同設計架構提出與探討的意義所在。

然後在過往的研究中，主要的研究仍是針對特性領域進行，譬如多數協同設計的研究是針對協同過程之中，進行資訊交換與溝通平台的建立等較為技術性的文章，亦有不少論文是針對交換資訊的內容與知識討論，但是截至目前為止仍沒有較為統整概括出協同設計範圍與需要考量範圍的架構的文章，故本文主要是提出一種協同設計的分析開放架構，彙整實行協同設計時需要納入考量的主要因素，以利管理決策人員或是系統發展人員，在開發或指定相關決策時，有依循參考的準則。

關鍵字詞：開放架構、協同設計、設計工程、資訊溝通架構、協同設計因子

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 Introduction

1.1 Background

With rapid changing global environment today, developing capability dealing with flexibility to environment changes would become a significant issue in today's business field. To get the favor wind of this battle, enterprises tend to focus on their own core competences.

The evolving progress of Rothwell's four generation innovation processes (Rothwell, 1994) specifies that collaboration and coordination within innovation progress, which is the design or R&D stage of product development, is more and more significant, and is also proofed by their proposition regarding productivity of parallel design activities (Figure1.1).

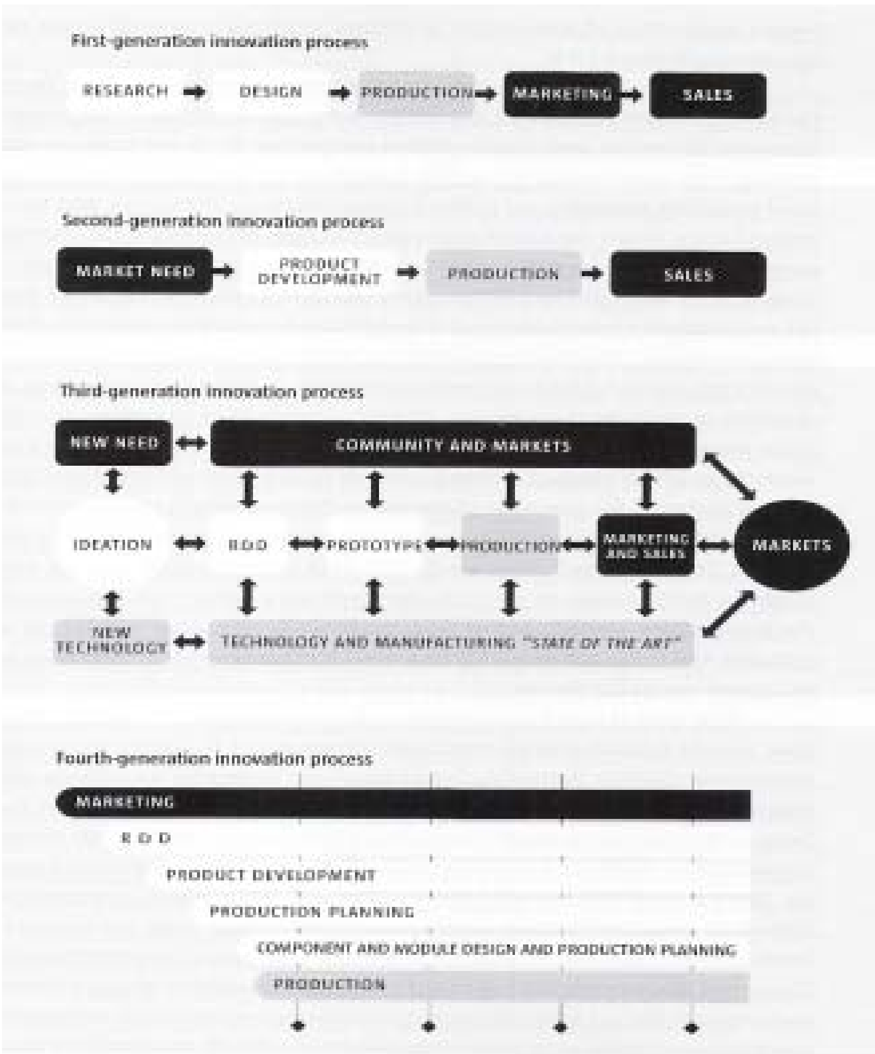


Figure 1.1 Rothwell's four generation of innovation processes (Source: Rothwell, 1994)

1.2 Motivation

META Group clarify collaboration into four key domains: design collaboration, marketing and selling collaboration, procurement collaboration, and planning／forecasting collaboration. In recent years, collaboration has become a common business pattern dealing with rapid changing global environment, which shows the importance of this topic empirically again.

Taiwan industries have carved out their own road of changing their business model from Original Equipment Manufacturing (OEM) to Original Design Manufacturing /Own Brand Manufacturing (ODM/OBM) as shown in Figure 1.2. Though manufacturing is important, the significance of design and brand will show its position in the future, and this is also the reason supporting us to focus on clarifying the ambiguity of CD contexts and essence in this paper.

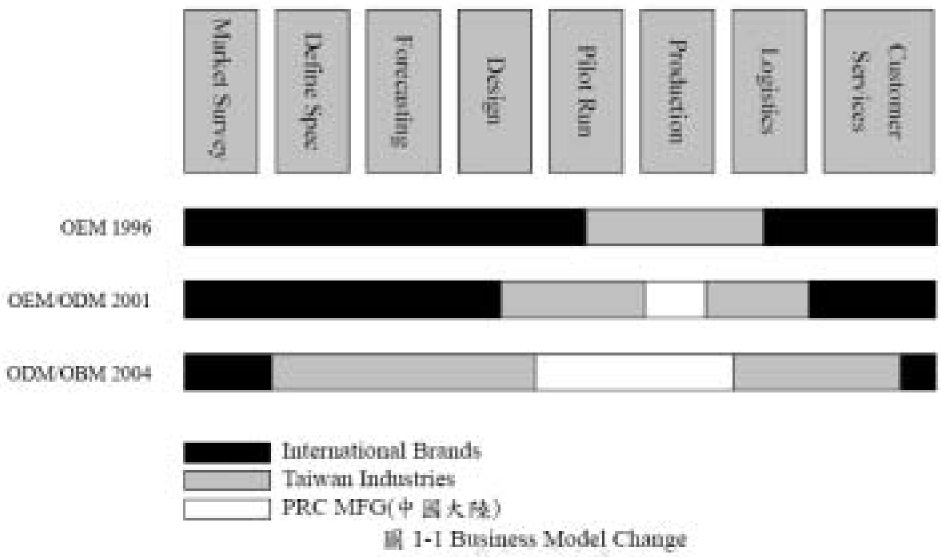


Figure 1.2 Business model changes (Source: Su et al., 2004)

In Taiwan, by the changing pace of business model and enterprising circumstances, more and more corporations have evolved from OEM to ODM, even OBM (Chen, 2008). DoIT, R.O.C. initiate e-collaboration project called ABCDE project to investigate on recent development of collaboration in Taiwan, and to animate Taiwanese corporations to pursue further progress on collaboration to be on an advantageous position and to cope with changing business architecture to ODM, OBM, even CDM under fierce global competition.

In view of recent state of play, ABCDE project hosted by Taiwan government does lend a impetus to practical would on applying collaboration as a solution or approach dealing with problems of today's fast changing global competition. According to survey of DoIT, R.O.C., we may see the extensive development of collaboration under various business environments all around the world. For example, application of collaboration has been stretched to industry of IT system designing, IT peripheral components, industry computers, photoelectric industry, textile industry and so on. (See Figure 1.3)

產業別	企業
IT系統	大同、大眾電腦[1]、華碩電腦[2,3]、神達電腦、華宇電腦[1]
IT週邊及零組件	華通電腦、聯昌科技、技嘉科技[2]、致茂電子、智邦科技[1]、環隆科、誠訊、智謙
工業電腦	凌華科技
光電產業	全台晶像[1]、科橋[1]
散熱零組件	建準[1]、奇鋐科技、雙鴻
紡織	力麗、宏遠興業、佰龍機械、興采實業、寶成工業、聚陽、世堡、盈透染整、南緯實業
機械及電機	榮剛材料、力山工業、程泰機械、主新德科技、合濟工業、福裕、三龍產業、上銀科技、華城電機
高爾夫球產業	復盛、明安國際
工業設計	世訊科技、浩漢
車輛	光陽工業、三陽工業、巨大機械
通訊	威寶電信、正文科技、奇美通訊
模具	雙葉開發、台灣富翔巴精模、線密高
其他	探微科技、寶熊漁具、翔威國際、西基電腦動畫、雙葉開發

1.亦有實施綠色供應鏈(RoSH)企業 2.亦有實施綠色供應鏈(WEEE)企業 3.有實施協同設計及協同專利加值系統企業
(資料來源：經濟部技術處示範資訊應用開發設計)

Figure 1.3 Examples of Taiwan enterprises applying CD (Source: Chen, 2008)

1.3 Thesis Objectives

To deal with these issues triggered by distributed cooperators effectively and efficiently, phases of product planning and design would show its importance in view of changes costs and management. Under the course of product life-cycle, collaboration can take place along the whole process from product development to mass production, even to the last recycling of products, and collaborative activities can come about through the whole supply chain. According to Aberdeen Group, 80% of the NPD cost takes place at the design stage before into mass production. Hence, this thesis would pay our efforts mainly on topics of collaborative engineering design to tackle with the collaborative problem to its root.

Topics of collaboration have become quite mature in recent years. However, issues and themes addressed by researches done so far are so extensively that when running the project of collaborative design, people may lack of the directions or guidelines without the whole picture. Hence, we try to give a comprehensive framework of collaborative design make practices implemented more thoroughly and easily. The open architecture for collaborative design proposed is try to provide the elementary skeleton and can be extended and adapted to diverse changing environment, which is the main contribution of this thesis. Therefore, developing an open architecture for collaboration design is important and necessary in academic and empirical field.

Reasons for focusing on the design stage are mainly because the most crucial policy and most of costs have been decided in the phase. Besides, the most common issues dealing with product development and design is the conflicts between departments of design and manufacturing, which have been extensively discussed of topics such as concurrent engineering, rapid product development, early-involvement, etc. Design stages along the whole process would play a significant role concerning returns and outcomes, which give us a support to focus on the topic of collaboration mainly on the phase of product or service design. Once the OA for CD is proposed, then those in needs would be able to have some references or a "bigger picture" for executing CD projects and making related decisions under decisive guidelines.

1.4 Thesis Organization

In this thesis, we give an introduction of the current development situation of CD, background of applying CD, and thesis motives first. Chapter 2 collects relating references and makes a summary category of CD related research topics, defining open architecture (OA), and introduces some concept such as ESI, CE, VE, PLM, etc. In chapter 3, we have an overview on the CD cube framework, and define all elements of each dimension. Chapter 4 represents one way to use the framework by discussing CD of three various scope and details on the topics of how CD works to different level of scope, namely cross-functional, cross-company, and cross-industry CD. Finally, the conclusion and future research suggestions are made.

Thesis organization

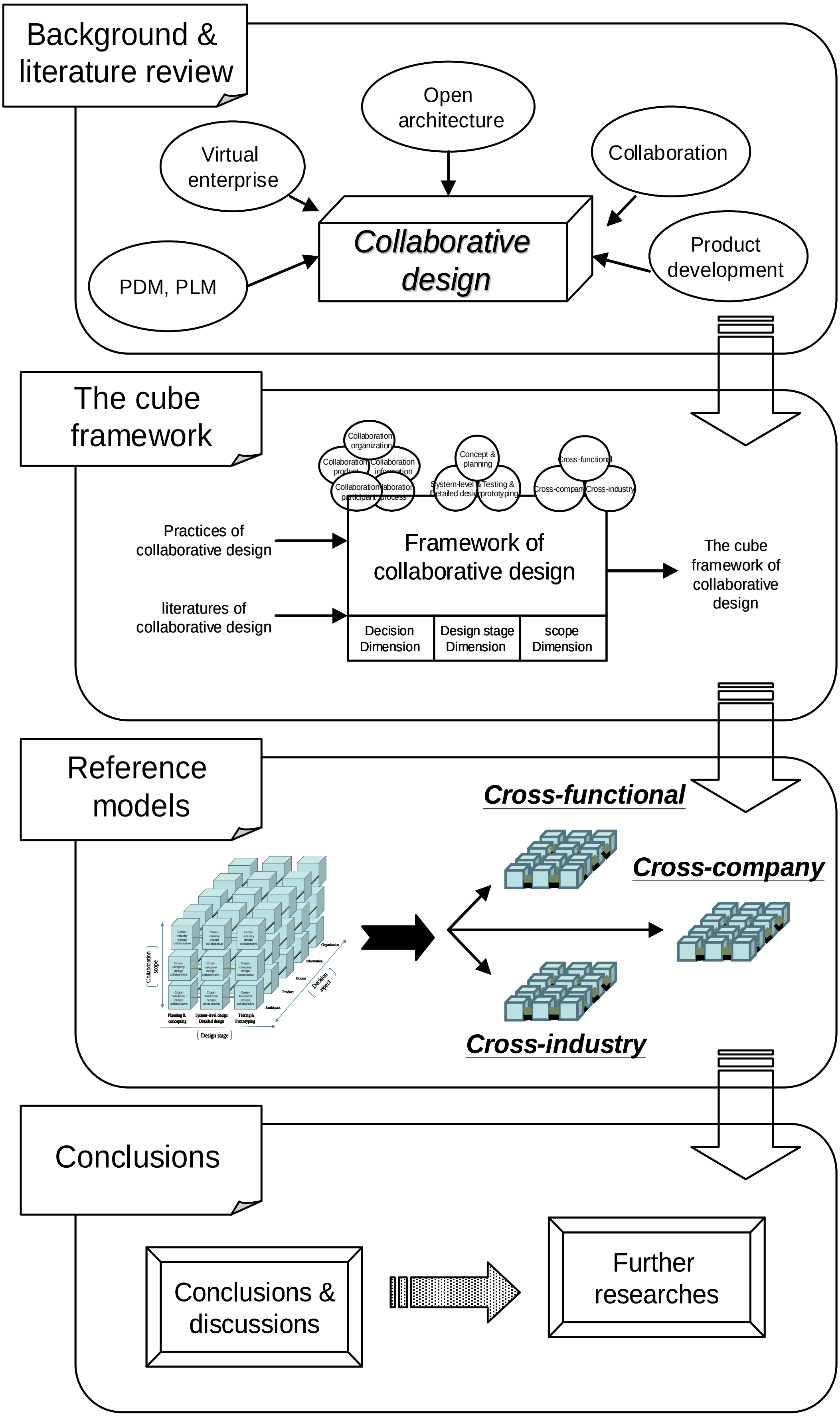


Figure 1.4 the thesis organization (Source: original)

Chapter 2 Literature review

2.1 Collaborative design

Researches regarding collaboration activities have become popular issues from 1990s, and also have developed into many schools and research branches of this domain for recent years. Collaboration is an extensive domain that includes domain issues from product development collaboration to even e-commerce. Hence, the core arena of this thesis would focus specifically on collaborative design under the key subject of product development collaboration. Two basic concepts of cooperation and collaboration used in this paper would be definition as proposed by Msanjila and Afsarmanesh (2008):

Cooperation involves not only information exchange and alignment of activities, but also sharing resources for achieving compatible goals. There exists however, a common plan, which in most cases is not defined jointly but rather designed by a single entity (e.g. coordinator/administrator of the cooperation alliance), and that requires some low-level of co-working.

Collaboration is a process in which entities share information, resources and responsibilities to jointly plan, implement, and evaluate a series of activities to achieve a common goal. It implies a group of entities that work together and enhance the capabilities of each other. Collaboration involves mutual engagement of participants to solve a problem together. We make a basic collection of CD related researches done so far into six main topics as following (See Table 2.1).

Table 2.1 Collection of CD topics (Source: Original)

Topics of CD	Literature & references
Methods and models developed for CD	Klein, 1994; Brelinghoven, 1995; Witzerman and Nof, 1995; Huang, 2002; Liao, 2002; Su, 2004; Hsu, 2004; Abdalla, 2006; Wang et al., 2006; Luo, 2006; Mahesh et al., 2006; Yang and Zhang, 2006; Luo et al., 2006
Collaborative environment developed to facilitate or support CD	Khosla & Brown, 1995; Chryssolouris et al., 2006; Hao, et al., 2006; Mervyn et al., 2006; Robin et al., 2007
Knowledge/ intelligent capital of CD	Thomas and Baker, 1992; Panchal et al., 2006; Bullinger et al., 2007
Management issues of CD (including social/ organizational aspects)	Lin, 2004; Wang, 2006
Applications of CD	Brelinghoven, 1995; Hao et al., 2006; Trappey and Hsiao, 2007
Performance evaluation of CD activities	Lin, 2004; Kung, 2004; Wang, 2005; Panchal et al., 2006; Lu, 2006

2.1.1 Methods and models developed for CD

Methods developed for collaborative engineering design to detect conflict intelligently or facilitate CD process more smoothly account for most of researches related to CD. Within CD process, flow of process and information is really important to acquire sufficient materials for decision-making, thus methods or algorithms developed to describe the process and information flow of CD activities is a significant issue of this arena, academic studies are like the collaboration system extended from single-location CAD application to multi-location (Huang, 2002), and the agent-based service-oriented framework proposed by (Wang, et al., 2006) etc..

Others like issues of tools and software support systems developed to facilitate collaborative engineering design such as model for cooperative concurrency design tools in a design environment (Brelinghoven, 1995), proposition of three reference CD models based on types of business models adopted of the collaboration circumstances (Liao, 2002), a new type of DSM which extends the traditional DSM from applying in cross-functional development to a higher level- CCDC (Su, H. –J., 2004), cooperative design system proposed by Luo (2006), web-based framework for distributed cooperative manufacturing system proposed by Nee et al. (2006), and so on.

A knowledge base called Active Semantic Network (ASN) proposed by Roller et al. (2002) has been concerned as a powerful tool for communication and coordination of all activities of developers working on various documents, representing product data. Information exchange and communication are also central issues, the framework of product information exchange and data semantics integration, product and process information representation and exchange proposed by Yang and Zhang (2006) is an example.

2.1.2 Collaborative environment developed to facilitate or support collaborative engineering design

Collaborative environment refers to surrounding circumstances of collaborative design activities virtually and practically. Research topics regarding virtual enterprise to illustrate cooperation relationships and situations and environmental infrastructure or required elementariness to initiate the CD activities such as communication platform and technology are all involved in this theme. Virtual environment discussed by Chryssolouris, et al. (2006) integrated ‘Plug-and-Play’ computer environment to support collaboration across an extended enterprise (Mervyn, et al., 2006) is one of the instances.

2.1.3 Knowledge representation and form of CD system

The above studies are the exterior part of CD issues concentrating on the information infrastructure and methodologies interpreting process information flow. However, the interior part of CD is also essential to illustrate knowledge/ intelligent capital of collaborative engineering design.

Knowledge representation and knowledge form of CD system is one kind of these branches. However, it has become more and more significant on the research issues recently, and focuses more on the accumulation of product and process knowledge of the CD activity. During the collaboration activity, how to preserve the intelligent capital (knowledge) related with CD or design task itself in order to facilitate further coordination and cooperation is taken as issues worth being discussed. For example, Thomas and Baker (1992) identify some of the various types and levels of knowledge involved in the design process as well as their relationships and characteristics; Mills and Goossenaerts (2001) emphasize on the importance to implement knowledge infrastructure, provide us definition regarding data, information, and knowledge, and trigger issues about relations and context among the three in the product realization process. Another branch of knowledge representation topics is related to CD process and management such as issues of integration of designing the product and process design (Panchal, et al., 2006).

2.1.4 Management issues of collaboration engineering design

CD has been discussed for years, and therefore existing frameworks and models are sufficient for empirical studies use or being applied to other management issues such as in view of social or organizational perspectives. For example, Lin classified collaboration models in view of supply chain entities and the objected domain such as collaborative design or collaborative production, focusing on the collaboration management issues such as CRM, QR are all practical instances for collaboration marketing (2004). Wang focus on CD organization formation using the CWA-based team design method (2006).

2.1.5 Applications of collaboration engineering design

Most of references collected in this topic are focusing on applying existing concept of CD whether on the field of information framework or collaboration environment to different industries to show CD application and approval of the theories.

Hao et al. (2006) present the results of an industrial case study in the development of a collaborative e-Engineering environment for mechanical product design engineering by applying intelligent software agents, Internet/Web, workflow, and database technologies.

Trappey and Hsiao (2007) provide e-solutions for SMEs to participate in the global automotive supply networks and fulfill the prime’s real-time information requirement for their ODMs and propose a prototype system for automotive supply chain integration and the case emphasizes the applicability for smaller suppliers and their information exchanges.

2.1.6. Performance evaluation of collaborative engineering design activities

One of the CD topics are emphasizing on evaluating the performance of CD activities. This kind of topic includes discussions of defining key measurement index of successful CD activities (e.g. Wang, 2005), collecting critical successful factors of CD (Lin, 2004), proposing the metrics for assessment of design process performance, evaluating the adaptability of CD system, including the quantity measurement to the openness of product and process, etc. Others may take CD as an independent variable to the effect on dependent variables such as performance of New Product Development (NPD) (see Lu, 2006), and so on. Researches from other perspectives are such as design freedom, robustness, complexity, modularity, and coupling, and so on (Panchal, et al., 2006).

The six dimensions of CD research topics would be included into the framework we proposed in chapter three dealing with capturing the whole picture of CD activities except for the sixth part performance evaluation of collaborative engineering design activities. The section of performance evaluation of a CD activity is also important of the whole collaboration process but belongs to post phase of the whole process. Therefore, the CD performance evaluation is not in the scope of our discussion.

2.2 Fundamental topics concerning to collaborative design

The concept of CD is originated from concurrent/ simultaneous engineering, for which purpose is to achieve Early Supplier Involvement (ESI) and improve the quality and efficiency of product design and development. Decisions of 80% product manufacturing cost have been made in the early 20% of design stage, which shows how significant the role design played of the CD process.

With collaboration design, enterprises can realize the concept of ESI to build a communication platform among downstream and upstream of the CD process, and ensure the completeness of product design by approach of design for x (DFx) in early stages without enormous cost and time spent in post design stages because of design changes. Talking with collaboration, Product Data Management (PDM) and Product Lifecycle Management (PLM) are taken as common senses dealing with information supports and infrastructure installment, application of Virtual Enterprises (VE), however, is mentioned basically because of its contribution on facilitating the occurrence of CD.

2.2.1 Early Suppler Involvement (ESI)

The early integration of different experts serves as a mean to develop innovative products. This is also an important factor concerning costs because the main part of the end costs is determined in the early phase of product development (Bullinger et al., 2007). ESI is defined as a form of vertical co-operation in which manufacturers involve suppliers at an early stage in the product development/innovation process, generally at the level of concept and design (Bidault et al., 1998). To achieve shorter product development cycles, suppliers must be integrated early in the design process. We can see supplier involvement in new product development activities in Figure 2.1 indicates the significant role of design collaboration played in the early design stages specification, concept design, detailed design, and production design. At the concept design stage, suppliers help in identifying the most up-to-date technologies to be incorporated into NPD. Suppliers participate in detailed design by providing solutions to component and part design and the selection of the most suitable materials and catalog components, and so on.

New Product Development Process			
Specification	**Concept Design**	**Detailed Design**	**Production Design**
• Establish specifications collaboratively • Avoid ambiguity and information distortion • Set technical targets • Articulate trade-offs • Identify early changes	• Key product and process technologies • Product architecture • Contribute key ideas / concepts / critical components • Participate in concept evaluation • Establish interfaces between product subsystems	• Selection of proprietary parts & components • "Black box" designed parts & components. • Tolerance design • Detail controlled parts & components • Prototype testing and demonstration • Design for Manufacturability • Material selection	• Make or buy decisions • Tooling & fixturing design • Equipment acquisition • Design for Manufacturability • Quality control & assurance • Raw materials
Supplier involvement			

Figure 2.1 supplier involvement in new product development activities (Source: Huang and Mak, 2000)

Having suppliers involved in early stages comes the communication and information sharing conflicts and issues among cooperators or supplier-buyer relationships. As shown in Figure 2.2, buyer and suppliers do have communication mechanisms dealing with product design and design validation tasks in diagram (a), but in diagram (b), the communication and know-how sharing platforms do support the connection gaps among customers. In this way, the concept of ESI does arouse the needs of CD, and also can be expended into CD not only at the scope of within departments but also companies or industries, which regard all both suppliers and customers as CD participants under the same cooperation standard.

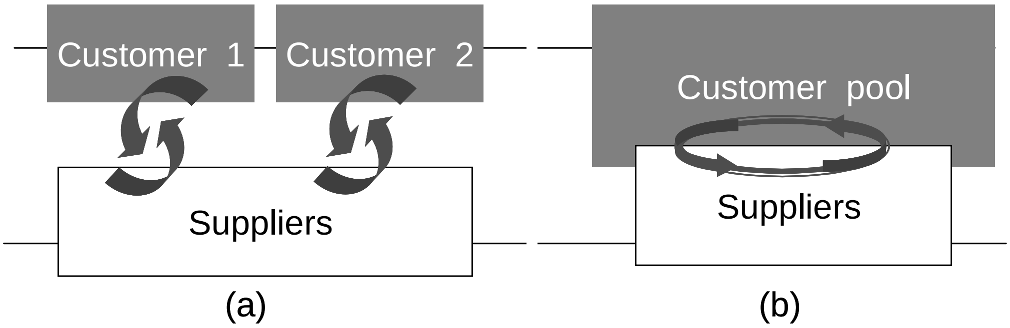


Figure 2.2 Generic ESI models: (a) decoupled ESI model; (b) integrated ESI model (Source: adapted from Tang, 2005)

In Figure 2.3, we can see the significant role of CD platform played in the common but complex scenario of global design collaboration activity. From the design stage idea/ requirement to mass production, information of product design, process and design process are required to share within cooperators. Therefore, to coordinate partners or participants of CD among multiple-disciplines teams would become an issue and worth to be discussed.

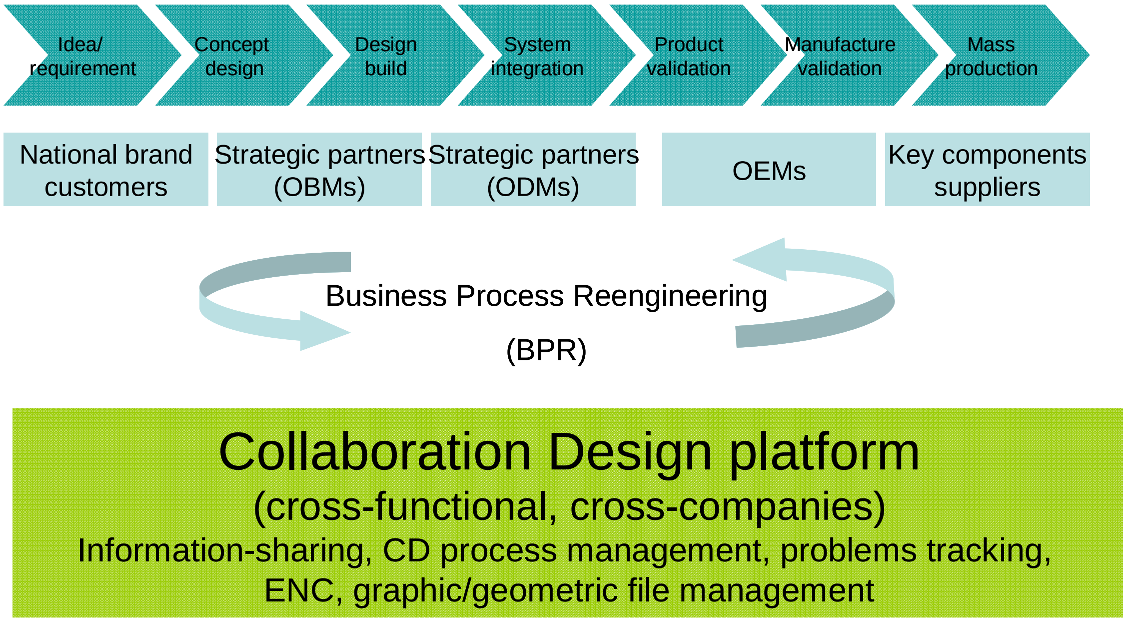


Figure 2.3 concept of collaboration design (Source: adapted from Chen, 2008)

Correlative issues would involve building connections within cooperators, setting up platform for information sharing, database or product repository installed combining Product Data Management (PDM), Product Lifecycle Management (PLM) for file and data management, approaches like E-commerce to facilitate the whole process, and so on. In the following sections (from 2.2.2 to 2.2.4), we would like to make a brief introduction to PDM, PLM, and VE, for which concepts are usually include and adopted when discussing issues of CD settings.

2.2.2 Engineering/ Product Data Management (EDM/PDM) system

Following the engineering solution center (ESC) concept brought up by Bullinger et al. (2007), the EDM encompasses holistic, structured, and consistent management of all processes and the whole data involved in the development of innovative products, or the modification of already existing products, for the whole product life cycle. The EDM systems provide the interfaces to CAD systems and other computer-aided applications (CAX), such as computer-aided manufacturing, computer-aided planning, and computer-aided quality assurance.

PDM, on the other hand, is a category of computer software used to control data related to products. PDM creates and manages relations between sets of data that define a product, and store those relationships in a database. It is an important tool in product lifecycle management. PDM systems provide the tools to control access to and manage all product definition data. It does this by maintaining information (meta-data) about product information. PDM systems, when tightly integrated with other product development tools; do this transparently and with minimal additional effort on the part of the user. In addition, PDM tools provide valuable functionality with process management particularly as it relates to configuration management or engineering change control. This environment is depicted below in Figure 2.4. (Kenneth Crow, DRM Associates, 2002)

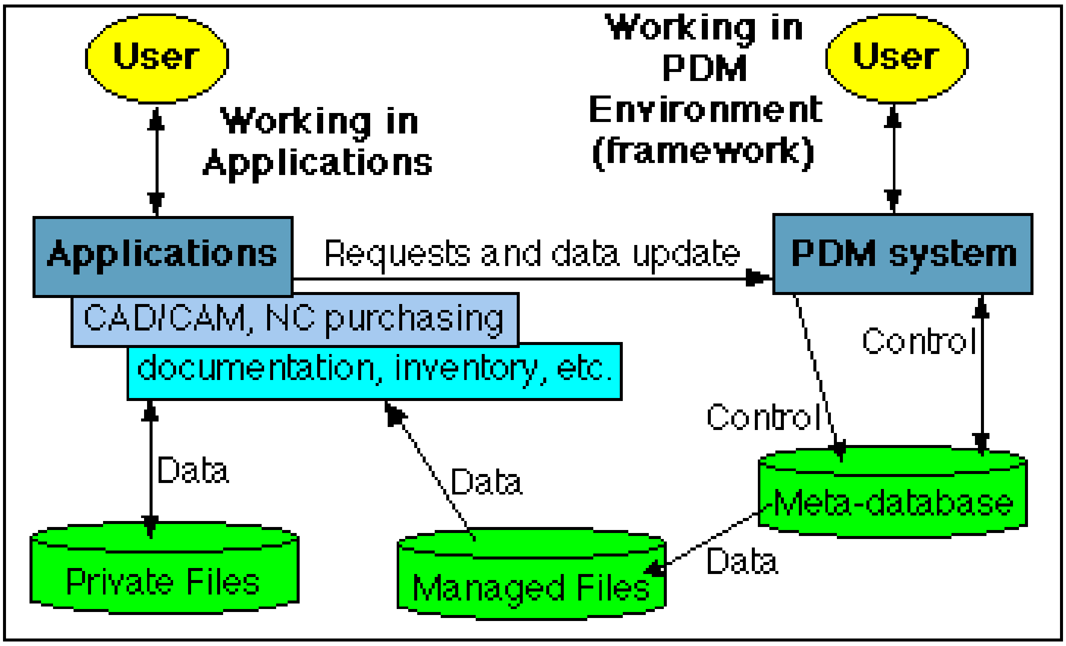


Figure 2.4 PDM environment (Source: Kenneth Crow, DRM Associates, 2002)

PDM systems support process management by defining process steps related to the development, distribution and use of product data. Collaboration can be supported in several ways. First, a PDM system may be the gateway that a team uses to access the information under discussion avoiding the need to copy and distribute a series of paper documents. Second, the PDM system may provide a synchronous or asynchronous collaboration environment for team members to access, present, review and product feedback on product and process information. Further, this collaboration tool may incorporate a view and mark-up capability and providing the ability to store marked-up files or documents by collaborator. Third, what are now described as collaborative product commerce systems (CPC), provide extended PDM functionality and access control outside the enterprise for customers, suppliers and interested third parties (e.g., regulatory agencies). This speeds the distribution of information, enhances coordination, and speeds the capture of feedback. (See Figure 2.5, Kenneth Crow, DRM Associates, 2002)

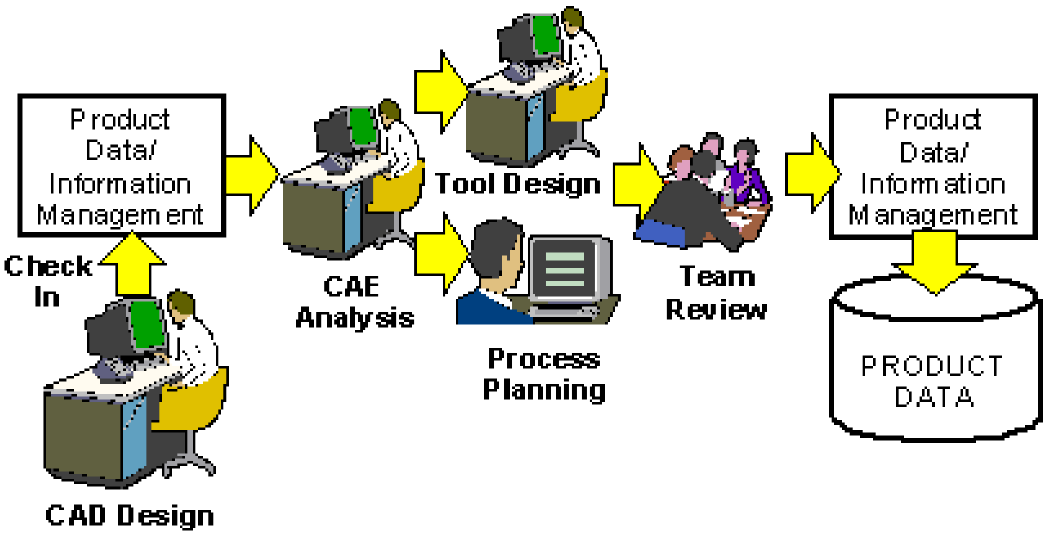


Figure 2.5 Example of process Management and Workflow applying PDM (Source: Kenneth Crow, DRM Associates, 2002)

2.2.3 Product Lifecycle Management (PLM)

PLM, sometimes "product life cycle management", represents an all-encompassing vision for managing all data relating to the design, production, support and ultimate disposal of manufactured goods. PLM concepts were first introduced where safety and control have been extremely important, notably the aerospace, medical device, military and nuclear industries. These industries originated the discipline of configuration management (CM), which evolved into electronic data management systems (EDMS), which then further evolved to product data management (PDM) (Active Sensing, Inc., 2007). Figure 2.6 shows the role of PLM play in the field of product design.

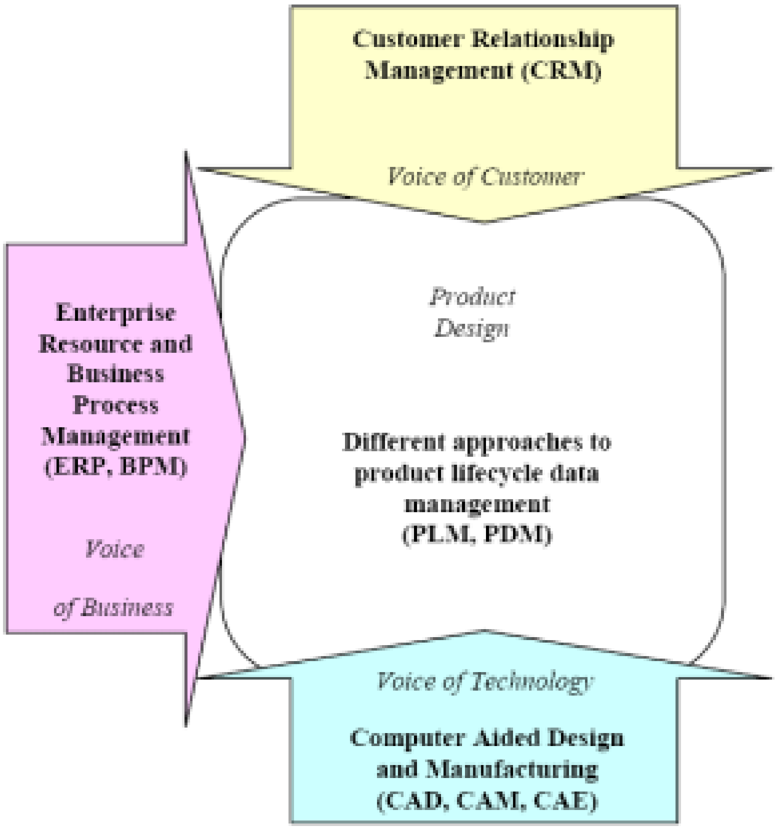


Figure 2.6 Inroads towards product lifecycle data management (Source: Jørgensen et al., 2007)

Managing the product data is not sufficient anymore in these days, therefore when talking about product development and design, concept of product life-cycle (PLC) should also be mentioned about. PLM is the process of managing the entire lifecycle of a product from its conception, through design and manufacture, to service and disposal. Within PLM there are four primary areas: Product and Portfolio Management (PPM), Product Design (CAx), Manufacturing Planning (MPM), and PDM. (Kenneth Crow, DRM Associates, 2002)

The essential elements of PLM are: Manages design and process documents, Constructs and controls bill of material (product structure) records, Offers an electronic file repository, Includes built-in and custom part and document metadata ("attributes"), Identifies materials content for environmental compliance, Permits item-focused task assignments, Enables workflow and process management for approving changes, Controls multi-user secured access, including "electronic signature", and Exports data for downstream ERP systems (Active Sensing, Inc., 2007). Along the process of PLM, existence of conflicts or coordination among participants involved lead to the chances of applying concept of collaboration. For example, under the situation of considering all the product lifecycle as a whole process in product design, with the support of PDM and PLM technology, CD is more necessary and easier to take place for cooperation of multiple-discipline groups from different design stages.

2.2.4 Virtual Enterprises (VE)

VE is a common concept used recently, which can be defined as a temporary aggregation of core competences and associated resources collaboration to address a specific situation, presumed to be a business opportunities (Goranson, 1999); as a network of independent organizations that jointly form an entity committed to provide a product or service (Jagdev and Thoben, 2001).

The synthesis is temporary, when the opportunity fades away, the VEs vanishes into constituent parts to reassemble into other configuration for other emerging opportunities. Four types of VEs are proposed by Goranson as opportunity-driven, capability-driven, supplier chain (top down), and bidding consortium. What should be noticed is that few pure cases seem to exist, and best practices may be composed of different types. With supporting of IT technology, and internet environment today, facing the Make-or-Buy issues, which can be regarded as a opportunity of cooperation for entities of two individual systems may have to interact with one another, along the process of CD or simply product development and design, VE make “Buy” decision more easily to be taken into consideration by supporting the cooperate environment and coping with unexpected environment changes.

What may concern here is not whether the decision is “Make” or “Buy”, each can be taken as an application situation and basis of executing collaboration. Take the stage of product design for example. The “Make” decision, which indicates the whole process from configuring the main concept design to pre-empt stage occur within an enterprises, can be taken as cross-functional design collaboration. With the “Buy” decision, we can take the cooperation of product design, which means that the concept design and other design validation tasks may be carried out by different entities of multiple systems, within enterprises across industries as cross-companies or cross-industries design collaboration applying the concept of ESI mentioned earlier. Therefore, the collaborative approaches through VE can also be assumed as applications and practices of cross-companies or cross-industries design collaboration.

2.3 Comprehension of CD researches developed and related issues

2.3.1 Development process of CD

For the past decade, collaboration has been developed and matured, especially within the domain of product design and development. Typically, we can follow the course of change and development CAD system to understand corresponding development of CD environment and issues.

Traditional Computer-Aided-Design (CAD) system improving the efficiency and effectiveness on design information interchanging and faster the concept design visualization process, but the way it only support single designer to do design does not fit the trend and business surroundings.

Evolved collaborative CAD system conquered this problem enabling multiple designers to work on a design together, such as system called Computer Supported Cooperative Work (CSCW). With IT such as internet supporting and Web 2.0 etc., Web-based CAD system comes to appearance and facilitates extensive application of CD. Web-based CAD system such as C-DeSS, CDFMP integrates web-based multimedia tools with web-based model display. So this allows multiple users from geographically distributed locations to share their design models through internet technology (Liu, et al., 2006).

2.3.2 Research issues

Panchal et al. (2006) also collect eight issues that need to be concurred regarding CD framework, which are:

(1) Adaptability to network architecture changes or malfunction,
(2) Usability on heterogonous platforms with heterogonous operating systems,
(3) Heterogonous languages for different agents (semantic interoperability),
(4) Capability to transmit message and data changes (semantic interoperability),
(5) Rapid configuration of the product realization environment (considering reasons like Time-to-Market (TTM), and so on),
(6) Minimize the impact of agent service changes,
(7) Readiness for future expansion, and
(8) Readiness for discrepancy of process information.

2.3.3 CD components

Interactions modeling between factors influencing the design system are basically involved in the following Figure 2.7 (Robin et al., 2005). Interest of this model is to make appear and define all elements which influence design system and interactions between them to support engineering management according to structuring of decisions making. From this fundamental map, it describes the context of product design system and integrates all these elements, including actor, process, organization, product, knowledge, and environments. Eight kinds of links within all of the six elements compose research contents of design topics. Each can be taken into more detail discussions and also indicates diverse researching branches of this topic.

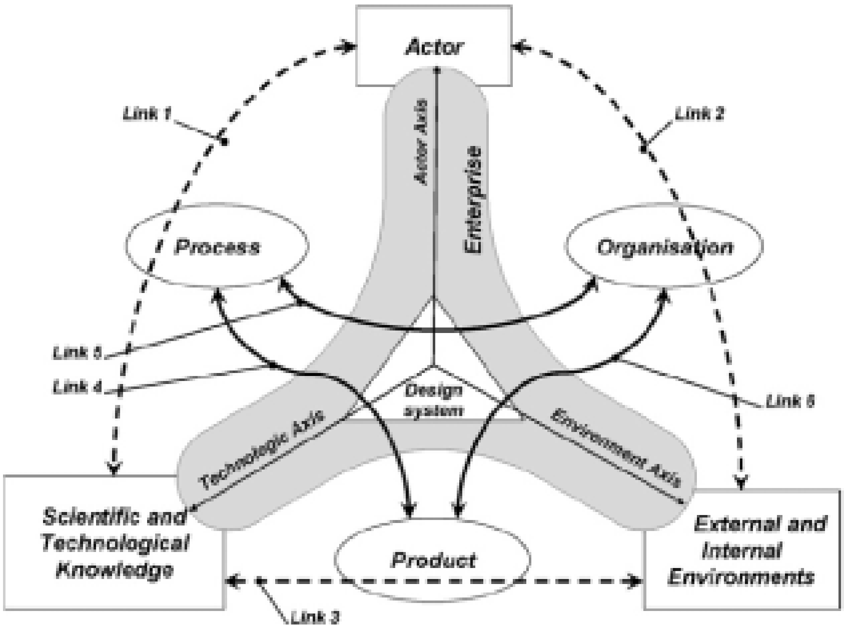


Figure 2.7 Interactions modeling between factors influencing the design system (Source: Robin, et al., 2005)

According to Panchal, et al (2006), scopes of CD issues can sum up to seven types, containing development of standards for information representation, communication between heterogeneous resources (semantic interoperability), seamless flow of information between humans and computers, methods for efficient collaborations between designers, strategies for conflict resolution, engineering repositories, and coordination and transaction management. As we mentioned earlier in section 2.1, most of researches are focusing on proposing information framework of CD or the communication mechanism of the collaboration activities.

2.3.4 CD scope

Dealing with CD mechanism from literature reviewed, we find that most papers handling cross-functional CD issues, and at most the scope of CD is reached to cross-companies.

Four major product CD types proposed by Lu (2006) are inter-corporation CD, multiple-sites CD, B2C CD, and cross-SC (supply chain) CD (Lu, 2006; Liao，2004；Wang，2005). The first and second type of CD belong to the cross-functional one in our proposed model, and the rest two ones are all included in our second type CD scope as cross-company CD since this can take place cross or within the supply chain, and the objects can be individual, corporation, customers, or buyer-vendor relationships, etc.

2.4 Open architecture

2.4.1 Development background and meaning of OA

In this thesis, we will propose an open architecture for CD. Hence, we would like to take a close look at the concept of open architecture (OA) first.

The origin concept of OA is first proposed and executed by the company International Business Machines (IBM) for expending their customer demand by building up more connections and channels. However, convenient channels is not the only and necessary condition for improved sales but to have the prerequisite of formalizing and clarifying the components of the computer systems so that customers may be able to choose their own specifications of the confined formation or architecture. Therefore, IBM developed the OA for computer assembly. For example, to assembly a computer, we may have to combine motherboard, CPU, CD-ROM, main memory, hard disk drive, power supply unit, and VGA Card; however, the specification of each component can have choices depends on owner's usage. Thus, with clear definition of necessary constituents to assembly one computer and make it work, rest would the selection of customers based on their needs.

Another application of OA is the concept of "Plug-and-Play" concerning computer features widely carried into execution in the field of computing technology. "Plug-and-Play" allows the addition of a new device, normally a peripheral, without requiring reconfiguration or manual installation of device drivers. Modern plug-and-play includes both the traditional boot-time assignment of I/O addresses and interrupts to prevent conflicts and identify drivers, as well as hotplug systems such as USB and Firewire.

Applying the OA conception, we would like to clarify the complex CD issues in view of dimensions of composing factors in a clear way. Once the OA for CD is proposed, then those in needs would be able to have some references or a "bigger picture" in executing CD projects and related decision-making process under the directions of decisive guidelines.

2.4.2 Definition of OA

The "openness" refers to the ability of a system to be readily adaptable to changes either inside or outside of it. (Panchal, et al., 2006) OA is a pattern of nonfunctional requirements that can help to create and maintain more open and flexible complex systems, and systems of systems. Organizations with large, complex systems are looking to OA to help manage complexity, increase flexibility, and reduce their costs. Others are looking to a broader approach of openness to enable greater collaboration, innovation, and social policy, for which the technical aspect of OA is a foundation. Nelson (2007) makes a clear comparison and introduction to this issue, defining the open and close system, OA technical requirements, and principles for OA.

2.4.3 Open and close system

The difference of open and close system depends on its condition of technical and business barriers, which means that if the condition is no technical barriers (such as dependency on closed components, incomplete specifications, and the like) and no business barriers (such as patents, licenses, NDAs, third-party agreements, and so on), then it means both the component and the system are open. On the other hand, a fully closed system is not extensible and does not interoperate with other systems. In an organization with fully closed systems, data is often exchanged through sneakernet, or its modern improvement instant messaging. Closed systems are characteristically structured around closely related tasks, highly optimized for performance, using database or file structures that include only the data relevant to task performance, and often structured to be optimal for the structure of the applications.

A partial example of such a close system is the Seibel CRM system. Though it does provide APIs for interoperability, its data formats are proprietary. It is unable to reconstruct the full data model outside of the Seibel server because many of the data element associations are built into the server application itself. (Nelson, 2007) OA technical requirements proposed by Nelson conclude that the OA nonfunctional requirements (NFR) contain open standards, modularity, interoperability, extensibility, reusability, composability, and maintainability. In addition, he also shows us the interaction and interdependences of these elements.

Following proposition described above, we would like to borrow the definition into our thesis applying the concept of OA. With the supporting framework of OA for CD, future discussions and applications for CD can be further developed and discussed by combination of various expending level of the three dimensions, which will be clarified more detailed in chapter three and four. The coming section will aim at composing prerequisite elements of CD, which are elementary, qualified, and advanced factors. After that we would like to go the body of this thesis the proposing OA framework for CD.

2.5 Elements of collaborative design

Collaboration seems to be an omnipotent solution dealing with rapid changing global environment. However, when issues come to decision of applying CD, the first thing we need to assure is whether we are capable of put CD into practice.

Decomposing elements of CD, we may find out there exists a hierarchy relations of them. Even we have already perceived CD in our solution pool, which means we would become a CD candidate, there is still a long way to the stage executing it. First of all, there are some elementary factors we should possess to move forward to next floor the qualified factors one. After that, the CD qualified factors make sure the CD process can be taken through, and finally we will be in the activities of CD the advanced factors one, which is the main topic we try deliver in this thesis. (See Figure 2.8)

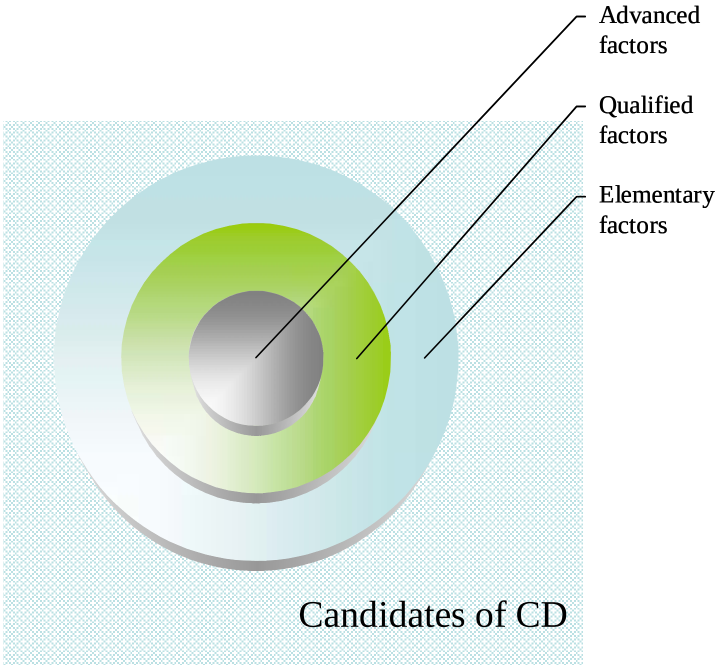


Figure 2.8 Hierarchy of CD elements (Source: this research)

2.5.1 CD elementary factors

Two substances should be taken into consideration are industry characteristics and the infrastructure of information system. The former indicates participants involved focus more on the product and process aspect in improving efficiency and effectiveness of design process for instance. The same object is also vital in this dimension to certain extent. The latter one means level or completeness the infrastructure of IT can support. The IT infrastructure meant here primarily aims at fundamental information system such as PDM or Collaborative Product Definition Management (CPDM) and adoption of software like CAD, CAPP to facilitate design process.

2.5.2 CD qualified factors

With the first screen of CD elementary factors, a step further see whether CD qualified factors are satisfied. If the CD project holding primary qualified factors collected below, then proceedings of CD strategy can be put into our choices pool. Luo et al (2006) proposed a framework model concerning communication network connection among cooperative participants. Based on comments proposed before (Lu and Luh，2001；Panchal, et al., 2006; Chryssolouris, et al., 2006; Luo, 2006), CD qualified factors consisting of four portions which can be summarized as below:

1. Data repository or database with profound contents: which stores the entities of the virtual world together with their attributes, and the database should include the functions like information storage and data retrieval.
2. 3D virtualized tool for product design: such as Dynamic UI Generation, CAD-CAE System, and 3D editor. This tool is a virtual product manager that provides the user with a desktop GUI.
3. Connecting systems for communication and interchanges: indicate systems connected with one another via LAN or internet. Therefore internet, web browser or process diagram tool which is used to model a product realization process, and then invoke the available agents integrated into the framework may be practical tools of CD. The cooperative support platform is another connecting system in this case (Luo, 2006).
4. Powerful Application Service Provider (ASP) Server or core application, which provides functionality for loading, processing and saving the objects of the virtual world.

2.5.3 CD advanced factors

With basic equipment as elementary and qualified factors, discussions of CD will be pulled into another confine. The CD Cube, which composed of three key dimensions, we proposed in the next section, will focus more on the advanced factors to implement CD. In decision dimension, five deciding elements would be mentioned; in design stage dimension, we will find out different characteristics of CD following classification of each stage; finally, in collaboration scope dimension, models of CD would be defined and discussed with various boundaries set of scope.

Chapter 3 Introduction of OA for CD

3.1. Introduction of the Collaborative design (CD) cube

Following CD elements hierarchy illustrated in chapter two, we know that there exist three successive levels of CD factors. With supporting of elementary and necessary levels, the advanced one would be discussed in the following chapter with the CD framework.

3.1.1 Contents of modeling a CD

At first, we will make a short introduction of the contents and dimensions to model a CD activity (see Figure 3.1). To structure a CD framework, we first clarify the situation of proposed CD, including the actors involved, the target objectives for collaboration, our competences in the CD activity, and the resources we have.

After that, we need to make sure we are acknowledged with our CD process model, corresponding CD organization type to carry out the project smoothly, information interchange mechanism, and the collaborative design stage and scope we are involved in. Considering all of these issues, we can build up CD framework containing taxonomy of each composing elements, principles for CD initiator to take into account, and so on.

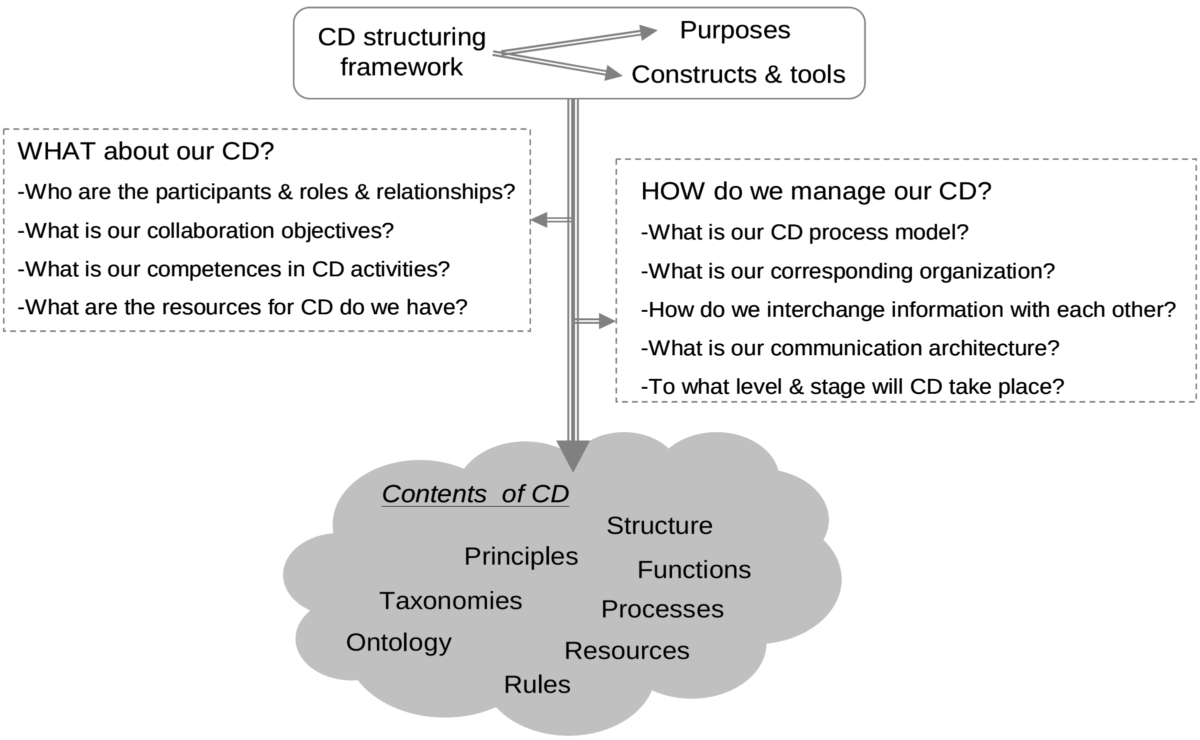


Figure 3.1 Illustration of CD contents (Source: original)

3.1.2 Framework for CD configuration

Three CD dimensions considered when initiating a CD activity are concluded as shown in Figure 3.2. The framework deals with CD issues from three aspects. One is the vital elements needed to be considered, which are composed of participant of CD (see section 3.2.1), CD products (see section 3.2.2), CD process (see section 3.2.3), information of CD (see section 3.2.4), and CD organization (see section 3.2.5).

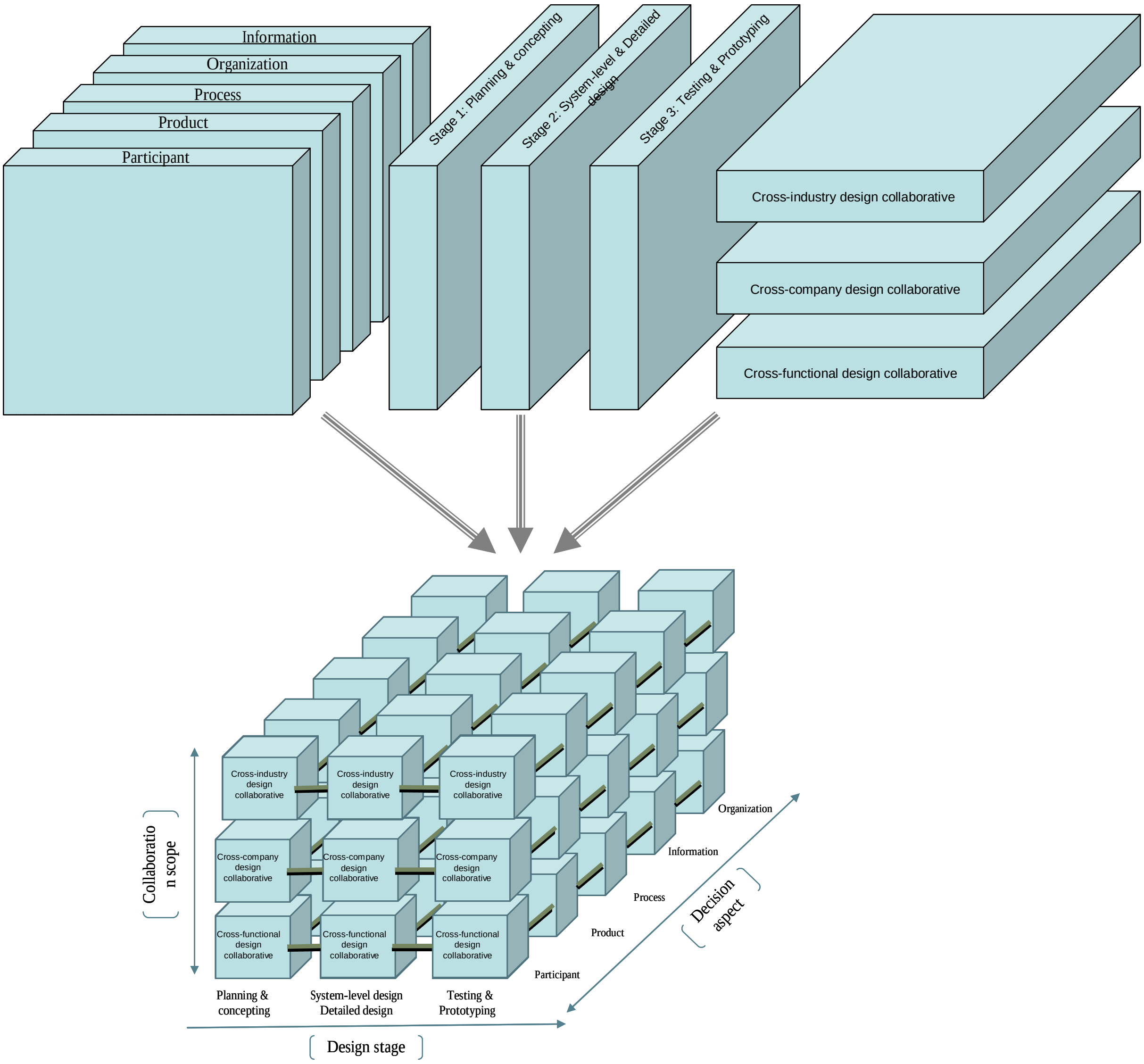


Figure 3.2 concept of constructing CD framework (Source: original)

The other two aspects are concerning which stage is collaboration located on and how far the collaboration scope is expended to. The former stands for aspect of design stages, consisting of planning and concepting stage, system-level and detailed design stage, and testing and prototyping stage, whereas the latter represents the vertical axis, collaboration scope, namely cross-functional, cross-company, and cross-industry design collaboration.

3.1.3 Analysis framework for CD

With this 3*5*3 structure of CD, we can figure out our current status of collaboration, or to see if we have already taken all details into considerations. For example, once the CD of product concept design is decided to be originated across two extended companies, then we would know that circumstances taken into considerations should fall on to the left bar of middle surface of the cube. In addition, if the CD is carried through the product design lifecycle from planning to production ramp-up, then the middle level surface would be our proposed boundary. (See Figure 3.3)

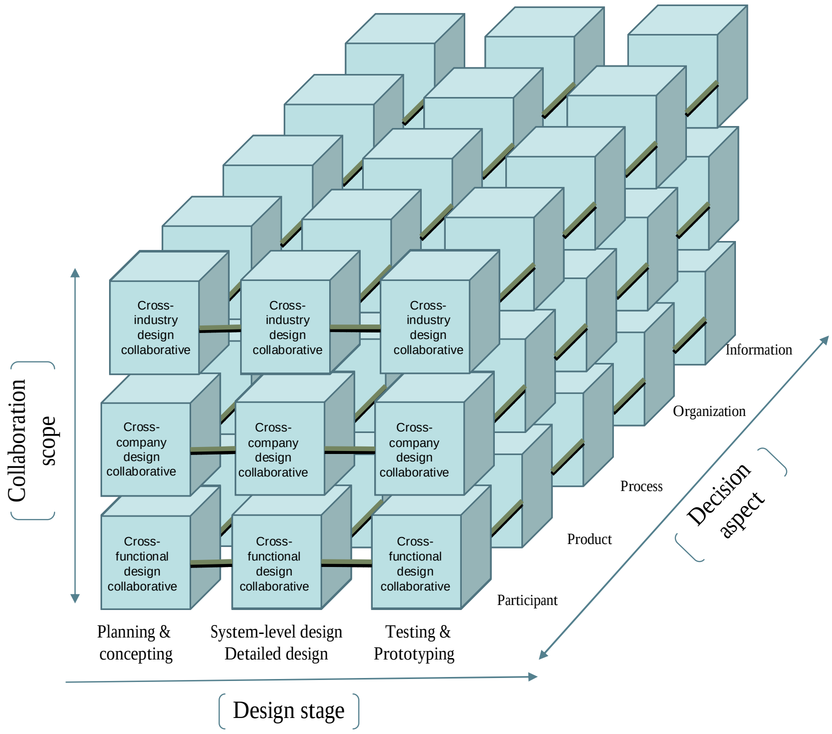


Figure 3.3 Collaborative design Framework: The CD cube (Source: Original)

In Figure 3.4, we see the overall picture of CD framework containing three aspects and corresponding details of each sub-aspect described in the above paragraphs.

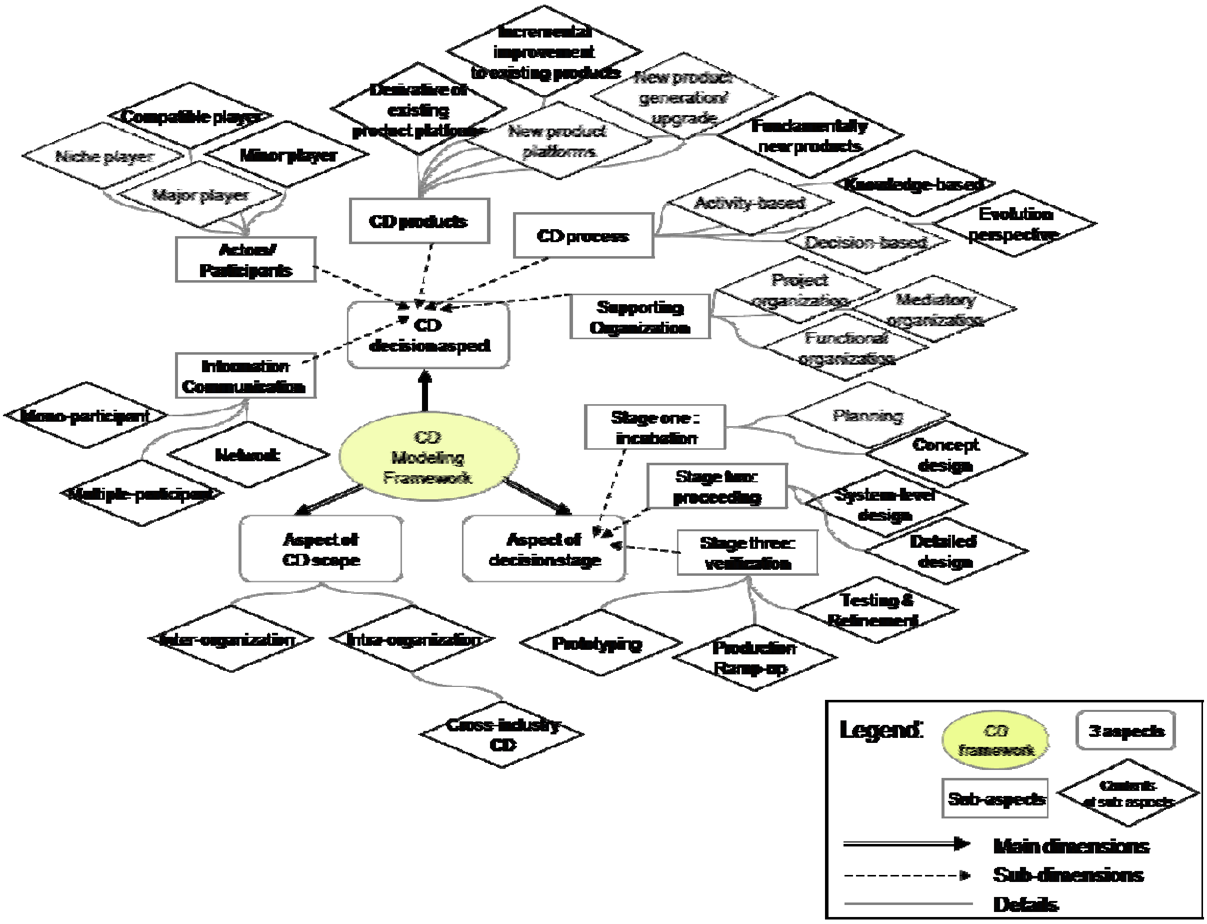


Figure 3.4 CD framework modeling map (Source: original)

3.2 Decision aspect of OA for CD

In section 3.2, we will illustrate the five aspects of decision dimension from session 3.2.1 to 3.2.5 accordingly, which are participant, product, process, information, and organization aspect of CD.

3.2.1. Participants aspect of collaborative design

In section 3.2.1, we plan to give a brief introduction of the concept concerning the roles of initiating and supporting characters, and then make a classification of actor's taxonomy to show readers the composition of CD participants' structures. After that, section 3.2.1.3 shows us examples of CD actors, and two dimensions used for formalizing the participant analysis framework will be delivered in section 3.2.1.4 and 3.2.1.5. At last, the analysis framework is introduced in section 3.1.3.6.

3.2.1.1. Initiating and supporting characters

Parties involved in CD can be separated into two parts initiating characters and supporting ones. Taking cross-functional scope as an example, the initiating characters are like R&D departments, while the supporting ones are like PM, engineering design and manufacturing department. Both can be extended to the scope of cross-company and cross-industry. Taking the supply chain of Dell Inc. as an example in the scope of cross-company, then Dell would be the initiating characters under the new product CD, while its first-tier, second-tier supplier would take the supporting ones.

In the following sessions, we will define the hierarchy of actors (interaction executers of CD), and then discuss about the relationship of actor in CD. At last, we will propose a participant taxonomy field with dimensions of deepness and extensiveness of actors to help clarify decision-making strategy regarding CD participants.

Thomson et al. (2007) show us a brief CD participant scenario of within-company practice with the concept of primary/secondary actors. The lead office play the role of primary actors of CD and the support office is the secondary one (see Figure 3.5). In Figure 3.6, they also provide us a basic concept of the possible interacting circumstances of primary and secondary actors belonging to different offices under the situation of distributed designers with respective project associates of each group.

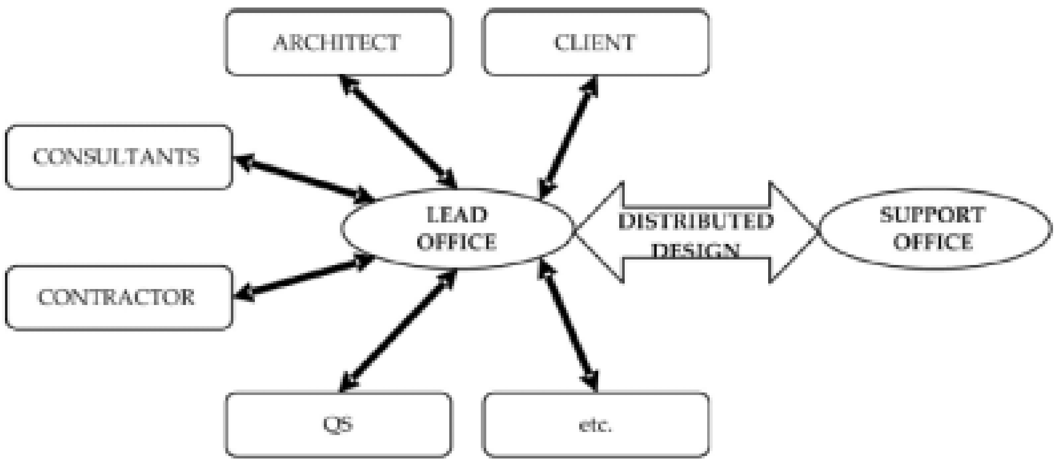


Figure 3.5 Company practice—distributed projects (Source: Thomson et al., 2007)

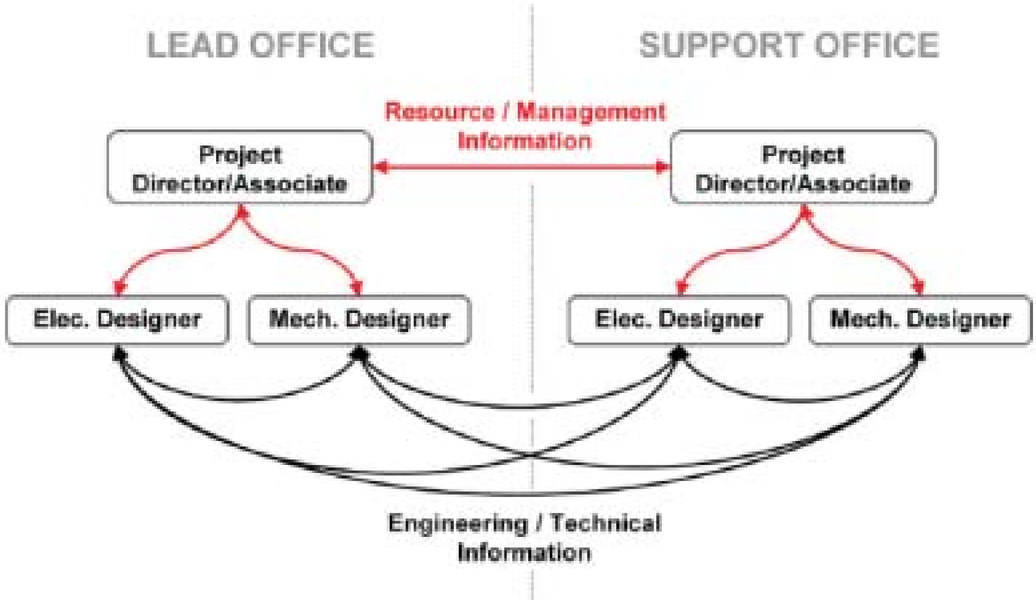


Figure 3.6 The recommended communication structure (Source: Thomson et al., 2007)

3.2.1.2 Actor

We define the representatives participated in communication and information interchanges as actors. Jacobson et al. (Jacobson, et al., 1995 & 1996) define actors as people or objects of the environments which play the role to interact with enterprises or any objects that have to exchange information with systems.

Actors can basically be divided into two categories primary actor and secondary actor (Wu and Lin, 2001). Primary actors are those who initiating the projects or the leading character of CD with advanced technology or superior position, while secondary actors are others except for primary actors which are the same concept of initiating and supporting characters mentioned in last session.

Based on types of agents involved in CD activities discussed by Mahesh et al. (Mahesh, et al. 2006), we combine concepts of actors and agents into the following classification table as shown in Figure 3.7 as the basis to define role of participants and those carrying out CD activities.

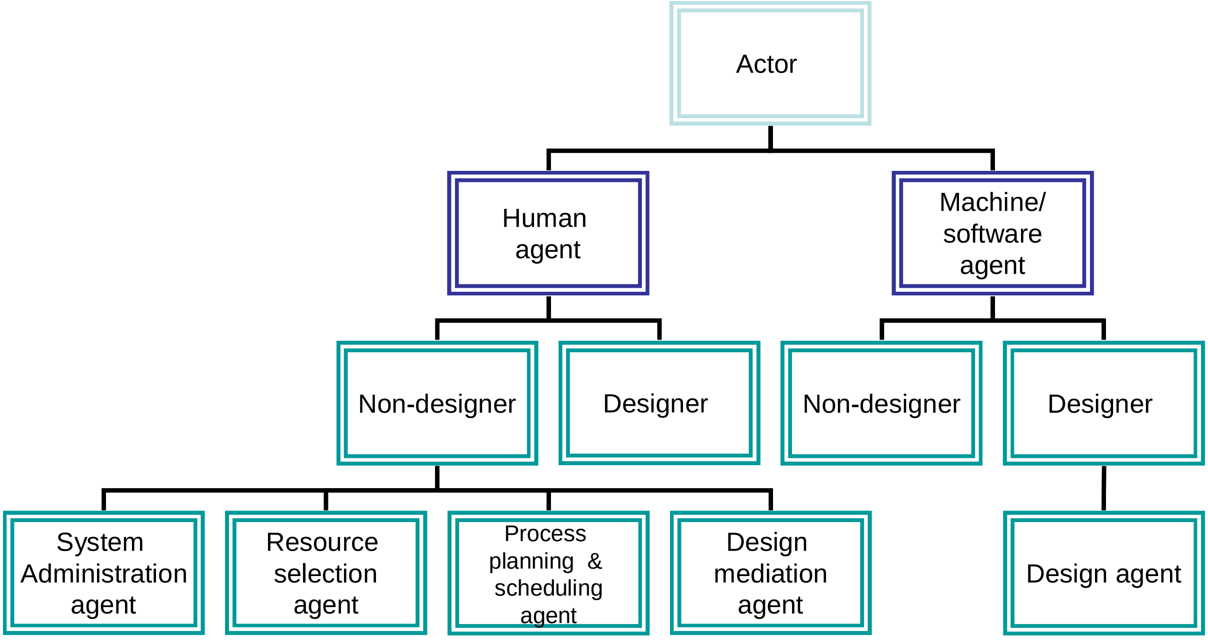


Figure 3.7 Taxonomy of actors (Source: original)

Following Figure 3.7, from the top actor level, we can divide it into two subtypes human agent and machine/ software agent. Inside this level, agents can be categorized into as designers and non-designers in view of tasks they carry out. Designers would basically do the design job during CD process, whereas non-designers would provide supporting functions and put into the following characters by their job domain. System administration agent would be responsible for stabilizing system operation and administrative works; resource select agents would be responsible for allocating resource among cooperators, coordinating resource applications, and maintaining or updating resource applying conditions. Details of how to build up agents, agent architecture, agent planning, and related engineering and communication issues regarding agents have been widely discussed by previous studies (Muller, 1997).

Examples like design project manager are defined by Brigitte (2003) shown in Table 3.1; process planning and scheduling agents are responsible for coordinating tasks among cooperators and make sure of CD process to move smoothly; design mediation agents are mainly concentrating on mediating conflicts among cooperative designers on design tasks. For example, if we regard AutoCAD as the design agents responsible for design tasks in detail design stage, then PM system or software can be viewed as the system administration agent in our domain. The role of design mediation here maybe be played by a project manager; the ERP system can do resource selection agent character, and CD process scheduling package software may be the process planning and scheduling agent we meant here.

Therefore, we may conclude that main characters involved in CD activities should be composed of the five kinds of agent, which are design agent, system administration agent, resource selection agents, process planning and scheduling agents, and design mediation agents.

3.2.1.3 Examples

Taking the matrix proposed by Brigitte (2003) illustrating design integration in a company as an example scenario, which is as shown in Table 3-1, Keinoneon (2006) concludes two main approaches for forming teams. Forming principals can be taken into three perspectives. One is by the view of concerning what functions should be involved, which refers to the necessary professional expertise to recruit. The second one focuses on individual problem-solving styles and social behaviors within team members. The third one addresses on task responsibility each role should take.

Table 3.1 the matrix of design integration in one company (Source: Brigitte, 2003)

Design Function	Graphic design	Package design	Product design	Environmental design
CEO	Corporate identity		Innovation	Work spaces/ factory
Corporate communication	Corporate identity			Event/trade show/ welcome area
R&D production	Technical documentation	Logistics packaging	Innovation	Factory
Marketing	Brand graphics Web sites	Packaging product/ promotion	Product range	Trade show/ store

Under the scenario following the matrix of designer career path illustrated by Brigitte (2003) as an example shown in Table 3.2, as we can tell that even within the same design domain, with different functions of design tasks such as product, packages, and so on, the role of designer agents can be divided into more specific types of agents with respect to the functions it required to deliver. For example, the role of designer as the design agents can be further divided into associated designer agent or assistant designer agent of product design, or even within different design domain such as associated designer agent of graphic design and assistant designer agent of package design. In this case, CD would be carried out within the design-specific team without non-designers, which can be one of the CD scenarios.

Table 3.2 the designer's career path (Source: Brigitte, 2003)

	Job title	Responsibility
Designer	Associate designer, assistant designer, Designer	Developing creative solutions to design problems
Design project manager	Senior designer, Project manager, Associate design directors	Coordinating resources in order to deliver a deign within a predetermined schedule and budget
Design staff manager	Creative director, Studio leader	Managing design staff, transferring design strategy into creative briefs, and assembling design teams to meet project needs.
Design organization manager	Director, Principal	Making operational and general management decisions that drive the development of a design group or organization
Strategic design manager	Chief design officer, CEO	Developing the organization ' s strategic business objective, along with the related design strategies that help meet the goals

To deal with actors' relationship in CD, we try to probe this issue by two manners, namely the depth of interactions done in CD and the breadth of interactions done in CD. The former measures the intensity of cooperators' relationship, while the latter try to figure out the level of extensiveness which the collaborators can carry out to. (Adapted from Chen, CSD Review 2008.1 p.46)

3.2.1.4 Deepness of CD among cooperators

In the part of the depth of CD among cooperators, we can take a further step from the following four directions collaboration scale, business models of participants, predominating degree of supporting characters in CD, and the degree of supporting characters' involvement in CD.

We will take the initiator role in illustrating the following contents. In collaboration scale, we have to ascertain that to what extent are the we going to collaborate with supporters as an initiator? To what level we are going to share our information and know-how? How many cooperators should be involved in? Do we have enough resources in having this kind of collaboration scale?

All of these decisions should be taken into account thoroughly, and will be intensively connected to the following four aspects of CD. However, the criteria in measuring the size of collaboration scale would be significantly diverse according to what industry and what kind of product are we talking about, but the basic thinking of this part is approximately as above, which are the number of participants, information sharing degree, and the size of resource spent in.

In the business model, we use the classification defined by DoIT (DoIT, 2005). It consists of OEM (Original Equipment Manufacturing), ODM (Own/Original Design Manufacturing), OBM (Own Brand Manufacturing), and CDM (Collaborative Design Model). This would relate to the interaction models of CD process, which would be introduced in details in section 3.2.3.

In the predominating degree of supporting characters in CD, we mainly want to assess on what level of activeness are the supporting characters in CD. We can classify them (supporting characters) by their co-design involvement situations, which are including three levels of co-designing by the given specification, of co-designing mutually on the same table, or of co-designing by providing design specifications or patterns actively. The last one represents the most highly involved in CD, while the first means the least level of involvement in CD.

In the degree of supporting characters' involvement in CD, we meant it in two ways, which are the ODM percentage of the whole CD project and percentage of positive responses or feasible proposals provided by supporting characters according to requests made by initiators.

3.2.1.5 Extensiveness of CD among cooperators

In this section, "extensiveness" meant how flexible the supporting characters can and are willing to cooperate with initiators and the extent of supporting characters' capability to accomplish/execute the tasks of its own completely and integrally. Directions of extensiveness of CD among cooperators are the efficiency of supporting characters in designing operations and process, supporting characters' degree of accuracy in CD (such as how exactly can the supporting characters capture the idea of product concept design delivered by the initiators and put it into practice), and capability of supporting characters to integrate information during CD activities.

To learn the capability of supporting characters, we can tell it from three perspectives: (1) R&D material managements, (2) types of project managements, and (3) communications within the suppliers and customers.

The manner they deal with their R&D materials can be in hard copies, in digital files, by file server, or by PDM (Product Data Management) or by PLM (Product Life-cycle Management), which is the most agile one dealing with

R&D materials for material reuses and information sharing. The manner they deal with their PM can tell whether they are well-prepared or not for applying Information Technology (IT) doing PM in view of efficiency and effectiveness in exchanging information among cooperators. Methods of managing PM are in hard copies, in digital files, by file server, or by PM system.

The third one is the manner they communicate with each other in CD can also affect the extensiveness their collaboration can reach. We introduce two measuring dimensions which are synchronous-asynchronous and convenient-inconvenient, and there we can have four combinations of these two axes, which are synchronous-convenient, synchronous-inconvenient, asynchronous- convenient, asynchronous-inconvenient. In the light of easiness of communication, we may see that the synchronous-convenient policy would have a better result than the asynchronous-inconvenient policy in communications and CD project execution. In addition, this kind of policy applied by cooperators of CD can tell their relationship of reliance and trust with each other.

3.2.1.6 Participant classification for CD

With the horizontal axis of deepness and vertical axis of extensiveness, the two-by-two matrix is produced to clarify the participant types involved in CD in a view of primary actors for decision making. We have four types of CD participants, which are major, niche, compatible, and minor player. Before originating a CD activity, we have to make sure the role of each player involved in the game. (See Figure 3.9)

Major Player represents deep connection of cooperators and advanced extensiveness of proposed collaboration. This kind of participant focuses on fundamental CD or NPD (new product development/ design), and is prone to adopt the manner of integrated corporation, joint venture, or extended enterprise.

Niche player represents deep connection of cooperators but limited extensiveness of proposed collaboration. This kind of participant possesses certain dominant or specific skills or technology that primary actors are interested in. For this reason, corresponding collaboration target would be professional NP or under the mature competence market applying the manner of VE, VO, etc.

Compatible player represents shallow connection of cooperators but advanced extensiveness of proposed collaboration. This kind of participant is capable of providing non-fundamental but customized or domain-specific NP with manner of contractual agreement, which is the same constitute manner applied by minor player. However, two of them can be discriminated easily by the product or service they are collaborative designed. Take Foxconn for example, although Foxconn seems like a minor player providing fundamental OEM jobs with many other potential competitors to their customers, it does solid its status on the market for its specific or customized capability to fulfill the needs of customers such as QCD etc and make a compatible rather minor.

Minor player represents shallow connection of cooperators with limited extensiveness of proposed collaboration. This kind of participant has the most potential risks in being replaced by competitors easily for lack of the sustainability with each other of close relations or extraordinary capability. They usually participate in general/ routine NPD or take the OEM role to their ODM/OBM customers.

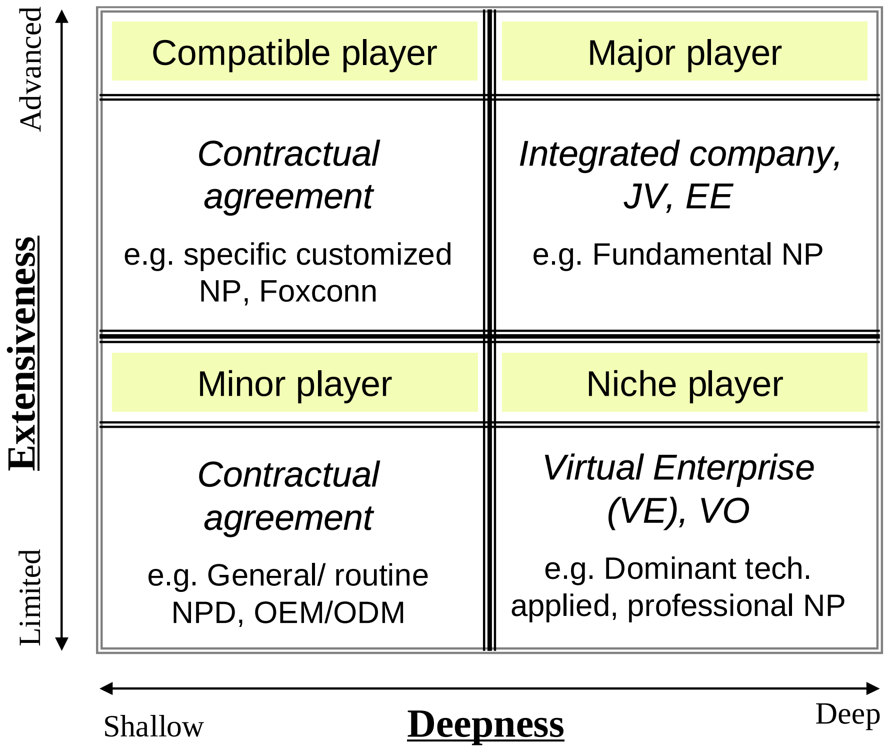


Figure 3.9 CD participant classification matrix (Source: original)

3.2.2 Products aspect of collaborative design

We would like to define product design first, and then move on to the classification of product development types and design product types from many perspectives, including in the view of product development, project team structure, product flexibility, characteristics of product properties, nature of technology, market maturity, and tendency of corporate/ industry strategy.

According to Keinoneon (2006), the word “product” in product development literature usually refers to an object of exchange brought to the marketplace. When product design related tasks are carried out without the objective of providing documentation for production that will eventually lead to market launch. There should be a conceptual differentiation from the core meaning of product design. We call these activities concept design, concepting, or product concepting."

Based on the proposition of Ulrich and Eppinger (2008), they give us a summary of product types in two different way, one is in the view of product development process and the other is corresponding to the product development projects types adopted. The description and details of product type classified in the product development process is showed in Table 3.3.

3.2.2.1 Product types in view of product development process

Corresponding to different types of product, the organization and product design process should also be revised to cohere with the characteristics of that product type. Followed by the summary of eight product types, Ulrich and Eppinger (2008) generally divide them into three main categories and provide us the corresponding product development process, namely generic product development process, spiral product development process, complex system product development process.

Generic product development process includes general types of product such as Market-Pull products, Technology-push products, Platform products, Process-intensive products, Customized products and High-risk products. Each phase of product development is completed and to determine whether the process proceeds.

The second one is spiral product development process, which is fitted to be adopted by the Quick-build products since it enables them to go through the process of detailed design, prototyping, and testing for a number of time.

The last one would be complex system product development process which enable for a complex system to decompose into parallel stages of work on the many subsystems and components. Process flow diagram for three product development processes is shown in Figure.

Table 3.3 Summary of variants of generic product development process (Source: Ulrich and Eppinger, 2008)

Types	**Description**	**Distinct Features**	**Examples**
Generic (Market-Pull) products	The team begins with a market opportunity and select appropriate technologies to meet customer needs.	Process generally includes district planning, concept development, system-level design, testing and refinement, and production ramp-up phases.	Sporting goods, furniture tools
Technology-push products	The team begins with a new technology, and then finds an appropriate market.	Planning phase involves matching technology and market. Concept development assumes a given technology.	Gore-tex rainwear, Tyvek envelopes
Platform products	The team assumes the new product will be built around an established technological subsystem.	Concept development assumes a proven technological platform.	Consumer electronics, computers, printers
Process-intensive products	Characteristics of the products are highly constrained by the production process.	Either an existing production process must be specified form the start, or both product and process must be developed together from the start.	Snack foods, chemicals, semiconductors.
Customized products	New product is slight variation of existing configurations.	Similarity of projects allows for a streamlined and highly structured development process.	Motors, switches, batteries, containers
High-risk products	Technical or market uncertainties create high risks of failure.	Risks are identified early and tracked throughout the process. Analysis and testing activities take as early as possible.	Space systems, pharmaceuticals

Types	Description	Distinct Features	Examples
Quick-build products	Rapid modeling and prototyping enables many design-build-test cycles	Detail design and testing phases are repeated a number of times until the product is complete or time/budget is run out.	Software, cellular phone
Complex systems	Systems must be decomposes into several subsystems and many components	Subsystems and components are developed by many teams working in parallel, followed by system integration and validation.	Airplanes, jet engine, automobiles

Generic products are the most common type faced, this type of product go through the whole development process, and usually is taken as ordinary project/ cases like annual product launch etc.. Generic product belongs to normal goods properties, though it still can use CD for product design and development process, CD is definitively not the only way to launch it.

Since technology-push products have to consider both marketing and technology applied issues, it may require CD for early involvement from the planning and concepting stages, thus this kind of products usually adopt CD at the first stage, whether cross-functional or cross-companies.

Platform products and Process-intensive products have already set to follow the existing product design and development technology or process, therefore, unless for specific specification or special circumstances, CD may be too complicated to be adopted for them.

Customized products, High-risk products, Quick-build products, and complex products are all reasoned for applying CD in the early stages of design and development process for both efficiency and effectiveness considerations.

3.2.2.2 Product types in view of product development project applied

In accordance with Ulrich and Eppinger (2008), there are four main types of product development projects, namely new product platforms, derivative of existing product platforms, incremental improvement to existing products, and fundamentally new products. Corresponding to these four types of project/ platforms, types of product being designed/ projected can also been classified. Summary of four product types in the view of development projects types is shown in Table 3.4.

In view of CD adopted territory, new product platforms and fundamentally new platforms are two kinds of product development projects for which products are more easily to be developed through CD for risk and integration considerations, especially the fundamentally new product for its new challenges to market demand, manufacturing capability, and technology applied consideration. In conclusion, all four types could take place through CD for certain concerns, but what we care about is the trend and properties for which kind of product should apply CD for product design realization, in view of risk preference of these four cases, new product platforms and fundamentally new platforms products are chosen, and the former bear more uncertainty than the latter. In other words, the level of new product platforms to apply CD is higher than fundamentally new platforms products.

Table 3.4 Four types of product development projects (Source: Ulrich and Eppinger, 2008)

Types	Description	Example
New product platforms	Involving a major development effort to create a new family of products based on a new, common platform. The new product would address familiar market and product categories.	The Xerox Lakes project, aimed at the development of a new, digital copier platform.
Derivative of existing product platforms	These projects extend an existing product platform to better address familiar markets with one or more new products.	To develop a new copier based on an existing light-lens (not digital) product platform.
Incremental improvement to existing products	These projects only involve adding or modifying some feature of existing products in order to keep the product line current and competitive.	A slight change to remedy minor flaws in an existing copier product.
Fundamentally new products	These projects involve radically different products or production technologies and may help to address new and unfamiliar markets. Such projects inherently involve more risks; the long-term success of the enterprise may depend on what is learned through these important projects.	The first digital copier Xerox developed.

3.2.2.3 Product types in view of concept orientation

Sääskilahti and Takala (2006) make a proposition of two types of design concept development product-oriented and system-oriented. The former focuses more on conventional market and technology, and centers on design trend and current user needs, while the latter aims at changing market and technology, business position, and centers on predicting current user needs and preferences. They conclude that the more one possessing characteristics of system-oriented, the more importance of product concept design would be. This proposition indicates that there exist positive relations of product characteristics and the role of product concept design.

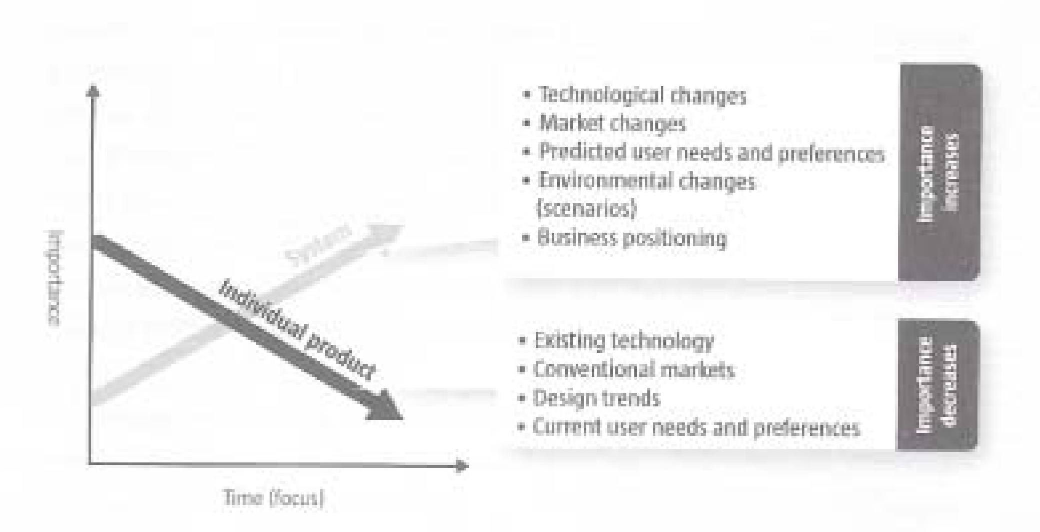


Figure 3.10 Characteristics of product-oriented vs. system-oriented concept development (Sääskilahti and Takala, 2006)

We can infer that the differences of concept development can also affect product design types. Since CD is primarily focusing on the portion of product co-design and development, we can propose that the more product-oriented is the CD target, the less possibility collaboration would focus on the front stages of whole product design and development process. In other words, the importance of CD would increase as the product development characteristics squint toward system-oriented products. Therefore, the characteristics of product designed would influence the way CD process is carried out in this way.

However, from the contents of Figure 3.10, as time goes by, the importance of system would become more and more significant. Hence, in our inferences, CD may take place to coordinate the gray zone of characteristics coping with product strategy. We shall not take the two characteristics, namely product-oriented and system-oriented PD, as individual ones, but may have interrelation within product development process from product-oriented to system-oriented one, or they may have overlapped periods of the PLC, and CD would play a more vital role in this case to build up the communication bridge of individual projects or participants, which reveal the interactions of the five decision factors of CD decision dimensions.

Following two sections would focus more on the details of respective key elements technology used and market targeted of product dealing with topics of CD.

3.2.2.4 Product types in view of technology used

After defining the types of product types from the above different view, we can take products from the other aspect, product technology used. Teece (1986) gives us clear key dimensions of the nature of technology consisting of Tacit, Codified, Product and Process. The technologies involved can barely classify into product technology and process technology, within this field of classification, both of these two technology can deeply divided into tacit- and codified-knowledge technology, which may play a significant role in affecting the ease of imitation, thus influence the way design collaboration take into practices.

Codified knowledge edge is easier to transmit and receive, and is more exposed to industrial espionage and the like, while tacit knowledge by definition is difficult to articulate, and so transfer is hard unless those who possess the know-how in question can demonstrate it to others. (Teece, 1981) The classification of product type in the view of nature of technology is summarized as following Figure 3.11.

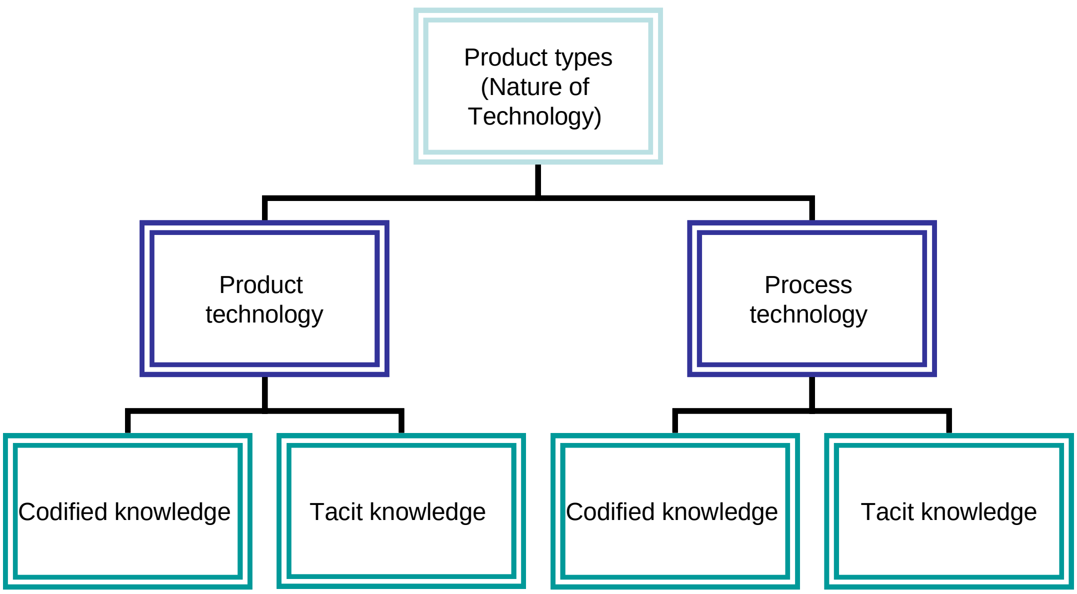


Figure 3.11 Classification of product type in the view of nature of technology (Source: adapted from Teece, 1981)

3.2.2.5 Product development strategy in view of product technology and target market

Sääskilahti and Takala (2006) propose the business-positioning field diagram for illustrating the links between vision concept choices and business strategy to make the appropriate concept-selection decision as shown in Figure. The horizontal axis represents market uncertainty, and the vertical axis is the technological changes. They define these four main business districts among the technical and market uncertainty field. Therefore if we want to stay within our existing technical competences but extend to new markets, we would like to choose the lower right-hand corner. With this technical-market framework, we can infer from this as our analysis base, and combine this structure with the four main product types illustrated by Ulrich and Eppinger. The field of product types diagram would be shown in Figure 3.12.

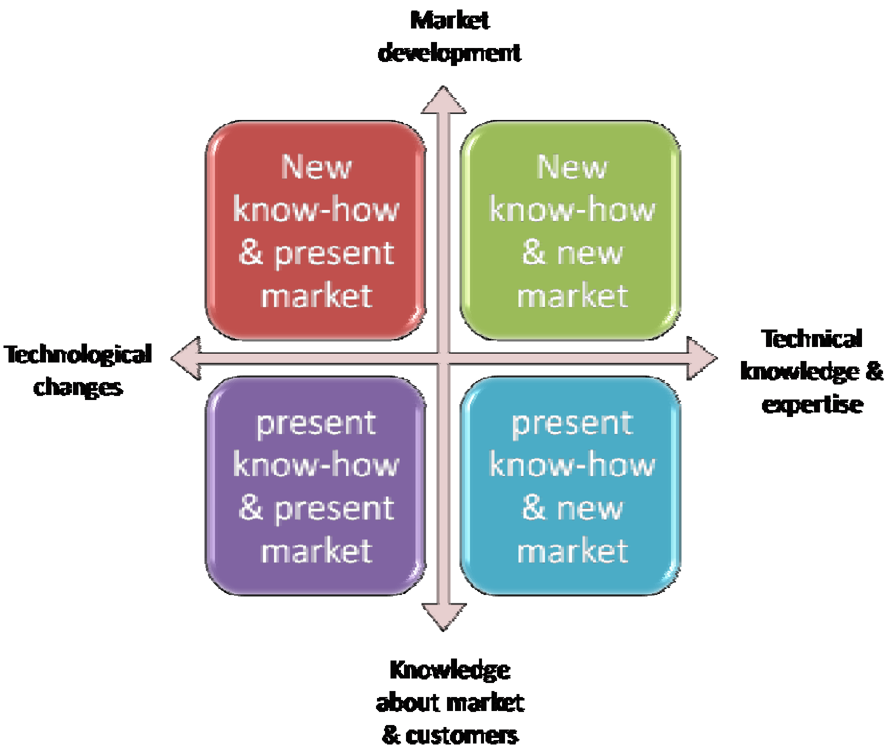


Figure 3.12 business-positioning field (Source: adapted from Sääskilahti and Takala, 2006)

We define the vertical axis represents the market status consisting of current market and new market, and the horizontal axis represents the current technology and new technology. In the very first place, we hypothesize our product in the status of current market using current technology. Then we can define four main product types in the view of product development projects under this structure as shown in Figure 3.13. When corporation strategy is the current market-current technology, Incremental improvement to existing products would be the product type. When corporation strategy is the new market-current technology, Derivative of existing product platforms would be the product type. When corporation strategy is the current market- new technology, new product platforms or new product generation/ upgrade would be the product type. In this region, we may still in the same product family but using different product platform in product realization. On the other hand, we also introduce the new product generation/ upgrade type in this corner, for this is also fitting in the conditions of under the same product family with different or update technology. When corporation strategy is the new market- new technology, fundamentally new products would be the product type.

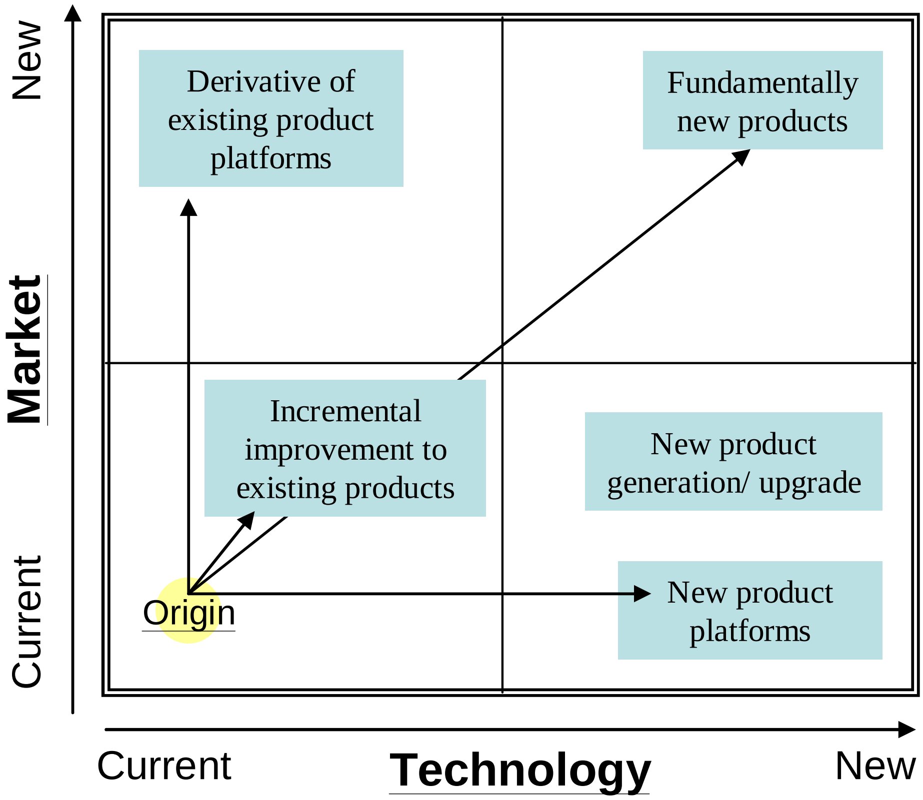


Figure 3 Product type strategy field (Source: Original)

3.2.3 Process aspect of collaborative design

Processes play an important role in CD, which can be prove by a few design companies that have developed their businesses from a well-defined concept and product creation process, and we can take IDEO's process as a example (Figure 3.17). A process shown below is kind of general case in product design stage. The process begins with obtaining an understanding of the design challenge, and then subsequently tested.

After gaining a relatively understanding of the current activities, then the next step is to start outlining new solutions within the iterative cycle of visualizing, modeling, evaluating, and refining. Rapid prototyping techniques can be used several times to improve the ideas that are developed before the solutions are communicated to the customers within team members (Takala et al., 2006). However, following the process similar as described above; each gap of process stage can possibly miss some vital information, concepts or consensus that would lead to misunderstandings or conflicts in the later stage. Next section would focus on composing elements of modeling design process first.

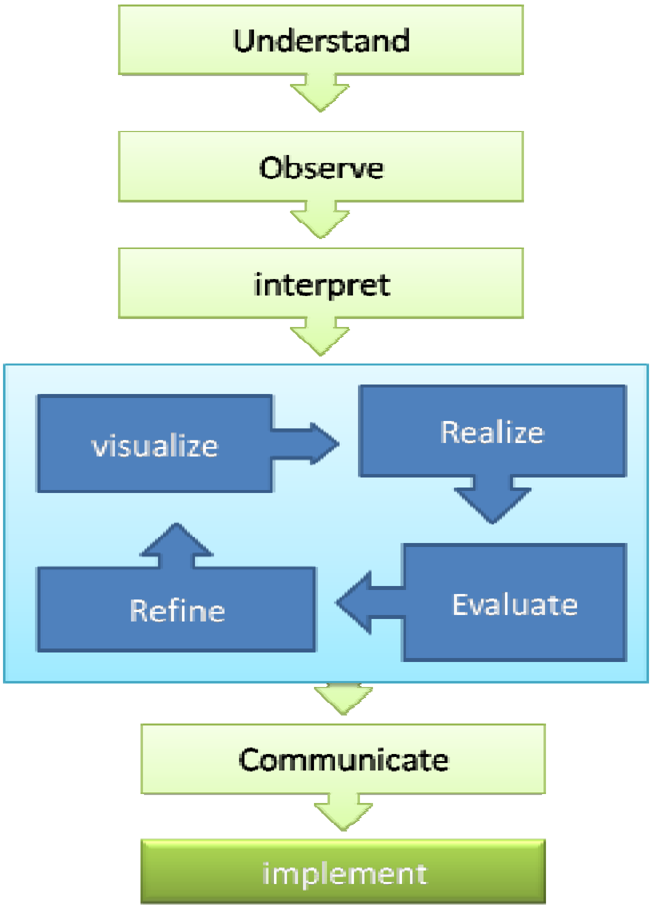


Figure 3.17 The IDEO Deep Dive process (Source: adapted from Takala et al., 2006)

3.2.3.1 The process of collaborative product design/development (CPD)

According to Wang (2006), CPD means two or more firms working together to develop and commercialize a specialized product. In a traditional collaborative design process, designers would work together to develop a single and complex design, these designers share data to solve the design problems in order to reduce the repetitious discussion. Since a collaborative design team often works in parallel and independently with different engineering tools distributed in separated locations, even across various time zones around the world, the resulting design process may then be called collaborative design process (Wang et al., 2002). The CPD process defines what activities are performed by whom, when they are performed, and how they are performed. By constituting a distributed product development network, the Internet-based collaborative product development chain (ICPDC) connects product development parties, including suppliers, assemblers, and customers, with a computer network so all the activities associated with product development can be carried out in the ICPDC. (See Figure 3.18)

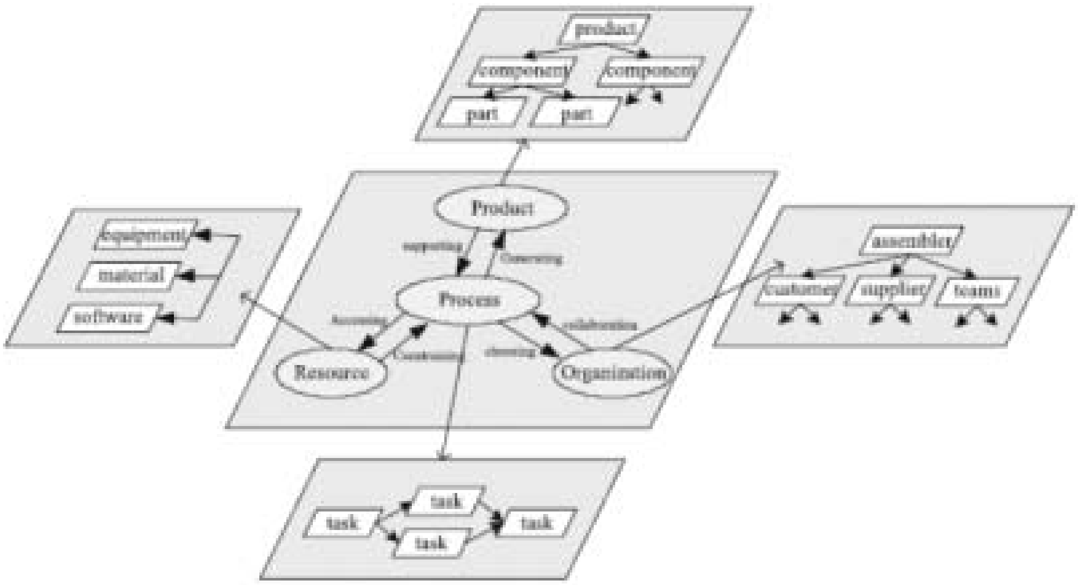


Figure 3.18 ICPDC integrated model (Source: Yang, 2006)

Collaborative product development process supported by ICPDC is clearly depicted. The horizontal axis means collaboration in the time dimension and the product development activity is executed in a sequence. The vertical axis means collaboration in the space dimension and task is decomposed to carry out at geographically dispersed units. (See Figure 3.19, Yang 2006)

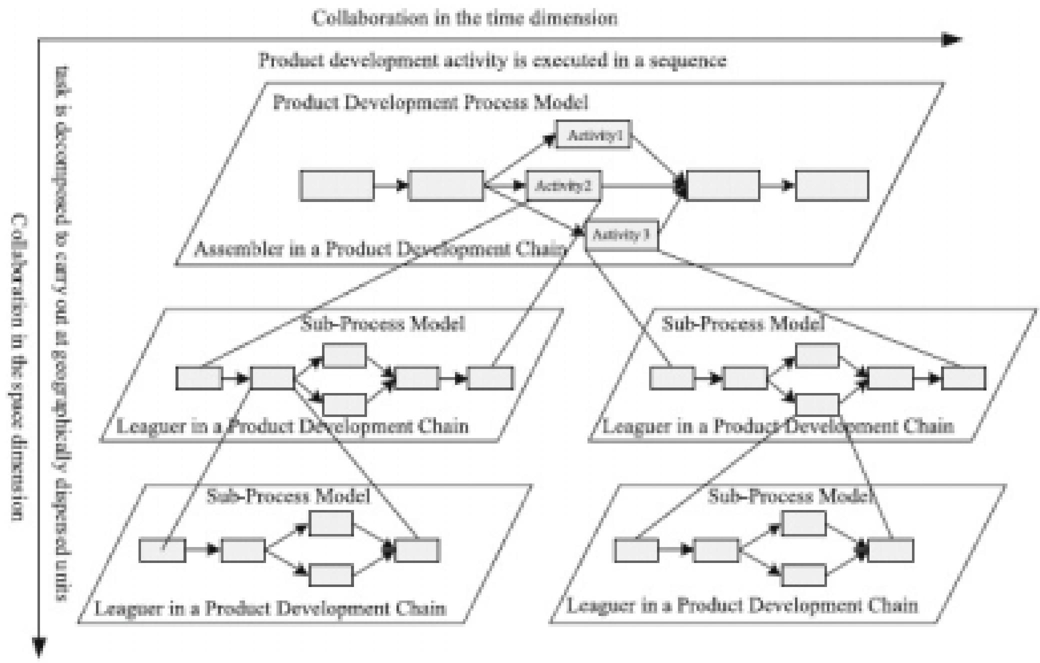


Figure 3.19 Collaborative product development process supported by ICPDC (Source: Yang, 2006)

3.2.3.2 Components of design process modeling

Panchal et al. (2006) provide a reference framework of important requirements in enabling the design of design processes described in following section. The design process modeling should basically have the following five main functions: information transformation support, decision-making support, design process configuration, design process analysis and design process synthesis.

Support for design information transformation means the design process design for the collaboration should aware of identification of design process goal, supporting the information transformation among multiple discipline cooperators and the storage issue of design information and design process information (information storage would be specified in details in the information session).

Support for design decision-making means design process should be set up considering the support of process related decisions, information transformations among different design stages or teams during the process, and computational model thereof selected to develop the design process.

Design process configuration part means we should have the modeling or configuration technique such as rapid prototyping technology and representation knowledge for designing the target process of particular product process.

Analysis of the analysis of design process function means defining or clarifying the transformation data or information that maps the physical product form to its processing behavior and quantification for design process impact on collaboration. Examples are like the information processed and reprocessed of the FEA result data from engineering department/team to product structure department/team (depending on the scope of collaboration), and quantify the effect on the whole product performance of each product process has such as the 20-80 rule applied on deciding the product cost and on the degree of difficulty to define the product architecture (80 percent of the whole product structure and cost has already been decided in the early 20 percent design stage, which tells us the importance of early involvement and collaboration of design).

The last one is synthesis of design process. Synthesis is a mapping from expected behavior to the product form, thus balancing the integration of product- and process-centric perspectives would be an issue of this part.

3.2.3.3 Modeling levels of process modeling

Browsing these four process models types in view of its business model, we would like to take a further look on the process modeling framework for having a whole picture on the topics of modeling design process. Architecture of process modeling framework is proposed by Panchal, et al., (2006) defining the modeling process by three main levels, namely process specification level, declarative level, and execution level as shown in Figure 3.20.

In the process specification layer, required information transformations and required information flows are specified accordingly. In other words, it is mechanics of information transfer that is captured in this level, and leaving problem specific information being defined at the declarative level. In the declarative layer, problem formulation related information is capture, and the independence of information fro process mechanics is also guaranteed. In the execution layer, the details of code execution are captured, and this level s specific to the design scenario and problem for which the process is used.

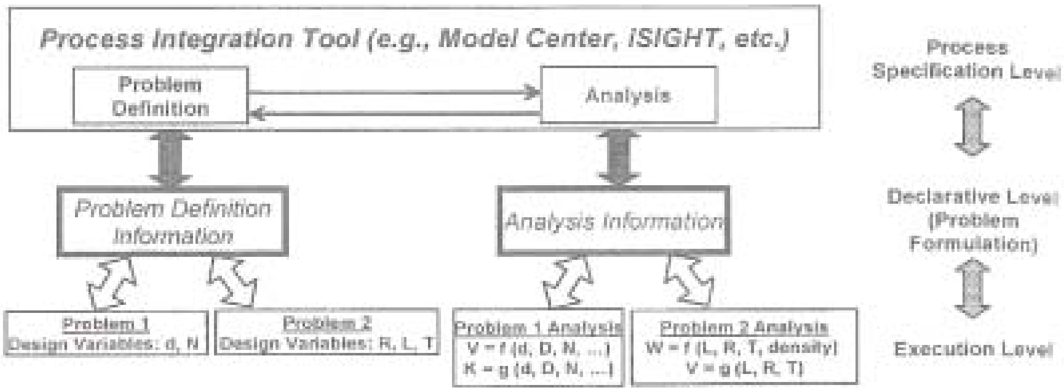


Figure 3.20 Architecture of process modeling framework (Source: Panchal, et al., 2006)

3.2.3.4 Types of workflow (in general cases)

About the types of product design cooperation frameworks of different business models, we would like to know more about the process types or process characteristics regarding our core issue---design collaboration. Before we introduce the basic properties for design collaboration process, let us have a eye on the classification of workflow pattern which was developed by Workflow Patterns Initiative. (http://www.workflowpatterns.com/) They categorize all the patterns of workflow among different industry and business, and total workflow pattern would basically sum up to 42 types.

These 42 workflow types can be mainly put into four large-scale categories, namely control-flow perspective, data perspective, resource perspective, and exception handling perspective. The control-flow perspective captures aspects related to control-flow dependencies between various tasks (e.g. parallelism, choice, synchronization etc). Originally twenty patterns were proposed for this perspective, but in the latest iteration this has grown to over forty patterns. The data perspective deals with the passing of information , scoping of variables, etc, while the resource perspective deals with resource to task allocation, delegation, etc. Finally the patterns for the exception handling perspective deal with the various causes of exceptions and the various actions that needs to be taken as a result of exceptions occurring.

3.2.3.5 Key issues related to collaboration design process

Now that we have a basic knowledge of workflow patterns, we can take a further step to collaboration design process. First of all, we will borrow the viewpoint provided by Khanna et al. of parallel computation system, and then look up for the essential features of microprocessor computing system proposed by Gajski and Peir, then take the proposition of analogies of parallel computation system with collaborative engineering design system illustrated by

Khanna et al., at last we would give a summary of CD process classification and give a taxonomy of design process strategy of our own viewpoint.

Khanna et al. (1998) provide two main types of parallel computing system SIMD and MIMD to make the analogy with engineering design systems. SIMD (Single Instruction Multiple Data) is the situation that either all processors are working on the same tasks or no task at all whereas MIMD (Multiple Instruction Multiple Data) is the situation that each processor gets its own piece of the program and executes it independently of other processors.

Gajski and Peir (1985) list five essential features of multiprocessor computing system, which are model of computation, program partitioning, scheduling, synchronization, and memory access. Khanna et al. make an analogy with the engineering design environment, which is as Figure 3.21 shown below.

Feature	*Parallel computing example*	*Analogy with engineering design systems*
1. Model of computation	(a) Control driven	(a) Integration in sequence
	(b) Pattern driven	(b) Integration according to a prescribed order
	(c) Data driven	(c) As the information is available, each process is executed
	(d) Demand driven	(d) As the information from a process is needed, that process is executed
2. Program partitioning	Each processor can process one employee's payroll in an accounting organization	Teams are made up of people with different expertise so that they can each be assigned a different part of the project; some can design mechanical parts, others, electronic components
3. Scheduling and resource allocation	Task-to-resource allocation problem; an example is deciding which and when program block will execute on which processor	Analogy is trivial; designers and engineers are constantly facing scheduling and resource allocation problems
4. Synchronization	Avoid deadlocks, conflicts, and out-of-resource usage; results must be available when needed	Accomplishing given project tasks under deadline constraints so they can be integrated with other parts of the project
5. Memory access	When many processors need to use the same memory, there is the risk of long waiting times that slow down computation	For example, the same drawing or expert engineer may need to be accessed; however, if one designer is working with them then no other may have access to the most up-to-date information

Figure 3.21 Major analogies from parallel computing with the engineering design environment (Source: Khanna et al., 1998)

From the view of Khanna et al., we may conclude that under the parallel engineering design environment, the following issues are necessary to be proposed, and one of them is that what is the composing attributes regarding the CD process. In terms of computation model illustrated by Gajski and Peir, four parallel models have been provide, namely control driven model, pattern driven model, data driven model, and demand driven model.

This viewpoint has been expended by Khanna et al., and used to define four main system models integrated separately in sequence, prescribed order (pattern), status of available information, and the demand of information. Hence the process formation discipline would be one of the necessary components in discussing CD process.

The second one is the component decomposition in view of program partitioning mentioned by Gajski and Peir. Khanna et al. expend the thinking into the team member, design function, and the design process, which can be inferred as the proposition that it is necessary to decompose the contents of teams (e.g. multidiscipline functions and members), information flow, and engineering design processes.

The third one would be related to the issues of memory access or data/information repository applying mechanism. When talking about designing the design process, the mechanism dealing with using the same repository among appliers simultaneously should also be taken into consideration, and this part would be discussed in the section 3.1.5 of this chapter.

Therefore if we can map this kind of characteristic of computing system into the collaboration process, we can have two main types of CD process. One is that each agent of the design collaboration team is doing their work independently, and the other would be all or some of the agents within the same design collaboration are working dependently.

Following parallel computing system principle, we may discuss design process strategy by two main sections, one is from the relationship of the dependence of design processes, and the other is exactly from the design process itself which means we can summarize types of design process from the view of the process flow itself. In the following session, we will give a summary of design process classification in two different points mentioned above, namely in view of design process tasks and in view of dependence of design processes.

3.2.3.6 Types of design processes strategy

From the first kind of viewpoint, Panchal, et al. (2006) propose five types of design process strategy, namely activity-based perspective, functional evolution perspective, the evolution of product states, the manipulation of knowledge and the decision-based perspective.

In activity-based perspective, it takes design process as a collection of many subsystem activities; therefore we can divide the design process into more detailed classification, namely sequence design, set-based design, use of surrogate models and parallel iterative design.

Sequence design type of design process represents that each end of the earlier design stage is the start of the latter design stage in the whole design process, which is the same concept with tandem process proposed by Khanna et al. (1998).

The set-based design process type means when making decisions, designers consider sets of design alternatives rather than pursing one alternative directly, which is still under the assumption of activity-based process perspective.

Use of surrogate models one emphasize the surrogate model helping make the process design decision, while parallel iterative design focus on the parallel cooperation of related design tasks of the process, and the iterative simulation activities would continued until achieving the optimality.

3.2.3.7 Types of information system development processes

Wu and Lin (2001) illustrate seven types of information system development processes, consisting of Code-and-fix Model, Stagewise Model, Waterfall Model, Incremental Model, Prototyping Model, Spiral Model and Concurrent Model. These above models are developed for software system development, thus we want to analogy the frames of these models into our collaborative deign field.

First comes the Code-and-fix model which is the earliest one proposed for software (system) development consists of following two steps: (1) coding part of the programs without thinking of its demand and usage. (2) refining the programs coded before. This kind of developing model has the problems like lack of customer demand analysis and sound planning of the whole development project.

The design principle of his type of developing model for CD field would be that each team of design collaboration has their own tasks without integrated with one another, but also doesn’t make a plan inside their own team, they just do what ever they can do at the moment, and may end up with the same result like the original Code-and-fix Model does, which is that they produce and design a lot of useless works and the results are still waited to be rework and reorganize.

The Stagewise Model represents stage by stage collaboration, using the concept of activity-based process mentioned earlier, and has the characteristic of the end of earlier stage would become the start of the later stage of the whole design process. The Waterfall Model is kind of similar to Stagewise Model in view of using the concept of activity-based process; however, Waterfall Model separates the stages of the whole collaboration process into few main stages, and clearly defines the output of each stage without iteration. Generally we have the three-stage Waterfall Model, namely analysis _ design _ execution. After that some ten-stage Waterfall Model is also proposed under the framework of three-stage Waterfall Model, concept of extended Waterfall Model is shown in Figure 3.22.

The Incremental Model can be seen as the expended model induced form the waterfall model. Unlike waterfall model, incremental model thinks demand can be decomposed into detailed increments. Since each increment can be taken as a part of the whole demand, incremental model establish these part demands a development cycle separately, and the detail development/design workflow would be taken by the way as waterfall model. The main difference between these two models is that subtasks of waterfall model have to be carried out simultaneously, while incremental model can implement the subtasks individually, which has more flexibility. The Spiral Model is also the expended model of waterfall model, involving the element of risk management and usually applied in large-scale government software development projects.

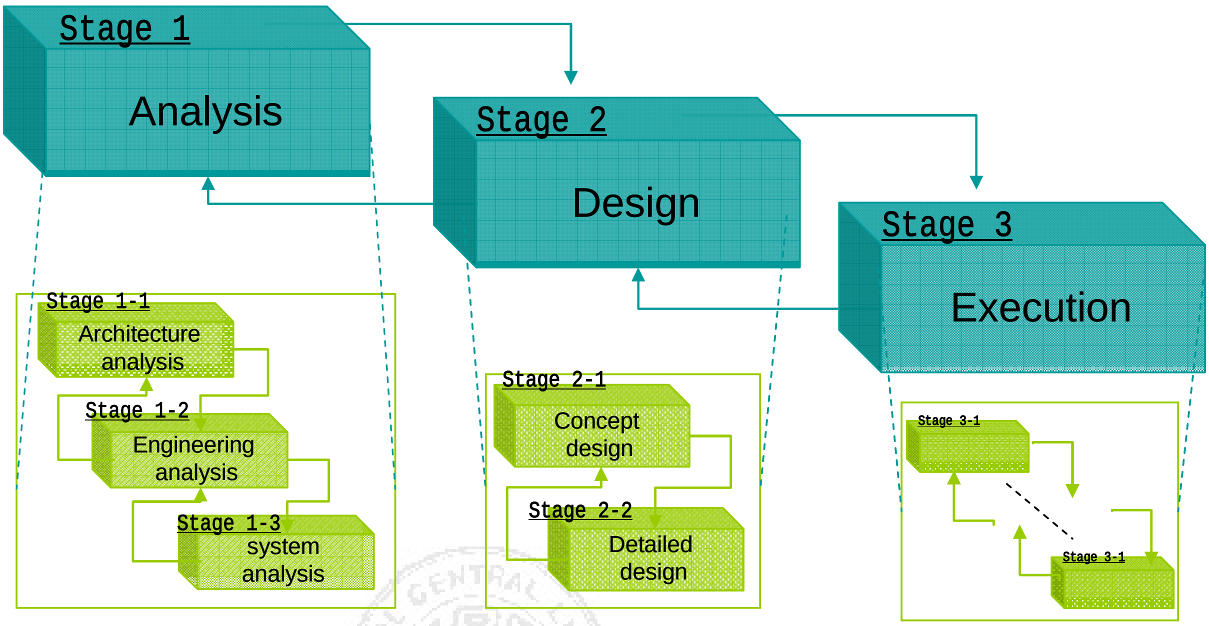


Figure 3.22 Extended Waterfall Model (Source: adapted from Wu and Lin, 2001)

The concept of Prototyping Model is more like the modular product development process. Each team can developed their own part constrained to the limited data, information or knowledge on hand, and follow the development process mentioned in the incremental and waterfall model section, collaborative works and team tasks can be carried out concurrently. This kind of model emphasized on the intense communication among cooperators and iteration of revising the output of prototypes among all teams until fulfill the customer demand. The Concurrent Model emphasizes on the concept of activity concurrency and information concurrency, which are both the bases of our discussion on CD.

One of the ideas worth to mention is that collaborative teams and generation of product design versions can be arranged with each other mutually, which means one design team can participate in many project at a time, same of the designer of the design team can join in many design team at the same time, and development or collaboration can be carried on among different generation of product design simultaneously. The interactions among cooperators will have more details in the collaborative organization section.

3.2.3.8 Types of design collaboration process in view of process dependence

According to Khanna et al., design collaboration process can be categorized into six types by its relationship of dependence between cooperators. A statement H is flow-dependent on G if the value computed by G is used by H as shown in Figure 3.23 (a) and (f). A similar type of dependence arises when a design task requires knowledge/data from another design task. A statement H is anti-dependent on G if the value used by G has its value changed by H as shown in Figure 3.23 (b).

Intuitively, H replaces the ‘old’ value of the variable by a new value and G requires the ‘old’ value to execute correctly. This type of dependence arises frequently in computer programs because it is consequence of reusing memory locations for different variables that are given the same name, thus anti-dependence arise naturally when resources are reused. A statement H is output dependent on G when G modified a variable that is also modified by H as shown in Figure 3.23 (c). This is also common in cooperation tasks.

A statement H is input-dependent on G when G reads a variable that is also read by H as shown in Figure 3.23 (d). design tasks that require the use of same design tool or an expert designer would be related similarly. Assuming that there is only one engineer who is capable of advising on wheel shaft design, then the tasks would be input-dependent because task H must wait for task G to stop needing the expert input before using input from the same expert.

The last one would be codependence or called interdependence which is the most common since there are so many activities involved in collaboration and to progress one of them may enables and also depends on progress of the other. This means that these tasks should be scheduled to run concurrently as shown in Figure 3.23 (e).

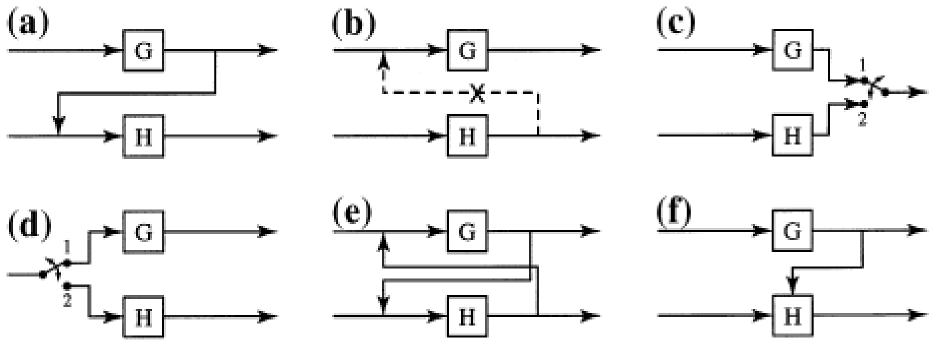


Figure 3.23 Graphical illustration of dependences: (a) (data) flow dependence; (b) anti-dependence; (c) output dependence; (d) input dependence; (e) codependence; (f) (control) flow dependence. (Source: Khanna, et al., 1998)

3.2.3.9 Types of design process in view of process integration types

Khanna et al. (1998) provide a comprehensive view of defining cooperation in two ways: (1) Integrate complementary engineering expertise and (2) Cooperate multiple competing perspectives (see Table 3.6). In the first part, three functions are defined as Distillation function, Selection function and Requisition function. In other words, to integrate complementary engineering expertise we could basically have three kinds of situations: Distillation function means to combine the complementary expertise and turn out to be a whole new outcome U; Selection function means the outcome would be either expertise A or B; Requisition function means outcome of combining two complementary expertise would need another expertise C to become U the output.

In the part (2), functions include Preference function, Multiplication function and Null function. Put it in a simple way, the cooperation styles are discussed in view of competing alternatives, and also have three kinds of integrator as a result. Preference function means either alternative A or B would turn out to be the final outcome; the outcome of the integrator would be able to be put into the mass production stages could be one of the examples of Multiplication function; Null function means neither of the compositing alternatives is capable of being chosen as the outcome.

Table 3.6 Taxonomy of cooperation in view of integrators (Source: adapted from Khanna, et al., 1998)

View	Types	Function
Integrate complementary engineering expertise	Distillation function	f (A,B)→U
	Selection function	f (A,B)→U = (A or B)
	Requisition function	f (A,B)→U = need (C)
Cooperate multiple competing perspectives	Preference function	f (A,B)→U = A
	Multiplication function	f (A,B)→U = nA
	Null function	f (A,B)→U = 0

Legend: f means integrator function, U stands for output or outcome

3.2.3.10 Types of process models in view of business structure

Types of process interactions and process models have already been extensively discussed among many researches, and we summarize four main process models in view of interactions among participants and three levels in defining modeling design process which is proposed by Panchal et al. (2006).

Based on the report written by Department of Industrial Technology, R.O.C. (DoIT, 2005), there are four cooperation frameworks proposed for clarifying the interactions between manufacturers and their brand customers, which are shown as following Figures 3.24. Four main types of product design process can be approximately categorized into OEM (Original Equipment Manufacturing), ODM (Own/Original Design Manufacturing), OBM (Own Brand Manufacturing), and CDM (Collaborative Design Model).

We can tell from Figure 3.24, OEMs have more interactions with their customers in the late development stages including Product Validation (PV), Manufacture Validation (MV) and Mass Production (Ramp/MP). From the stage of Idea/Requirement，Concept Design (CD), and Design Build (DB), to the stage System Integration (SI), business of marketing, product management, and product development are all controlled by brand firms but OEMs, which are only responsible for the poor-margin businesses including manufacturing and quality control sections in the development stages PV, MV and Ramp/MP.

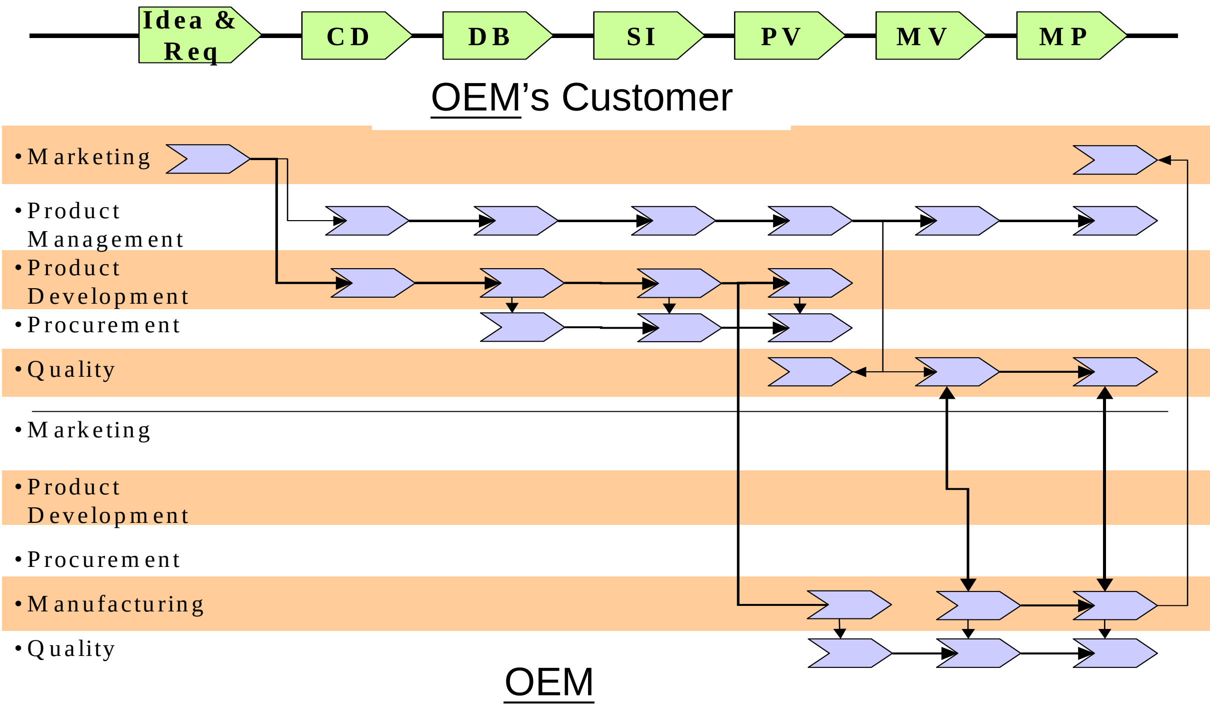


Figure 3.24 OEM cooperation framework (Source: DoIT, 2005)

Cooperation frameworks of ODM can be put into two situations considering the degree they involved in the design stages and authority of design decision-making given by their customers. (see Figure 3.25 & Figure 3.26) ODM-1 model represents manufacturers only has the product specification and not fully-authorized by brand firms in product design sections, while ODM-2 one represents brand firms give their ODM manufacturers total authority in not only designing product but also keeping the design up with technological standard and codes, which means this kind of manufacturers have a better capability and technology than the former ones. The former ones have more interactions with their brand customer in the stages of Design Build (DB) and System Integration (SI) than the OEMs, and ODM-2 even starts from the Idea/Requirement stage of collaboration.

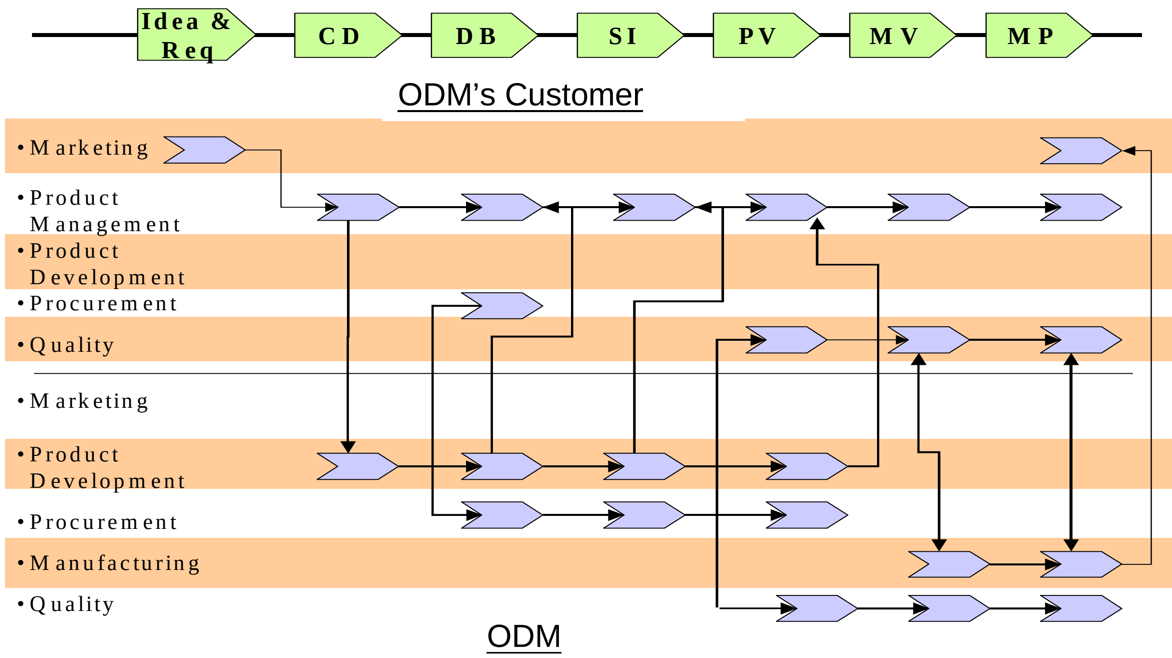


Figure 3.25 ODM-1 cooperation framework (Source: DoIT, 2005)

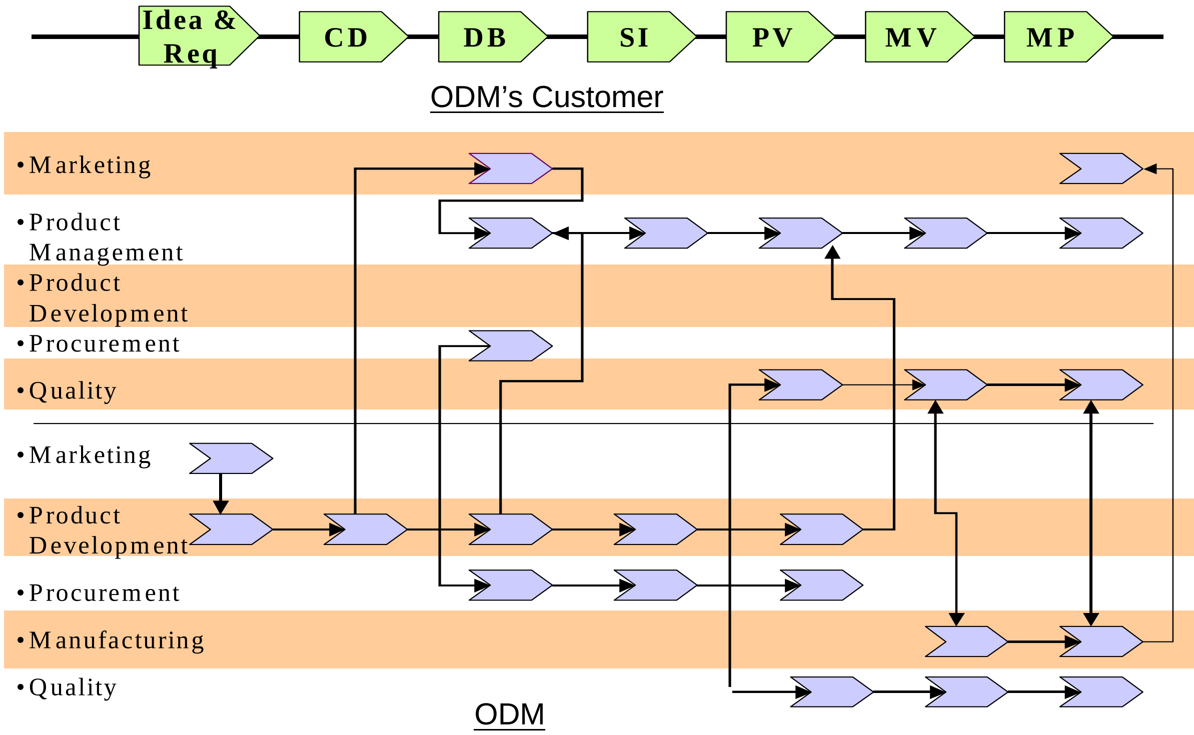


Figure 3.26 ODM-2 cooperation framework (Source: DoIT, 2005)

The interactions between OBM and its customer would focus mainly on the Idea/Requirement stage and have more cooperation on marketing and product development than ODMs and OEMs as shown in Figure 3.27.

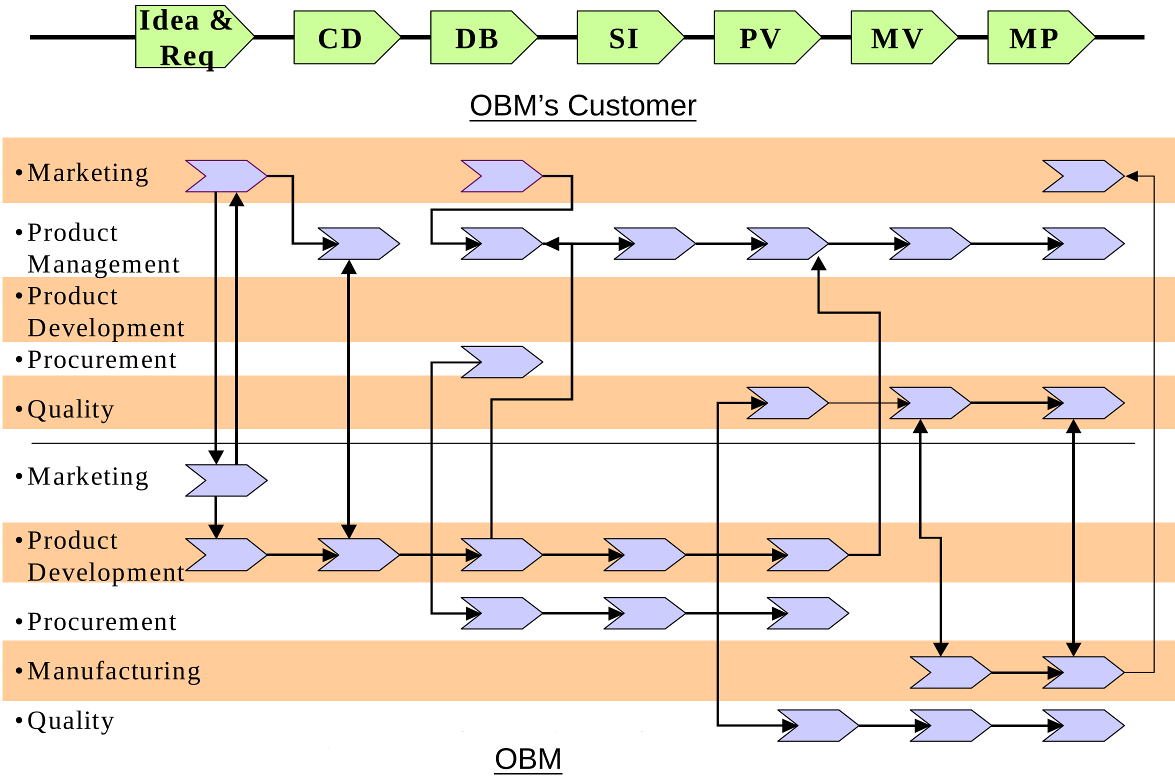


Figure 3.27 OBM cooperation framework (Source: DoIT, 2005)

Even OEM, ODM and OBM can have collaboration with their brand customers, what we really focus on would be CDM in this paper. Since OEM, ODM and OBM has their collaboration status more on the supply chain dependence of each other, CDM give us more space and flexibility on this collaboration design topic since the cooperation relationship can be extended to virtual enterprises and extended enterprise without the constraint of supply chain connections. The cooperation framework of CDM is as shown in following Figure 3.28.

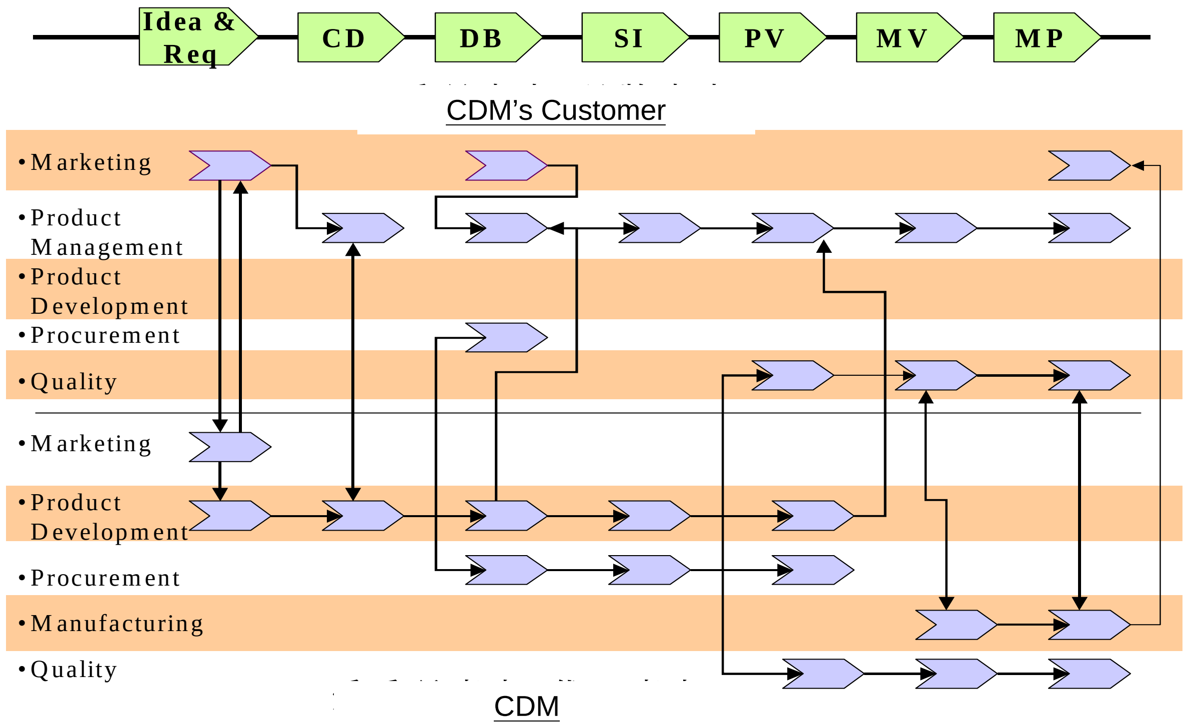


Figure 3.28 CDM cooperation framework (Source: DoIT, 2005)

3.2.3.11 CD process models

Frameworks or models regarding CD are much pretty, but not with the case of CD process. Proposed CD process models have not been investigated thoroughly so far in our literature review. Britton et al. (2000) present us their proposed CD process model as shown in Figure 3.29, and do clarify the interactions and collaboration relationship of involving entities to certain extents.

However, this proposed model would be more like a given scenario concerning CD process and components rather process model itself for the reason of lacking in elaborating on core issues like design process details related to design collaboration, process types classification, process adoption guidelines, and so on.

In our view, topics regarding CD process including reference models proposing should be taken accordingly in view of different perspectives based on the result of our literature review. We summarize four perspectives dealing with these issues in the next session.

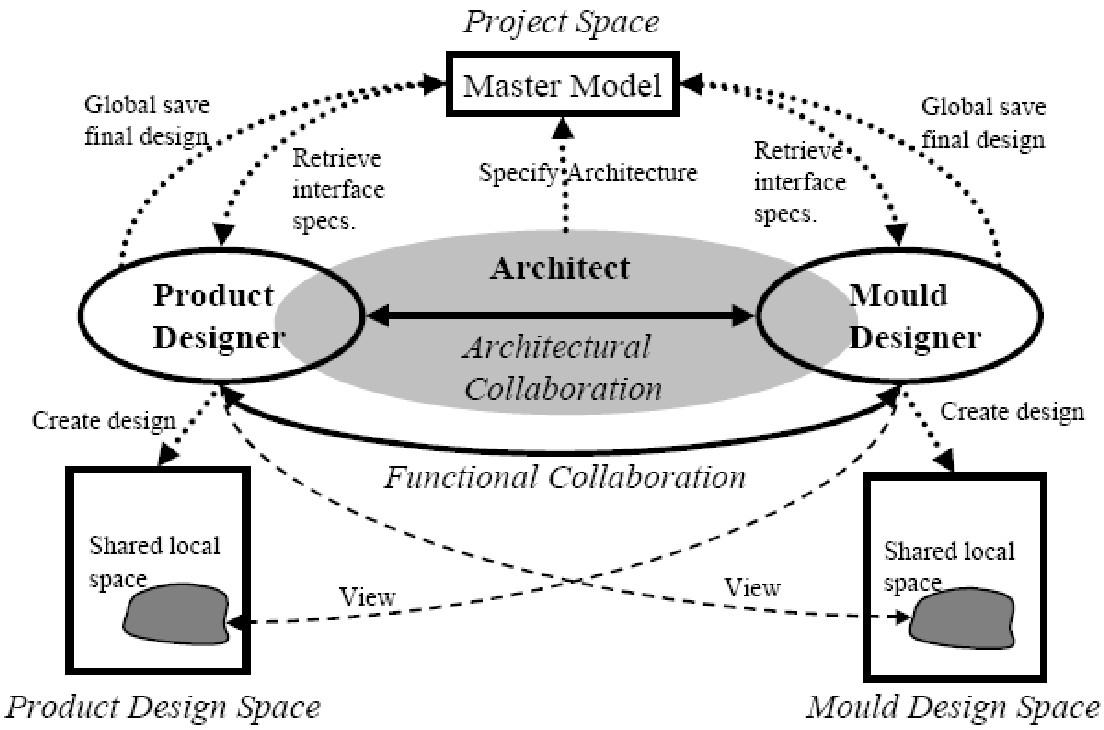


Figure 3.29 Proposed collaborative design process (Source: Britton et al., 2000)

3.2.3.12 Perspectives for design process

After reviewing so many researches regarding process types of design, product development and collaboration, we try to give an overview aspect of design process strategy. Under the design strategy domain, we barely conclude four perspectives in this topic (See Figure 3.30).

1. <u>Activity-based perspective</u>

In the activity-based perspective, process of collaboration (whether design collaboration or engineering collaboration) has the attributes of capably decomposed into sub-process, which is used for following process type classification, which are defined by Panchal, et al. (2006).

In the sequence design process, examples are as traditional product development process, Stagewise model proposed by Wu and Lin (Wu and Lin, 2005), and so on. The components of this kind of design process are activity-based; order of each activity is executed one by one, following the sequence discipline.

Examples of set-based design are like waterfall model (Wu and Lin, 2005), incremental model (Wu and Lin, 2005), product developed using Darwinian selection method (Tuulenmäki, 2006), and so on. The surrogate-using activity-based model take the design process as decomposable, thus could be discussed in the unit of activities. Example of surrogate-using activity-based model is like spiral model (Wu and Lin, 2005), considering the risk management issue into the design process.

Parallel iterative design can be viewed as the combination of concurrent model and prototyping model mentioned by Wu and Lin (2005). Parallel iterative design is the process strategy not only puts activity concurrency and information concurrency into consideration, but also emphasizes the refinement of design process model. The iteration concept of this model also has a similarity to product adopting Product morphing method (product design is iterated repeatedly to gain the competence in the fierce market) (Tuulenmäki, 2006).

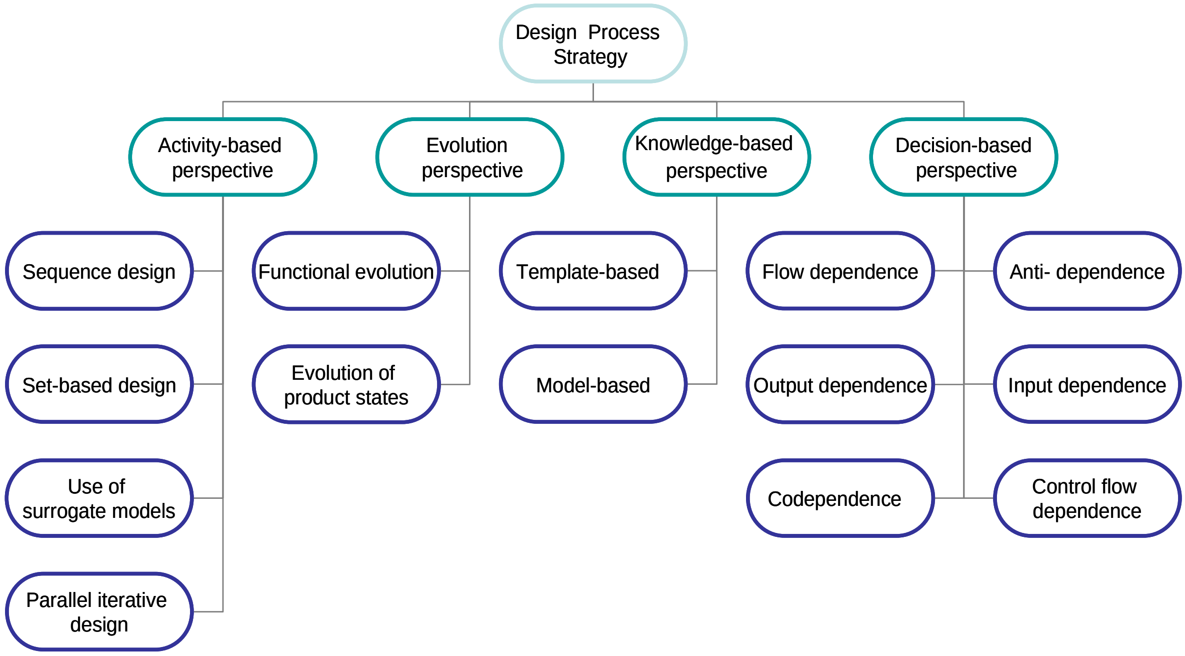


Figure 3.30 Taxonomy of design process strategy (Source: Original)

2. Evolution perspective

In the evolution perspective, object to be evolved can be taken in two way function and product status (evolution of product design activity is in activity-based perspective). In this viewpoint, process types are categorized by the evolution object it takes. Some product design process is modeled in view of product function evolution such as product developed adopting the variant strategy (Tuulenmäki, 2006) and the prototyping model (Wu and Lin, 2005), while others may be in view of product state evolution.

Example of product function evolution is like the product generation displacement of PS series product of Sony for better product function, the product developing process in Figure, and so on. Product adopting variant method has a transformation of product function from product variant 1-1 to product variant 1-2, and the product function is still evolving under this stage to achieve a better performance.

On the other hand, in view of the product status changes, process could be identified by the changes of its states. The meaning of state here represents the situation of versions or status of the product design/ collaboration. The concept of parallel iterative model and prototyping model could show kind of the content of what we are talking about.

Take product development as an instance. Each state here represents the version of final product but alternatives because version here has the implication of product improvement and refinement not just the difference of appearance, appliances and functions of the product. Hence, in this viewpoint, product development process may compose of many product versions generation by generation along project time.

3. Knowledge-based perspective

In knowledge-based perspective, design process can be modeled into two types as template-based type and model-based type. The concept of template-based is come from design process mention by Panchal (2006) as shown in Figure 3.30. Template can be divided into two types complete and partial. Complete templates contain all the information required for carrying out a transformation and can be executed, while partial templates do not have sufficient information for executing a transformation (transformation willed be clearly defined in the information section 3.1.5.).

With basic definition as above, the template-based design process we meant here represents process can defined by the template it required and used. Take the instances shown in Figure 3.31 provided by Panchal, the process of this task can be taken down as T1 _ T2, T3, which means process should go through complete template T1 first and then to partial template T2 and T3 separately according to the knowledge it required from state A (represents geometry and loading conditions) to state B (represents their combination and resultant behavior).

The combinations of templates in the design process can be arranged by information environment conditions and its knowledge requirements (also see Workflow Data Patterns). With different situations of knowledge repository, design complexity and transparency of information, configurations of template combination are also different among interactions of organizations.

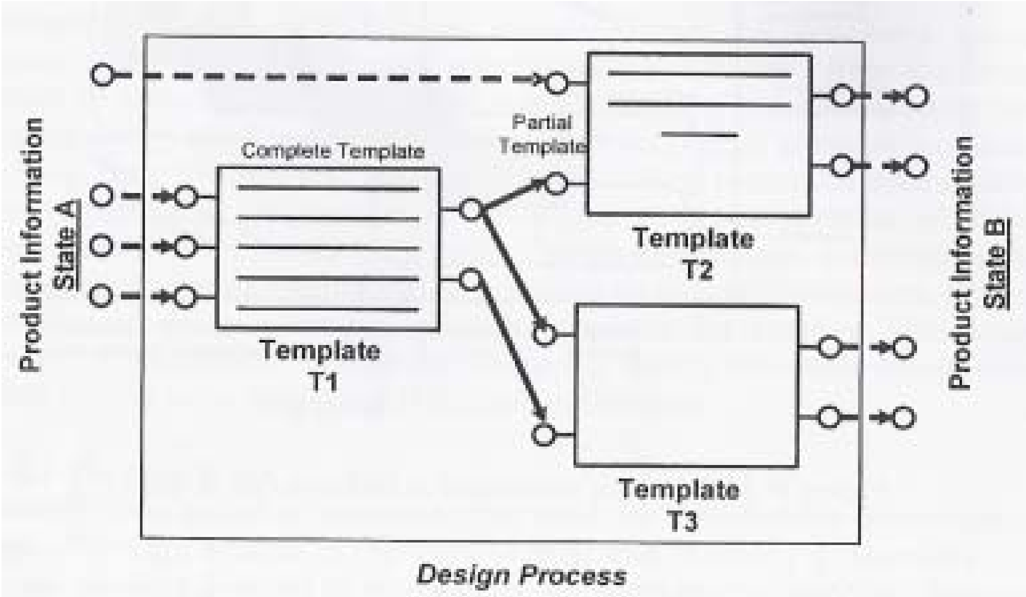


Figure 3.31 Modeling design processes using process template (Source: Panchal, 2006)

In model-based aspect of the knowledge-based perspective, design process selection/modeling mechanism can be chosen from existing process knowledge models in view of product lifecycle (PLC). General definition of PLC would start from the stage of idea/requirements to mass production (stage would extend to recycling when talking about green product design). In this paper we focus on CD, hence the domain of PLC would be constrained in product design.

Knowledge model of product design can be sorted into concept definition model, concept formation model, concept validation model, product formation model, product architecture model, prototyping model. Models proposed here mainly discriminate the process into sub-process not by the activities or functions but by the knowledge applied during the design process. The six models can mainly be separated into two phase: concept developing phase and product design phase. (See Figure 3.32)

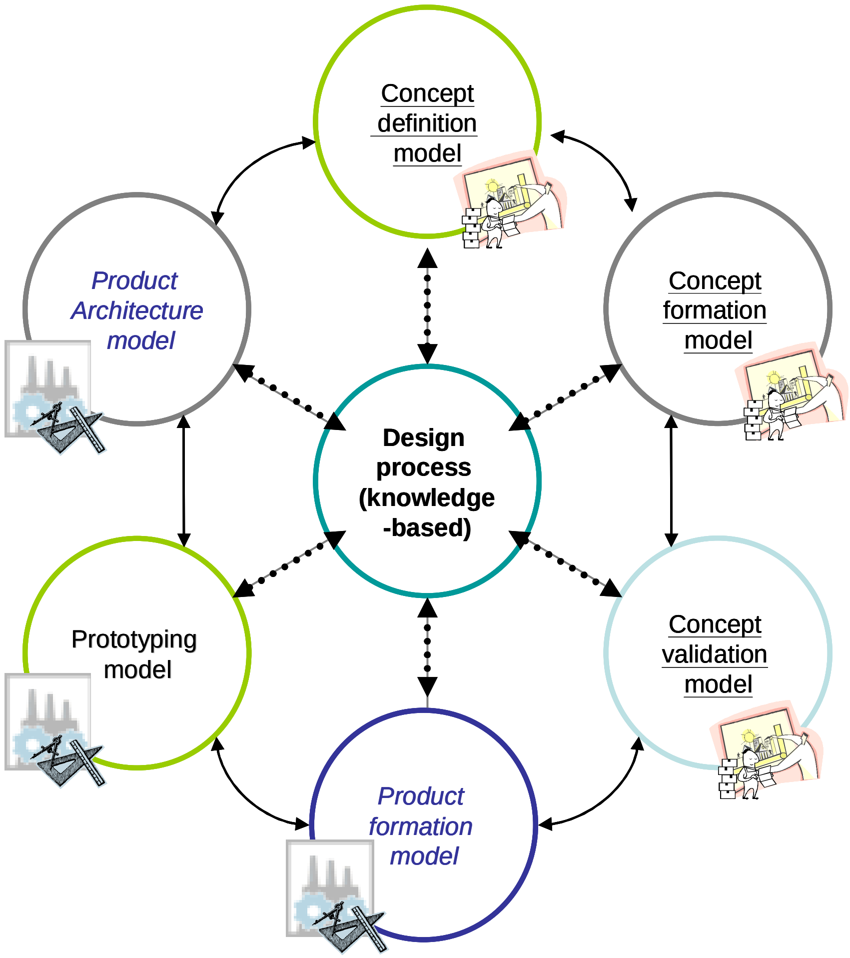


Figure 3.32 Design process models (Source: original)

Concept developing phase including model possessing knowledge tools and framework for concept definition such as the goal of this project, elements required to put inside, and so on. Concept formation model focuses on concept realization, thus under this model, environment definition, technology required, demand analysis for customer etc. should be included.

Concept validation model shed more lights on technological analysis and radical concerns such as financial and market elements. Passing through all these three parts, next would be product formation model, which is to help clarifying the elements, procedures needed to become a product and including main design tasks of the whole project such as properties definition, detailed design, and so on.

Product architecture model for physical product structure confirmation and make sure there's no conflicts between each function of the product, after all of the above procedures, the design process would go to prototyping phase to prepare for manufacturing and mass production in the later stages of product development.

4. Decision-based perspective

In decision-based perspective, design processes are defined by the relationship of dependences between design tasks. We take the six types of design process proposed by Khanna, et al. (1998), which are introduced in earlier section.

3.2.4 Organization aspect of collaborative design

In order to improve and to facilitate the effectiveness and efficiency of CD process, we define the types of organization and kinds of perspectives of organization proposed by other researches. A team is a group of two or more people who interact and influence each other, are mutually accountable for achieving common objectives, and perceive them selves as a social entity within an organization (Mcshane and Travaglione, 2003). A structured organization can facilitate design communication and consequently contribute to the success of the design project (Chiu, 2002). With different kinds of CD cooperation, organizational changes are necessary to be adopted for better fitness of design process rearrangement and coordination between entities of collaboration.

3.2.4.1 Perspectives

Bullinger et al. (2007) believe organization in RPD (Rapid Production Development) play an important role to make sure the accuracy of product development, and they make a comparison between SE (Simultaneous Engineering) and RPD, and find out that SE Figures a more or less formalized frame with milestone whereas RPD requires a reactive project-management methodology, which give us the support of the necessity of organizational changes within collaboration.

Gasser (1994) takes society as the concept of collection of interactions, therefore under the process of collaboration, the definition of agents (e.g. work groups, organizations) boundaries/ structure formation, language they used and the way agents negotiate with each other are all included in the issues of collaborative organization, so he thinks organization is a network-level coordination mechanism, which could be taken as a perspective in dealing with topics of collaboration and information exchanges.

Chiu (2002) provides a basic understanding of the role of organization in design collaboration and how it affects design communication and collaboration by empirical case studies and design experiments. The results of case studies in architectural practice and design studios and a process model of design collaboration are presented.

Hayes et al. (1988) and Ulrich and Eppinger (2008) illustrate the importance of forming product development organizations, and do provide us kind of view on the classification of organization structure. CD is usually implemented in the form of project, thus we would focus on project organization domain applied to CD coping with the organization types of CD;

3.2.4.2 Types of organization for collaboration (in view of PM)

To trigger the collaboration, it is almost impossible to accomplish the whole task by a single individual; therefore we need a team (such as project team) or an organization which is the collection of individuals to work together though to make this happen. According to Hayes et al. (1988), no matter how complex or how hard the product design process is, it is the members of the organization who achieve the goal of the design project mission that really matters, therefore to define or refine the collaboration organization within the whole planning works and execution process of collaboration does show its importance. This is the reason why we add the organization section into our discussion. In the following section, we will first introduce the composition of a project team defined by Ulrich and Eppinger, and then we will find out more about the types of product development organization, and this kind of classification could also be used on design collaboration.

Ulrich and Eppinger (2008) divide the composites of project teams into two main groups, one is so called core team, which consisting of main member of the whole design collaboration project such as electronics designer, industrial designer, manufacturing engineer, mechanical designer and the team leader, while the other one is called extended team, which take the responsibility of supporting the whole collaborative operation and activities to move forward smoothly, members of the derivative team may include such as the department of finance, intellectual property, and so on.

In order to work together effectively, the core team usually remains small enough to meet in the conference room, while the extended team may consist of dozens, hundreds, even thousands of other members. The composition of a team for a development of an electromechanical product of modest complexity is shown in Figure 3.33.

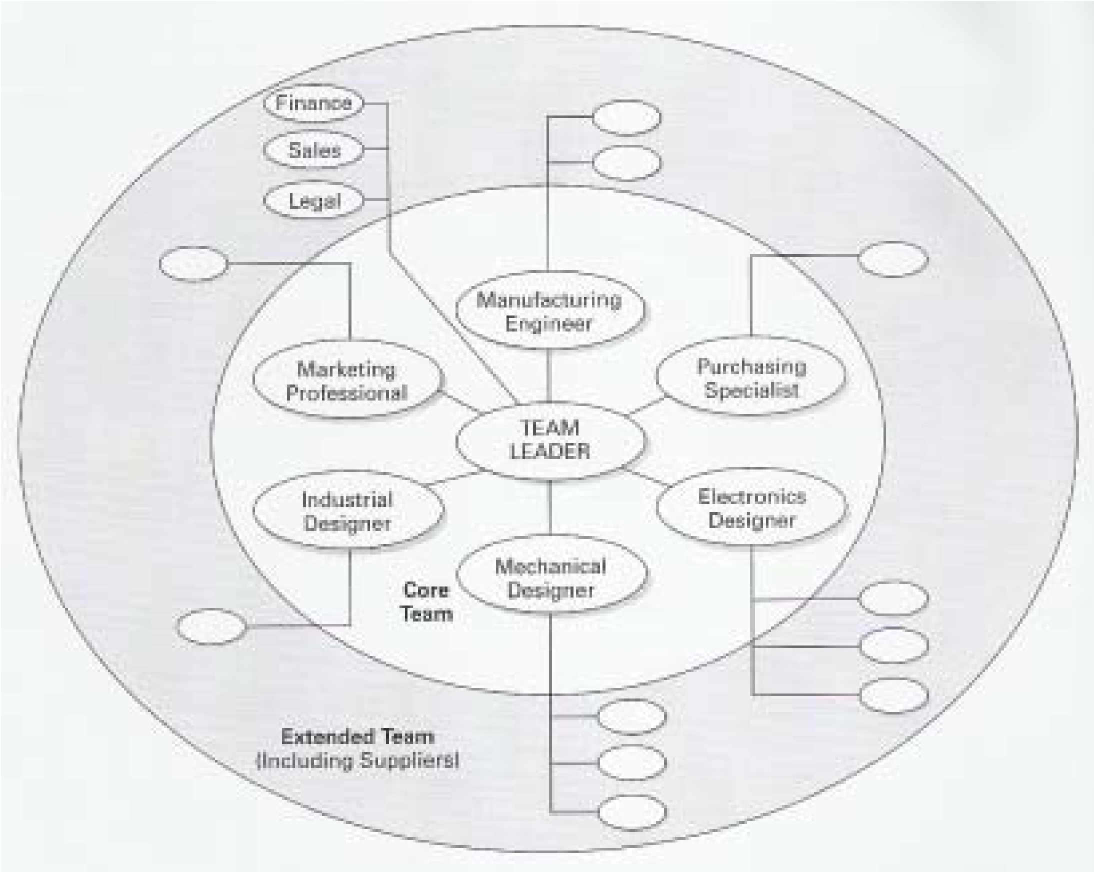


Figure 3.33 the composition of a team for a development of an electromechanical product of modest complexity (Source: Ulrich and Eppinger, 2008)

Two main project team types discussed above are defined from the view of the importance or how deep the entities of the organization are involved into the cooperation of product design and development. Hence we want to learn more about the types of the project team, or we may expend the concept into the types of organization. Ulrich and Eppinger (2008) adapt the concept proposed by Hayes et al. (1988) about the product development organizations, and summarize four main organization types by two main different ways, which are according to their function and according to their projects they are work on.

Four types of product development organization (proposed by Hayes et al. and adapted by Ulrich and Eppinger) are shown in Figure 3.34. Hayes et al. introduce the main four organizational types of development projects, namely functional organization in panel (A), lightweight project manager in panel (B), heavyweight project manager in panel (C) and tiger team organization in panel (D), which is the same concept of project organization mentioned by Ulrich and Eppinger.

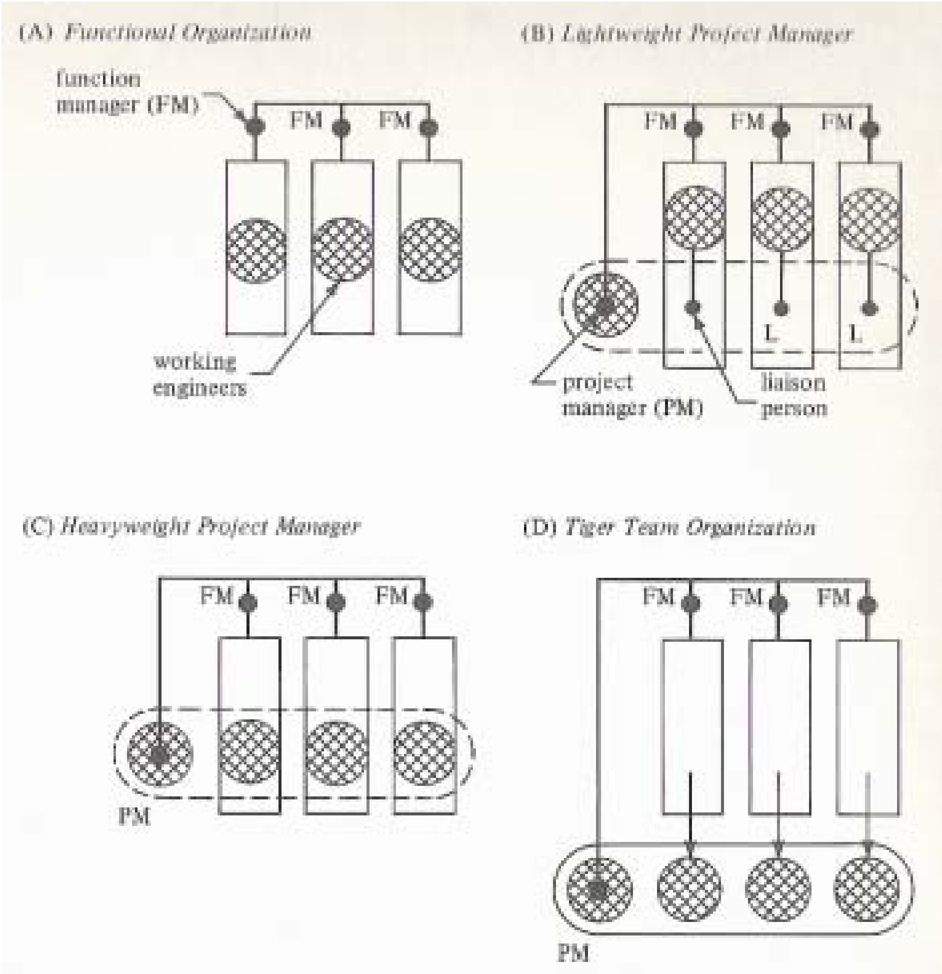


Figure 3.34 Organizational types of development projects (Source: Hayes, 1988)

According to Ulrich and Eppinger, two classic organizational structures arise from aligning the organizational links according to function or according to project. In functional organizations, the organizational links are primarily among those who perform similar functions. In project organizations, the organizational links are primarily among those who work on the same projects.

Type between these two is the matrix organization, conceived as a hybrid of functional and project organizations. In the matrix organizations, teams can be defined as lightweight project matrix organization or heavyweight project matrix organization as shown in Figure 3.35. The former contains weaker project links and relatively stronger functional links, while the latter contains stronger project links.

The heavyweight project manager has complete budget authority and makes most of the major resource allocation decisions. A heavyweight project team in various industries may be called an integrated project team, a design-build team, or simply a product development team, each of these term emphasize the cross-functional nature of these teams.

Different organizational structures have their own strengths and weaknesses, there is no the best type of project team organizational structure that one enterprise can use it forever, therefore we are going to take a closer look at the characteristics of these organizational structures. Functional organization has the advantage of fostering development of deep specialization and expertise, but has difficulty coordinating among different functional groups, thus how to integrate different functions to achieve a common goal would be a challenge to this kind of structure.

Project organization has the advantage of pursuing the optimality of resource allocation within the project team and quick evaluation of technical and market trade-off, on the same time having the weakness of maintaining cutting-edge functional capabilities, and this is also the reason why maintaining functional expertise over time and sharing technical learning from on project to another would be a major issue for it.

Project organization is suitable for firms competing in extremely dynamic markets. Among the matrix organization, lightweight and heavyweight have the strengths of both functional organization and project organization with the only shortcoming of increasing personnel cost, the difference would be slightly on the power of resource allocation and performance evaluation assigned to the project manager.

Lightweight project organization could maintain more development of specialization and expertise, while heavyweight project organization is more agile in coordinating functional integration. How to balance functions and projects priority and how evaluate the performance of both organizations at the same time would also be a challenge.

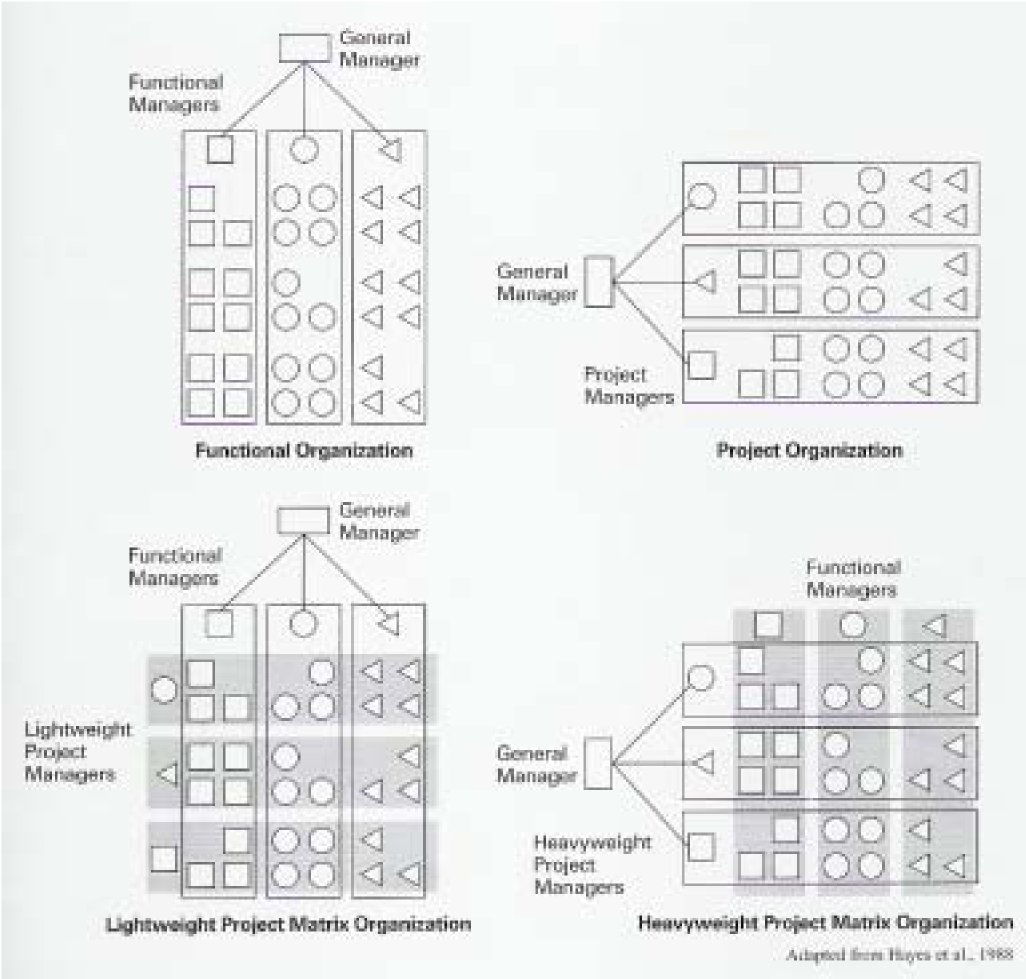


Figure 3.35 Organizational types of development projects (Source: Ulrich and Eppinger, 2008)

3.2.4.3 Team organization for CD

Chiu (2002) collects four empirical cases of design collaboration as shown in Figure 3.36. The larger the scale of project, the more organization becomes hierarchical. It is necessary and useful to break a large group into smaller groups for facilitating design communication. Generally, the project managers of each architectural firm control the design information flow, coordinate the design tasks, and distribute information to individuals.

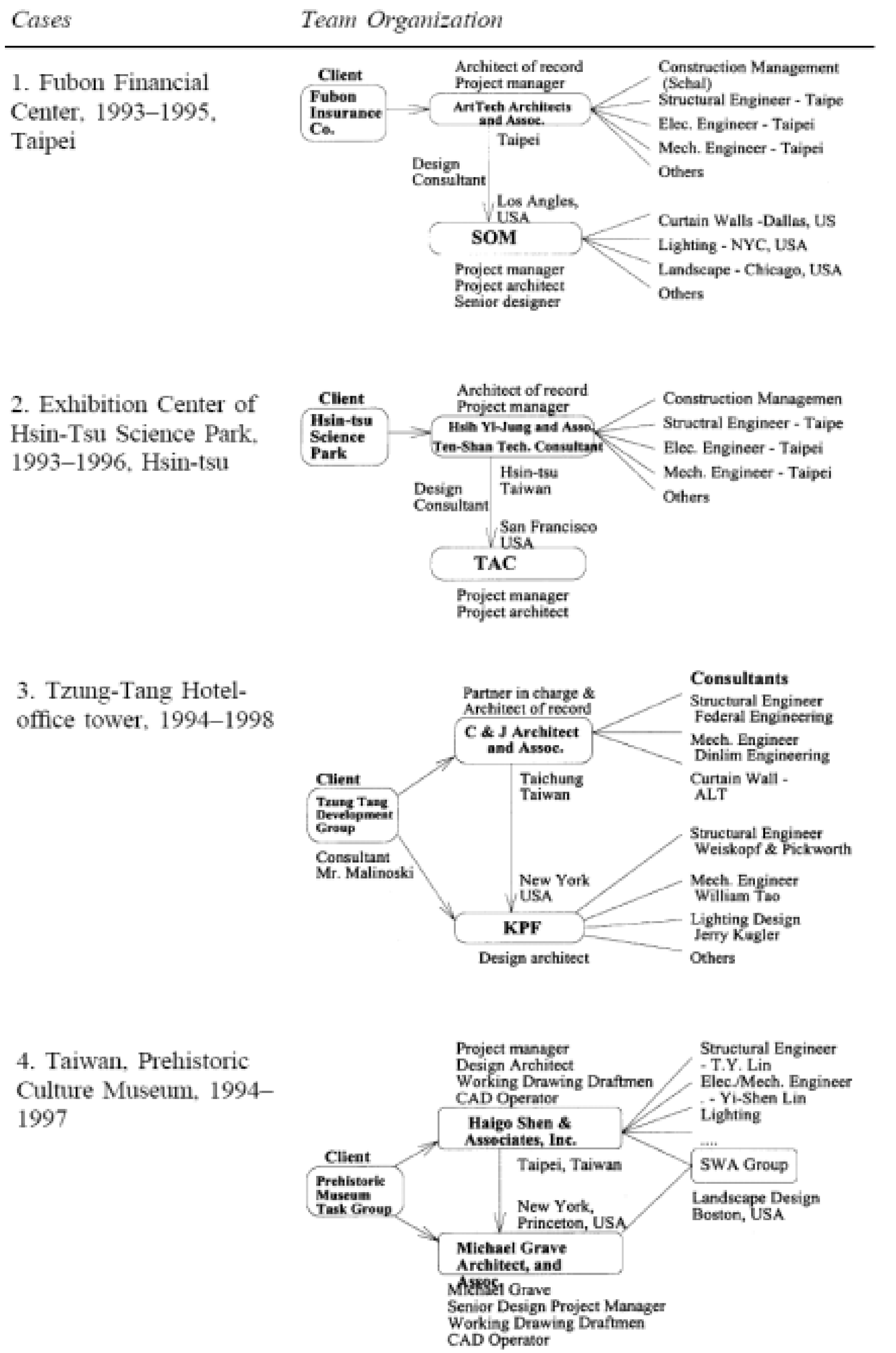


Figure 3.36 an organizational view of design communication in design collaboration (Source: Chiu, 2002)

Two types of team organization are typically found in practice, i.e. mesh and star, as shown in Figure 3.37. In either type, the project manager could coordinate and control the information flow. While the scale and type of project may determine the level of communication, both types are often used simultaneously among groups or within groups.

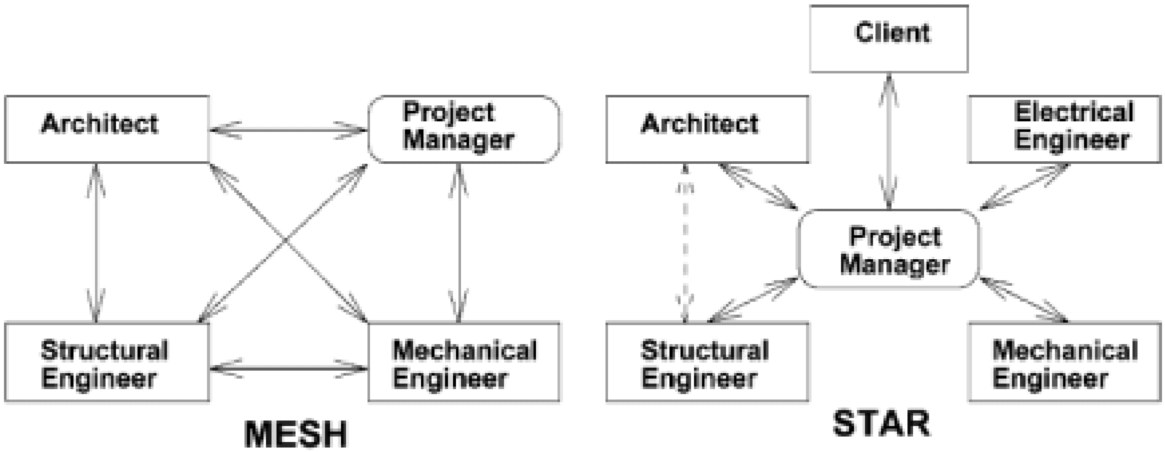


Figure 3.37 the design organization in the mesh and star network (Source: Chiu, 2002)

3.2.4.4 Taxonomy of CD organization

Combining theories proposed by Hayes (1988), Ulrich and Eppinger (2008) and Chiu (2002), taxonomy of CD organization is presented in the following Figure. These three organization categories can explain considerate CD situations and provide a reference.

Extended from classical project management organization classification proposed by Ulrich and Eppinger (2008), we bring up the following taxonomy of organization types, namely project, mediatory, and functional organization. (See Figure 3.34 and Figure 3.35)

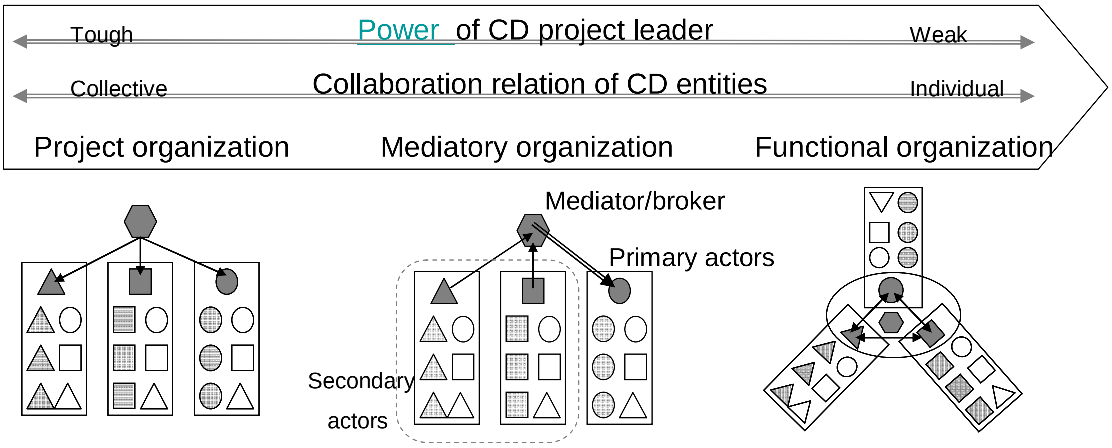


Figure 3.38 CD organization spectrum (Source: original)

In Figure 3.38, it is easily being realized of the tightness both in virtual and physical connections within entities in the spectrum from right to left. Details of the three CD organizations will be elaborated in Table 3.7. The power of CD leader we meant here can be clearly defined by the following six power bases proposed by Forsyth (2006) shown in Figure 3.39.

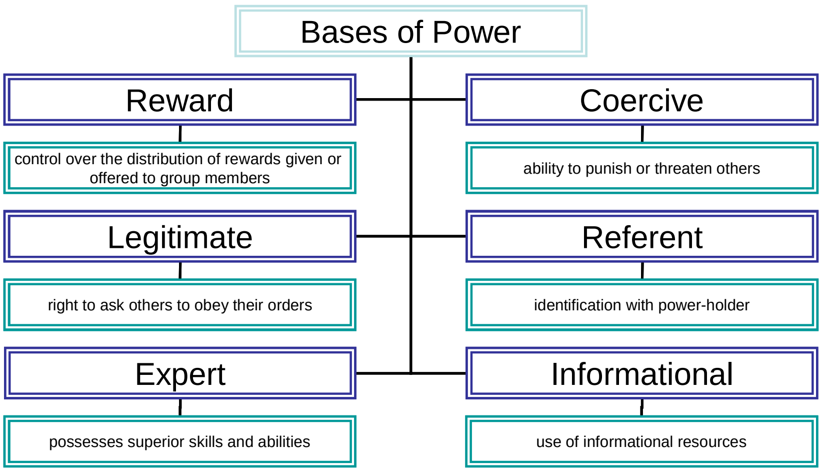


Figure 3.39 Bases of Power (Source: Forsyth, 2006)

The hierarchical power can comes from reward, coercive, legitimate, referent, expert, and informational aspect. Reward represents the authority to evaluate or assess over the whole entities, and concerning the effects of performance audit results, the more one possess power of rewarding, the more power one own. Coercive represents the ability to punish or threaten other, which can be taken as the adverse element to reward. The latent impulsion forming coercive may be possessing good up-/down-stream relationship, power of ordering, etc. The more one can coercive others, the more power one own. Legitimate represents right to ask others to obey their orders, and the more legitimate one is, the more powerful one shall be. Referent represents identification with power-holder, and this comes from the background and the status one occupied on the market. The more referent one is, the more powerful one shall be. Expert represents superior skills and abilities one may possessed, and the more expert one is, the more powerful one shall be. The last one would be Informational, which we meant scarce resources here, and the more Informational one is, the more powerful one shall be.

Table 3.7 organization type taxonomy (Source: original)

Characteristics of various types/ CD organization types	Candidate of CD leader	Hierarchical power	Final decision maker	Power of entities	Collaborative relation of composing entities	Interfaces of the entities
Project organization	From one of the entities	Tough/ demanding/ powerful	CD leader	Non-independent, Decision making capability possessing	Collective-oriented	Demanding protocol etc.
Mediatory organization	Primary actor	Clear hierarchical relation (primary/ secondary)	CD leader (Primary actor)	Independent entities but cross-decision dependent	Partnership/ supporting relationship	Mediator/ broker can be software agent or human
Functional organization	From one of the entities or the third party above all entities	Weak/ Harmony hierarchical relations	All of the entities	Independent, Entities are responsible to individual belonging enterprises	Individual-oriented	Focusing on interfaces integration

Project organization represents those kinds of CD in the manner of being composed of more collective entities with a tough or centralized leader. The controlling role, which we call it CD leader here, comes from one the CD entities and is usually the primary actors of the activities or being the most demanding one. For its centralized hierarchical organizational structure, CD leader from the entities has the right to make the final decision. The target CD case of this kind of CD is prone to be architectural engineering design collaboration such as co-design the air-conditioning system and engine system of a car concurrently. For this reason, CD whose organizational structure falls on to project one would have collective-oriented relationship among entities. In addition, the interface applied for communicating between entities and the leader is more likely to be demanding protocols but bilateral-equal ones.

Functional organization represents those kinds of CD in the manner of being composed of more individual entities with a weaker hierarchical power leadership. The groups of this kind of organizational structure are more likely to focus its product types on the issues of interfaces or structures integration rather concern interactions of interrelating architectural collaboration or tasks as of project ones. CD of this case is usually supported by IT platform, LAN, or

Internet to accomplish the design collaboration, and relationship of entities is less hierarchical and more individual on the co-design table.

Cases of the two types of extreme organizational structures are much more less take place, and there emerges the third one, mediatory organization. This kind of CD usually initiated by primary actors and the mediator role can be played by primary itself or the third-party broker (could be software agent, individual, group, organization, or corporate). Its hierarchical power is clear, primary release the need for CD, and secondary can propose solutions to primary through mediators/brokers, keep in mind, primary still have the decisive final decision making authority.

3.2.5 Information aspect of collaborative design

In this section, we are dealing with the information aspect of CD. Topics of information of CD can be extended to many issues such as information sharing mechanism among cooperators, collaborative design platforms configurations, information exchanges of CD, and so on. To have an integral view of information in CD activities, we will deliver the following paragraphs by the sequence: (1) collaboration environment, (2) static aspect of CD about information interactions, (3) dynamic aspect of communication within interactions of CD.

First of all, we will provide a summary of collaborative environments for the sharing of design information, data and knowledge among distributed design teams. Having the elementary knowledge of the environments dealing with CD information, we will take a further step to introduce the contents of interactions during CD, which is the static side of the information aspect of CD. In this part, we would first give a sketch on reasons and instances of information exchanges and communication, and then we will introduce the contents of the subjects (we call it actor) doing all the interactions.

The *artifact* part would show us the definition and classification of data, transformation, information, and knowledge. The actor part which is set to define the participants of CD, and introduce the taxonomy of actor has already been introduced earlier in section 3.2.1. On the other hand, the dynamic side of information in CD is also a significant issue, which would focus on the process and framework of exchanging information, and we will outline the necessary elements to configure a framework of CD.

3.2.5.1 CD environments

In this section, we summarize environments of CD base on literatures reviewed (Hao Q. et al., 2006; Panchal, et al., 2006) and find out that CD environments can be divided into three categories, namely web-based CD, agent-based CD, and the combination of the two. Web-based CD is indicated to the environment of design system using the client-server architecture such as web-based framework for distributed cooperative manufacturing system proposed by Mahesh, et al. (2006).

In the environment of web-based CD, we need models of domain specific integration tools and models of general distributed computing applications to fit the configuration requirements.

Domain specific integration tools do the function of providing a communication interfaces within two different domain specific actors in CD such as the CAD-CAE integration tools and collaboration architecture to allow distributed designers to work on the same CAD. On the other hand, General distributed computing applications are like e-web portal to illustrate how web-based standards and distributed object technologies can be integrated to provide controlled access to any type of information and resource within the extended enterprise.

In *agent-based CD*, we define the environment whose system has properties of loosely coupled network of problem solvers that works together to solve problems which are beyond their individual capability as agent-based CD. Most agent-based systems have used knowledge-based standards for achieving interoperability between agents. Knowledge based standards involving defining common ontology and/or definitions agents agree upon (Panchal, et al. 2006). The Agent-based Web service integration framework proposed by Wang et al. (2006) can be regarded as an example of agent-based CD system as shown in Figure 3.40.

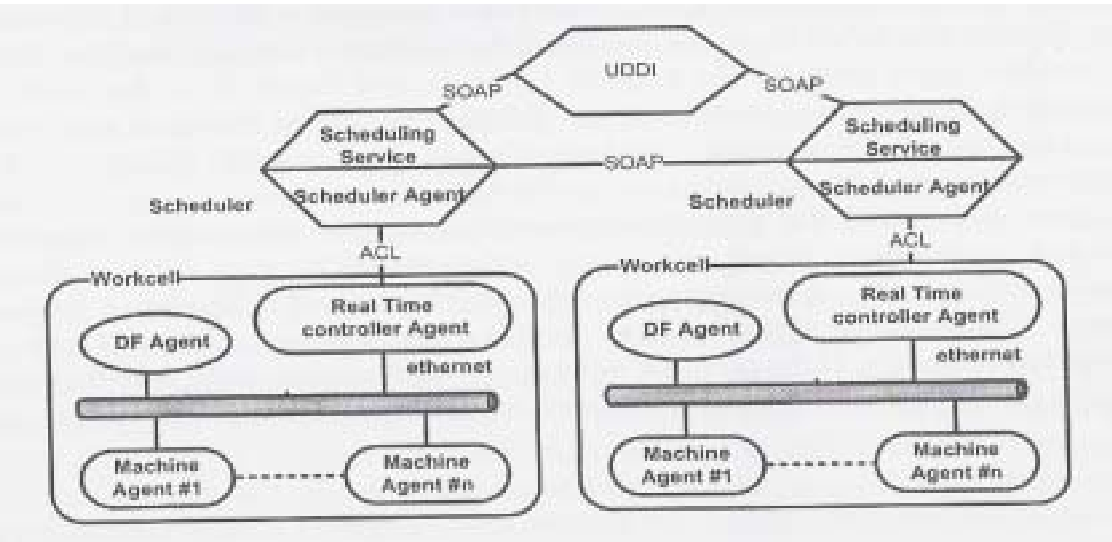


Figure 3.40 Agent-based Web service integration (Source: Wang et al., 2006)

Combining these two types of CD circumstances, we will have the third one called agent-based Web which has the client-server architecture but the approach to demonstrate the collaboration is agent-based. The Agent-based Web service integration framework proposed by Shen, et al. (2007) will give us a example of this case.

3.2.5.2 Contents of communication

Brazier, et al., (1995) provide us a big picture about why information sharing is so important and show us the reasons for communicating with one another during CD process, and this would let us catch on the background of communication itself before entering to the details of specific contents.

1. Reasons for communication

As Liu and Leu (2006) point out, conflicts are unavoidable during collaboration among multiple stakeholders, who have different objectives, requirements, and properties. The extent of communication complex would follow exponential distribution as shown in Figure 3.41, which accentuates the importance of communication dealing with cross-functional cooperation. Hence, communication is absolutely the further basis for cooperation to deal with the conflicts occurring around the CD process. It guarantees the continuous exchange of data, information, and knowledge. Particularly dynamic processes, like the development of innovative products, demand willingness to communicate from the developing partners, especially when the partners have not worked together before (Bullinger, et al., 2007).

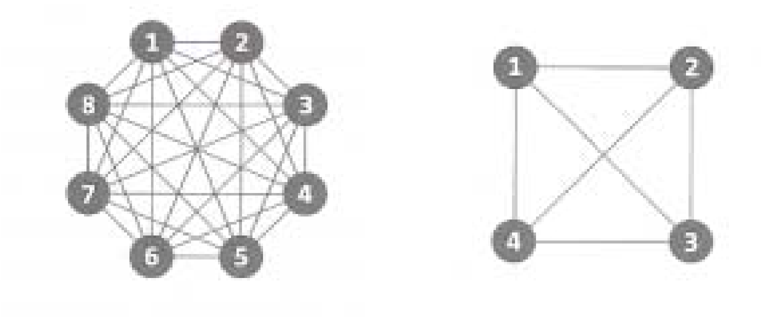


Figure 3.41 communication in teams of eight and four members (Source: Keinoneon, 2006)

Take the example of information interchanges and integrations within domain of CAD tools. As scenario proposed by Roller, et al. (2002) shown in Figure 3.42, a dependency between interdisciplinary design models is a relationship between a MCAD (Mechanical CAD) and an ECAD (Electrical

CAD) component. Even within the same scope of enterprises and under the same interfaces of visualization tools like CAD, dealing with different function modules such as MCAS and ECAD would cause the interoperability problems or resource sharing constraints during collaboration.

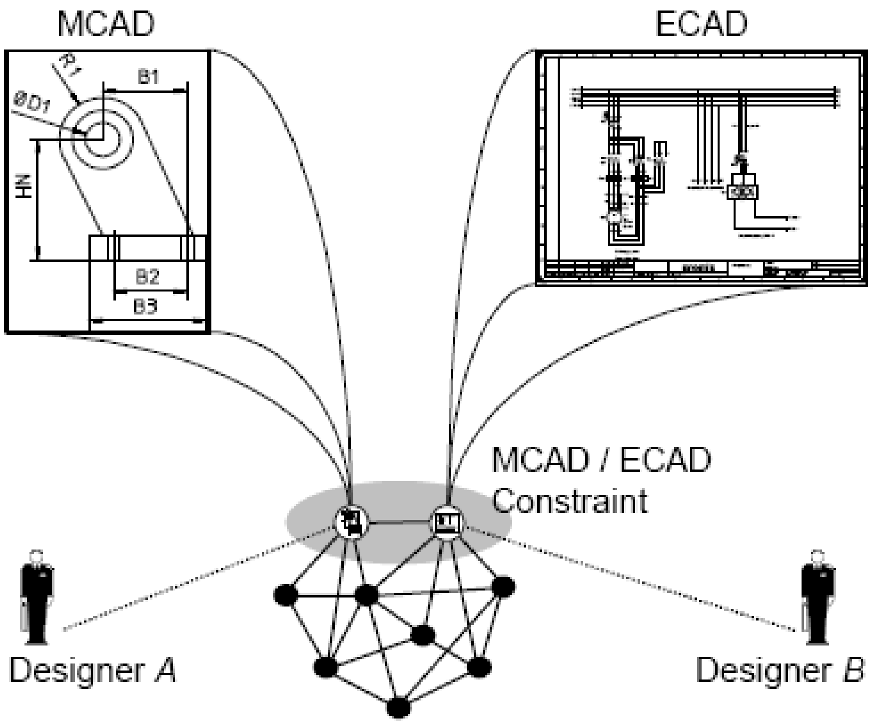


Figure 3.42 Example of an interdisciplinary work between a MCAD and an ECAD model
(Source: Roller, et al., 2002)

Based on Liu and Leu (2006), we summarize that conflicts of CD can be basically divided into three types of reasons: conflicts regarding tasks itself, interactions of tasks or activities, and conflicts of system level with different disciplines. The first one is conflicts during management of information content, which includes two main parts, one indicates the conflicts like design tasks with different objectives, properties, requirements, or even sharing same resources such as machines for making components, material for manufacturing, and so on. We can take the framework for design argumentation proposed by Liu, et al. (2006) as an example of this kind of conflicts.

The other stands for conflicts occurred while actors updating design description simultaneously within or without the same disciplines or functions. For instances, this kind of conflicts may happen in computer software if there is no any restriction of user authorities like both of sales A and sales B are trying to book the same original material C for their own WIP, WIP a and WIP b, and then with the constraints of limited original material C, conflicts may occurred if there is no any decision rules or user authority restrictions of the material booking system.

The second kind of conflicts is related to coordination between activities of design process or agent communication. This kind of conflicts occurred when two subjects have certain relationships to the same object, which can be one certain design activity under the same design stage they have to cooperate simultaneously or with sequentially. Take the above example for instance. If WIP a of manufacturer A and WIP b of manufacturer B are two key components of final goods C, then to how to coordinate both of them ready on shop floor to be assembled simultaneously would be one situation of this kind of conflicts. In addition, if WIP a and WIP b owned by manufacturer A and B accordingly are required to be assembled sequentially from process a to b, which may have coordination issues of time, quantity required, function and interface compatibility, and so on, then this would the second situation of conflicts within activities of design stage.

Further more, if manufacturer A and B represents two agents either human beings or software agents, then situation of communicating within two agents whether for confirmation of product specifications or function compatibility would become the kind of conflicts within agent communication. For example, we may regard the engineering system of engine design and air-conditioning of Boeing 747 as two unrelated ones, but the interaction between these two systems should be taken into consideration together during system-level design to avoid situation of incompatibility when systems operated as a whole.

The third one is communication issues of standalone systems. Here we focus on interfaces incompatibility of two systems and the loss of associated information under situation of design changes mentioned by Mervyn, et al. (2006). The former one may occur with the lack of mutual interfaces support between systems, while the latter one may relate to lack of simultaneous retrieve repository and sound interpretable technical support within systems. Beside, we will also discuss about cooperative working capability of the system, namely on-line cooperation and off-line cooperation such as M3D framework proposed by Luo (2006).

Working capability of two different systems can be totally conclusive either within the same function or not. Even under the same design stage working on the same design engineering tasks, properties and interfaces of two systems may just have conflicts of communication. For example, in the same system-level design stage, ECAD and MCAD which are both working on product architecture modeling have to deal with interface incompatibility of information interchanges geometrically and the capability of interpreting meta-information of interchanges soundly.

2. Actors

Following the definition and classification of actors in CD activities, we want to discuss about information shared and protocols applied among agents in the section. Agents have dynamic links with each other. The dynamic links can be achieved by having common information exchange protocols, syntax and semantics for communication. The process of information exchange within agents should include input, input translator, processing, output translator, and output. For example, results of FEA files would go through process of regarding as a acceptable input messages, passing through common interfaces to the repository of product architecture design department to be processed as useful information, then with the out translator, the ECAD files of engineering design can be taken as output to other departments and end this loop. In view of life-cycle of agents, states of agents can be regarded as available, busy, and unavailable during the whole CD process. Hence, protocols within systems to support transformation and information sharing of agent states should also be taken into consideration.

3. Artifact

We use "artifact" to represents contents interchanged within multiple-discipline cooperators. Types of artifacts including data, transformation, information and knowledge would be clarified in the following sessions.

- Data

According to Mills and Goossenaerts (2001), data is defined as simply symbols with no context and no relationship interchanged with cooperators. Two groups of data can be classified as geometric data and process data during CD process. Geometric data means data interchanged among cooperators are about the geometric aspect of target product such as geometric features of the target product, geometric engineering data files, CAD files, CAD models, properties-files, design dimension, etc., while process data refers to data occurred during the collaboration process including feedback to upstream applications and data for downstream application, data about product related persons and teams and much more like that. Tools used to cope up with conflicts regarding data are like user interfaces for argumentation-based conflict resolution, whiteboards for design alternatives, and so on (Liu et al., 2006).

- Transformation

Transformation is defined as the process function, process structure, and process behavior of target product during CD. Data contents such as results of firmness-analysis (FEM), NC-Programs, function and simulation files are all instances of transformation. We give a summary of transformation types in Figure 3.43. Types of transformation can be defined by two perspectives contents related to design decision made during CD and to design tasks of CD.

In view of design decision, contents of transformation include the selection of either one or none of the alternatives from conflicting cooperators and the compromising result of refining the given substitute. In design task perspective, transformation of cooperators stands for different interchanging processes of CD, including abstraction of data interchanged among multiple-discipline cooperators for better efficiency, easier communication, and so on.

Other five types are concretization of data interchanged, composition of data interchanged, decomposition of data interchanged, mapping of data interchanged, and evaluation of data interchanged. Most of them have the same purpose to make communication easier, and other include better understanding to cooperators such "concretization", consideration of incompatible interfaces between systems such as "mapping", assessment of data exchanged to be as the basis of decision-making such as "evaluation" or simply to let collaboration easier such as "composition" and "decomposition". For example, "decomposition" tries to have the design tasks capable of being divided into sub-tasks so that concurrent engineering can take into place more easily.

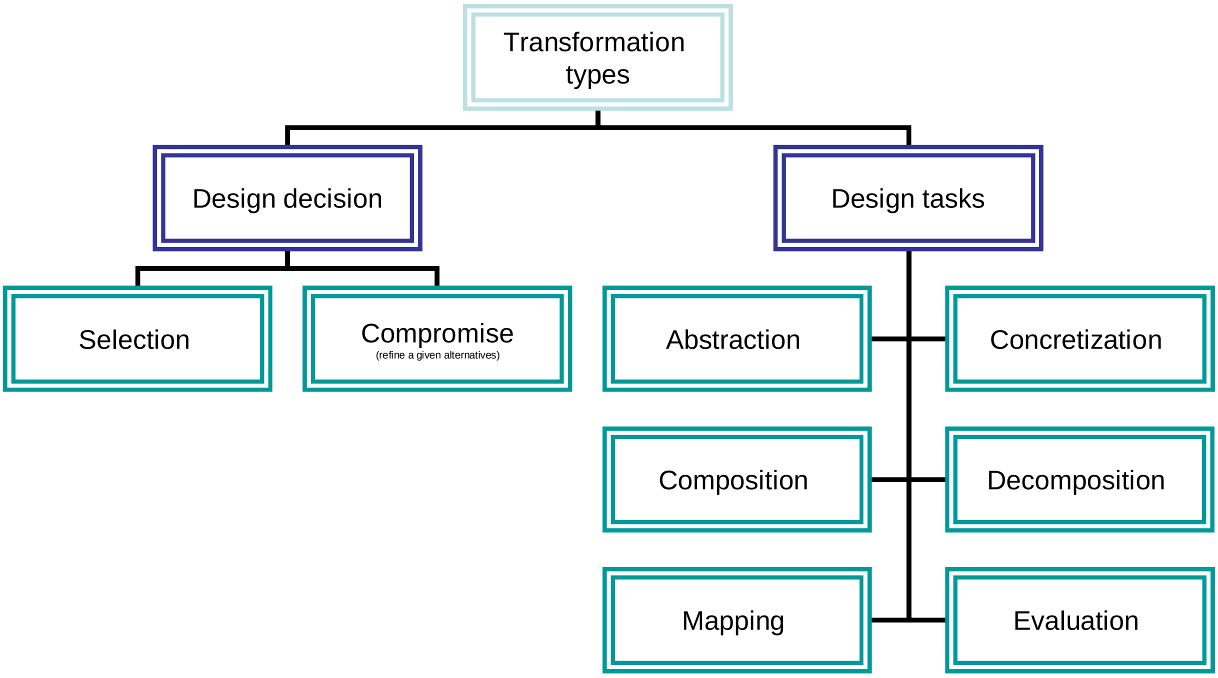


Figure 3.43 Taxonomy of transformation types (Source: original)

⌖ Information

According to Mills et al. (2001), data within a specific context is defined as information, which can demonstrate further details of product and data interchanged. To understand contents of information, we follow taxonomy of information proposed by Panchal et al. (2006) and adapt it into Figure 3.44. Information types can be approximately divided into two categories flow-information and meta-information. Flow-information represents information processed by transformation, and basically refers to the product information along CD process such as product status along the design process.

Product information emphasizes on the relationships of entities of the product, which contains product characteristics and constrains, intra-part relationships or constrains, product status along the design process, etc. (Panchal, et al., 2006). We clarify four kinds of information model representing the flow-information contents during the CD process.

Product data models, which are built to formalize product design information in order to communicate with one another more easily with standard forms, contains basic product information exchanged among cooperators; CAE/FEA/CFD models, which are made to analyze data and transformation of product functionalities and product simulations among cooperators, contains information of geometric data and engineering transformation; Product development process models, which are made for design activities control, process information flow, task dependence description, process planning, and so on, provide information to stable the CD system during collaboration and information about configuring the product process itself; Product performance models , which are put up for controlling information regarding quality, cost, Time To Market (TTM) management and design process improvement, provide assessment of information interchanged to facilitate design decision-making.

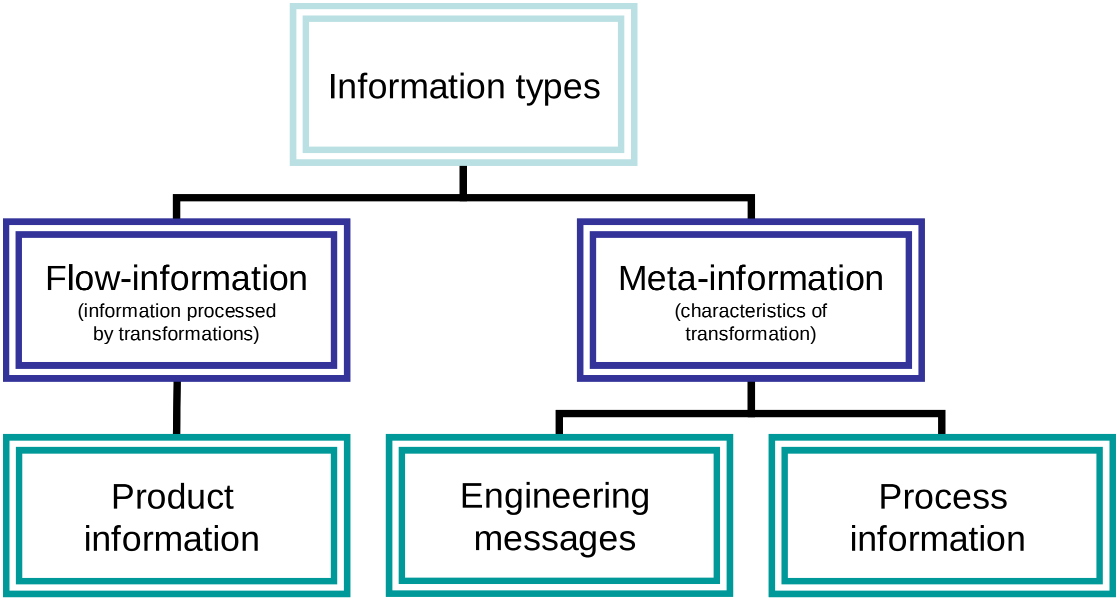


Figure 3.44 Taxonomy of information types (Source: adapt from Panchal, et al., 2006)

Meta-information represents characteristics of product transformation, and contains engineering messages and process information during CD. Engineering messages stand for information related to product engineering design of CD such as specifications, meta-data (information and interpretation of the data), and resource constrains, etc., while Process information emphasizes on information of sequences and relationships of the design process activities. Three main categories of process information proposed by Panchal et al. (2006) includes architecture of the design process (such as flow of information between tasks or activities, execution priority and time sequence of the tasks, etc.), the manner in which individual design activity is carried out (such as design process flow of certain product realization, etc.), and the parameters of the design process (such as process attributes and specifications, design of experiments (DOE) parameters, etc.).

- Knowledge

Mills, et al. (2001) define knowledge as information relationships within and across contexts. Types of Knowledge (knowledge used within the CD activities) can be classified as explicit knowledge, which is documented, and implicit knowledge, which are unwritten experience of employees. By categorization proposed by Bullinger, et al. (2007), explicit knowledge contains Knowledge of facts (know-what), including design knowledge and design process knowledge; implicit knowledge on the other hand, include process design knowledge and design capability knowledge.

Design knowledge stands for knowledge to design products/ services, including visible and invisible capital such as product design model database and CAx tool box, and so on; Design process knowledge, which is same with the concept of know-how knowledge mentioned by Bulleringer et al., stands for knowledge regarding building the design process and its impact on the design performance or assessment, and relationship built between the entities of the design process; Process design knowledge, which is same with the concept of know-why knowledge mentioned by Bulleringer et al., refers to knowledge regarding setting up or modeling the process of design; Design capability knowledge, including the concept of care-why knowledge mention by Bulleringer et al., represents knowledge used for product design, such as the level of creativity , design capability, and so on.

KQML (Knowledge Query Modeling Language), K I F (Knowledge Interchange Format), and ACL (Agent Communication Language), etc. are commonly used as language for knowledge interchanged, which is also a significant issue. Knowledge is not similar to data, transformation, and information discussed above, it does need a certain language used for knowledge representation and following the standards for communication. As what mentioned by Robin et al. (2007), four types of knowledge shared during CD process is shown in Figure 3.45, and indicates that contents of knowledge basically should contain the five following parameters, which can be represented by following equation:

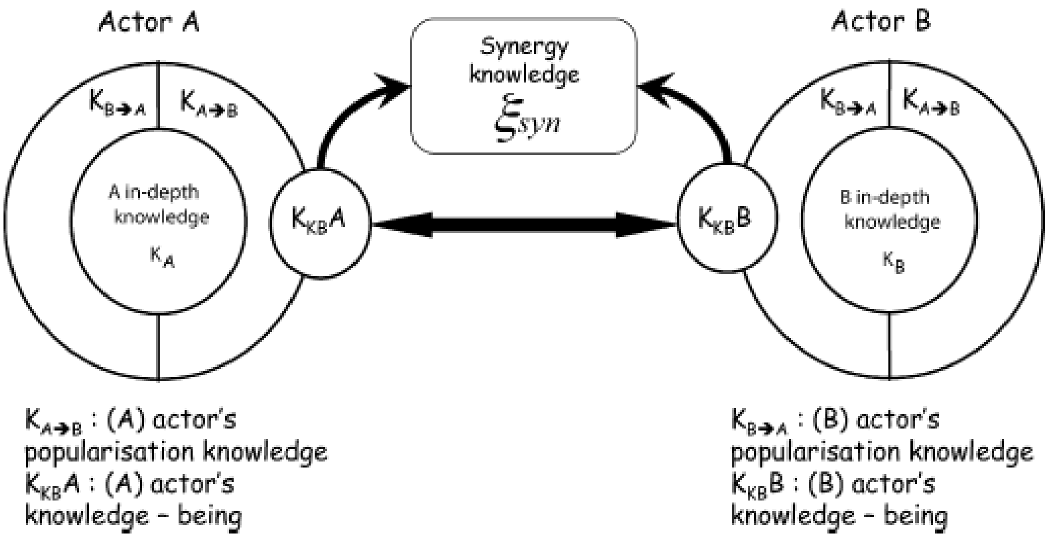


Figure 3.45 knowledge shared during collaborative design process (Source: Robin, et al., 2007)

$$K_g(A) = K_A + \sum_{i=B}^{N} K_{i-A} + \sum_{i=B}^{N} K_{A-i} + K_{KB} + \xi_{syn}$$

In this equation:

$K_g(A)$ is actor's general knowledge;

K_A is the actor's own expertise;

$\sum_{i=B}^{N} K_{i-A}$ is the addition of the popularisation knowledge acquired by the actor, coming from the other memebers of the group;

$\sum_{i=B}^{N} K_{A-i}$ is the addition of the popularisation knowledge distributed to other actors of the design project;

K_{KB} represents knowledge-being of the group, depending on the culture of each actor;

ξ_{syn} represents synergy knowledge which is necessary to develop the process of popularisation of knowledge between all the actors.

In Figure 3.45, knowledge shared during collaborative design process contains Popularisation knowledge acquired by the actor, coming from the other members of the group; Popularisation knowledge distributed to the other actors of the design project, which supports for problem solving; Knowledge-being used by each actor when he has to initiate communication with the other actors. It can be seen as interface ports to reach other actors of the surrounding context; Synergy knowledge, implemented to carry out and maintain the intra-group knowledge exchanges. It is a support of communication. (Robin, et al., 2007)

3.2.5.3 Process of communication

Communication is the basis for cooperation. It guarantees the continuous exchange of data, information, and knowledge. Particularly dynamic processes, like the development of innovative products, demand great willingness to communicate from the developing partners, especially when the partners have not worked together before. The dynamic in network of cooperating partners requires a higher degree of communication than most companies are used to because spontaneous agreement concerning the further development process are often necessary (Bullinger et al., 2007). In this section, we will introduce process of communication and information interchanged during the CD process, elements of CD framework such as technologies used, and classification of framework types.

1. Advantages of communication

Contemporary product design is characterized by design teams developing a product collaboratively. Since changes initiated by designers often interact with the work of colleagues, transactions want to see the results of parallel transactions and access shared data, knowing that this data may be incomplete and inconsistent. These intra-transaction interactions lead to definitively non-serializability making the traditional transaction model not appropriate for knowledge bases. In A, the parallel work of the users is serialized. In B, the consequences of the cooperative transaction model are sketched. The design work of user 1 is interrupted and a communication is being done with user 2. After this communication, the design work can cooperative with effectiveness and efficiency. (See Figure 3.46, Roller et al., 2002)

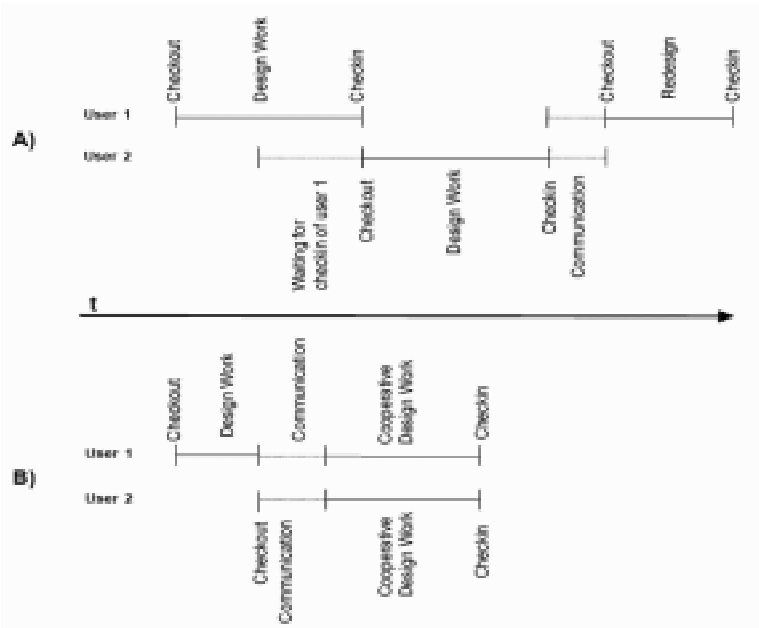


Figure 3.46 Comparison of productivity of parallel design activities (Source: Roller, et al., 2002)

2. CD Framework

In this session, the critical functions should put into considerations for facilitating CD will be collected and the elementary components of CD framework will be defined.

According to solutions provided by *Actify*, business of today shall be able to not only use product data in collaborative activities, but also have the need to be able to find the relevant data, effectively and safely share that data with others, and to further manage the data involved in the collaborative process. Hence, before allocating a CD framework, we have to make sure that product data visualization, which is the most fundamental function one shall think over, supply chain management, intellectual property protection, and product innovation management are all the four that should be taken into considerations. In addition, to facilitate a CD framework, three aspects of technologies applied should paid attention to, which are technology of collaboration operation management such as collaboration process integration via web-service, infrastructure technology for service interoperability and information security, and technology supporting CD such as the CAX tools environment for CD, etc (Liu, 2003).

As the CD frameworks proposed by Panchal et al. (2006), necessary elements of the CD frameworks should include Data repository, Web browser, Process diagram tool, Interface mapping/integration tool, and Dynamic UI (User Interface) Generation. Purpose of data repository is to make information storage and retrieval done rightly in use of multiple-discipline cooperators. Representation schema of database can call on framework proposed by Tang et al. (2004). Process diagram tool is used to model a product realization process, and then invoke the available agents integrated into the framework; interface mapping and integration tool is applied to convert the output from one agent into a format compatible with the input to another agent. Mapping mechanism can take a reference from Ma et al. (2006).

Examples of frameworks developed so far are such as X-DPR (eXtensible Distributed Product Realization) Framework proposed by Jitesh et al. (2006), M3D (Multi-site cooperative 3D design for architecture) Framework proposed by Luo et al. (2006), and so on. The engineering solution center (ESC) concept brought up by Bullinger et al. (2007). Present Commercial product such as Alibre Design is a collaborative solid modeling tool for creating 3D designs and 2D drawings. It allows engineering teams to work together concurrently over the internet to create, visualize, review, and modify their designs and drawings.

CD frameworks types can be divided into four kinds of basis by literatures reviewed, including: (1) Web-based such as Web-DPR, (2) Agent-based such as DOME and NetBuilder, (3) Product-centric, and (4) Process-centric, as shown in Figure 3.47.

Product-centric frameworks regard each agent models as a sub-system of the artifact such as framework DOME, and so on. In product-centric perspective, attentions have been addressed on how to exploiting the reusability and scalability of products through product platform and product family design. (Panchal, et al., 2006), whereas process-centric focuses on taking each agent model as an activity in the design process, such as mechanism used in framework NetBuilder.

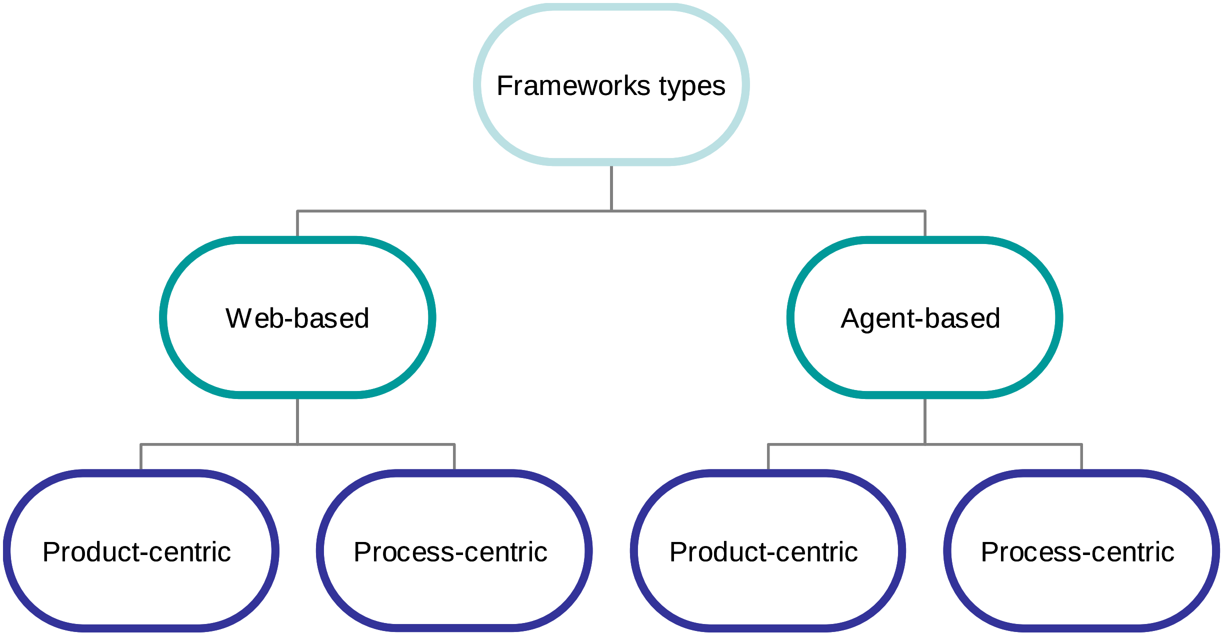


Figure 3.47 Taxonomy of framework types (Source: adapted from Panchal, et al., 2006)

In accordance to different collaboration environments, CD framework would also do some adjustments to fit the situation; however, elementary components of CD framework should never be omitted. Based on elements of Communication infrastructure and framework architecture proposed by Panchal, et al. (2006), we make a summary of elements of CD framework in two ways: one is standards and languages for communication of CD; the other is technologies used for communication of CD.

3. Standards applied of CD framework

In standards and languages for communication, common vocabularies, domain ontology, semantic schema, and mappings are defined to represent, interpret, map, and share the semantically interoperable product information across collaborating applications. Standards are used for product information exchange, sharing, and interoperability. Based on modeling mechanism proposed by Yang and Zhang (2006), standard file formats are as following, which we can take them as a standard in building up communication framework for CD:

1. Property Relationship Extension Mechanism: behavior and property

objects are defined into property sets (Pset), and the relationship entities are specified to assign the relating Pset to the related entities. Example of Pset is <Name, Description, GlabalId, HasProperties>, which stands for properties of product name, descriptions of design or specification, identification number of the product, and the product properties.

2. Subtyping Extension Mechanism: definitions of new entities are created as subtypes of an existing entity, which can be used for creating hierarchy relationship among information shared.

Ontology represents formal, explicit, and shared understanding about application semantics, domain concepts and their relationships, which can be distinguished into two categories: Logic-based ontology and Non-logic-based ontology (Yang, et al., 2006). Logic-based ontology explicitly specified the semantics of terminologies through definitions and axioms such as OWL (Web Ontology Language), while Non-logic-based ontology does not use axioms to specify the semantics of terminology.

Examples of Non-logic-based ontology are like XML (eXtensible Markup Language, an industry markup language used for representing data in a platform independent manner) schema, STEP (Standard for the Exchange of Product Model Data) specification, ebXML (e-business XML) (Gibb and Damondaran, 2002), IGES (Initial Graphics Exchange Specification), KQML (Knowledge Query Manipulation Language) technology, VRML (Virtual Reality Modeling Language), SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) schema, etc..

Standardized message protocols such as CORBA protocols are also often used for communication of CD activities. The framework for communication protocols between a requester (client) and a provider (server), as an extension to the well known Contract Net protocol (Parunak, 1987; Figure.), involves at least seven stages:

(1) Preparation of the proposal or request by the requester;
(2) Sending of the proposal or request from the requester to the provider;
(3) Acceptance of the proposal by the provider (i.e. understanding the proposal and expressing the willingness to do it);
(4) Execution by the provider of what is proposed by the requester;
(5) Preparation by the provider of the result of the proposal;
(6) Returning of the results of the execution back to the requester by the provider;
(7) Acceptance of the results by the requester.

4. Communication protocols for CD framework

Abdalla (2006) give us three examples of communication protocols at different interaction stages, and in this protocol for each message sender object to get the final response it needs to send at least two messages as shown in Figure 3.48 and Figure 3.49, which can be regarded as a instancing for better understanding the concept of what a protocol is and the role it play in the communication activities.

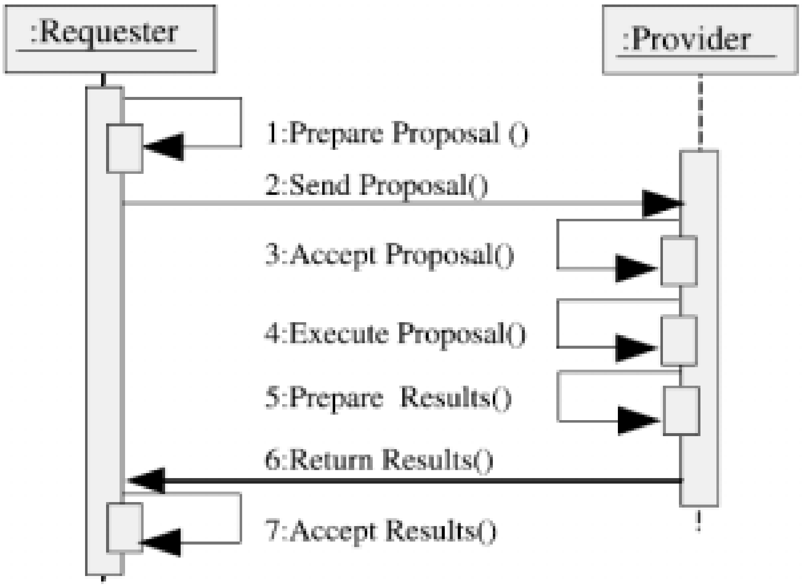


Figure 3.48 Sequence diagram for a communication protocol (Source: Parunak, 1987)

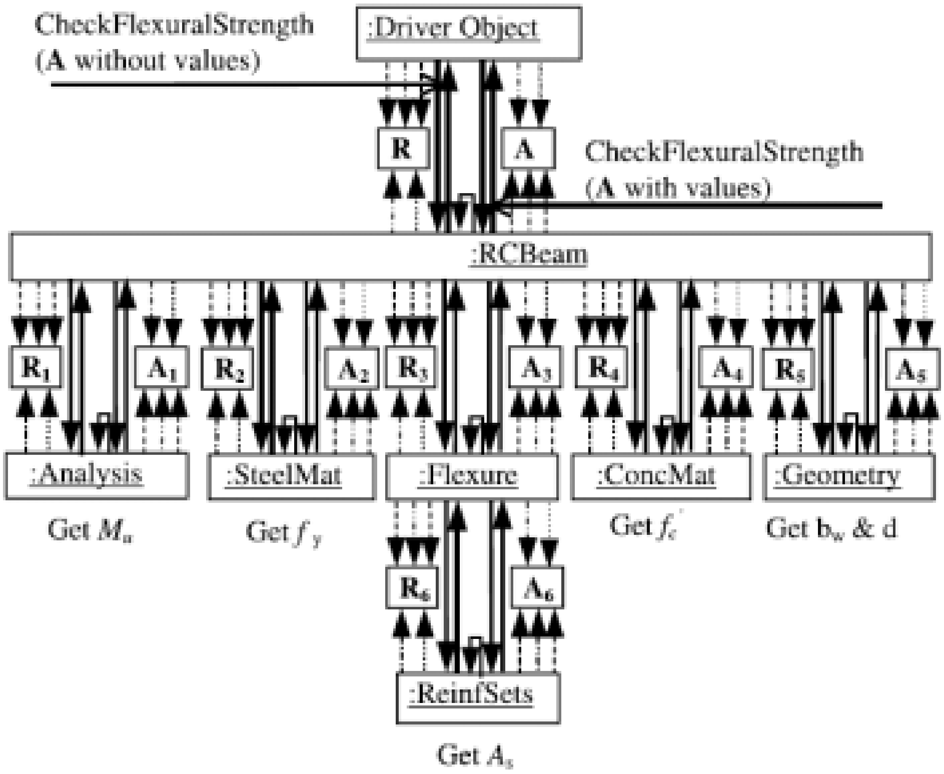


Figure 3.49 Collaboration diagram of the Conversational protocol for CheckFlexuralStrength Message (Source: Abdalla, 2006)

Communication primitives supporting conflict resolutions provided by Brelinghoven (1995) are used to define protocols which allow designers or tools to communicate directly as conflicts occurred. With supports of semantics schema mapping knowledge, ontological relationship and mapping knowledge information can be fully controlled. From Figure 3.50 semantically equivalent concepts in both the supplementary and the STEP extension definitions are mapped to each other based on the mapping knowledge mentioned before (Yang and Zhang, 2006). With connecting mapping schema described above, information and knowledge interchanges among multiple-disciplines cooperators would be able to take place under well-defined formats and rules.

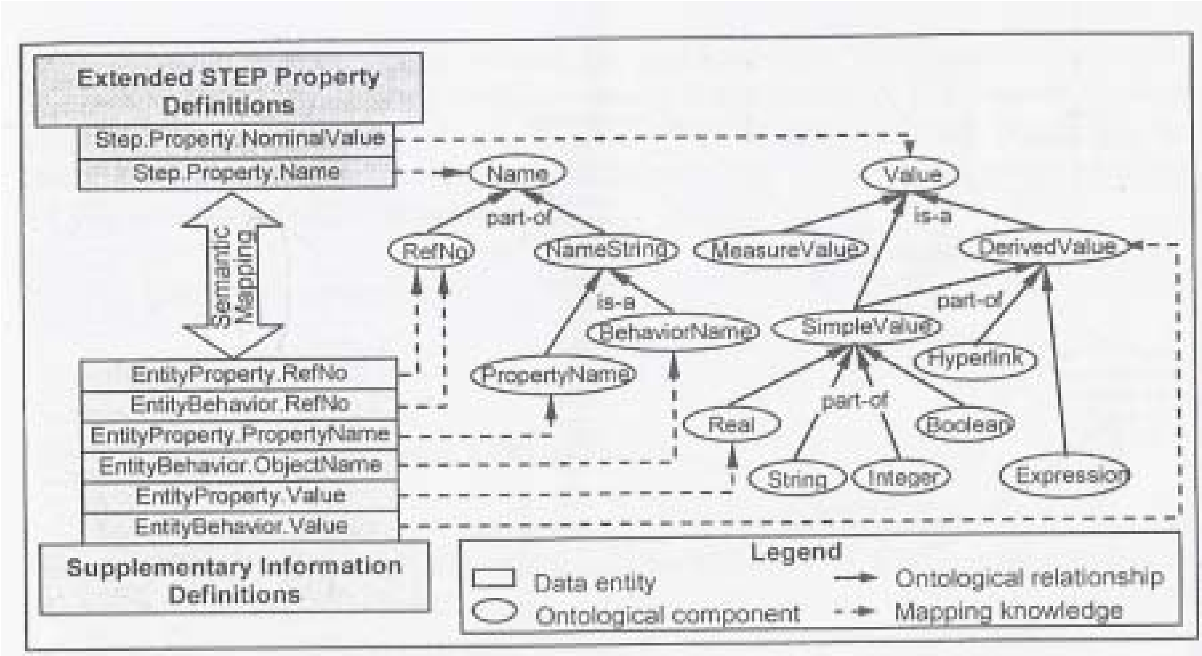


Figure 3.50 Semantics-driven schema mapping (Source: Yang and Zhang, 2006)

5. <u>Technologies used for communication</u>

The distributed teams need technical support for the development of a product to enable synchronous and asynchronous interactions. ATM (asynchronous transfer mode) networks and gigabit Ethernet networking enable a quick and safe exchange of relevant data and thus support he development process tremendously. Communication and cooperation are further supported by CSCW (computer-supported cooperative work) tools like videoconferencing and e-mail (Bullinger et al., 2007). The following is collection of technologies used for infrastructure and facilitate CD.

CD structuring technologies like Java, web browsers, and so on, are used to build up the CD infrastructure; Semantic web and web services technology is applied to engineering information management and knowledge sharing (Yang, et al., 2006). Virtual environment using VR (Virtual Reality) Technology such as virtual-based collaborative environment (VRCE) is used to enable cooperators communicate within virtual environment for better efficiency and effectiveness on interactions and collaboration (Chryssolouris, et al., 2006). Middleware technologies play the role of building the bridge within different discipline collaborators as the translators, which include distribution middleware technologies, dealing with connectivity issues, i.e., how programs on different computer can connect to one another, and domain specific middleware technologies, providing domain specific middleware technology contains geometric modeling services and process data exchange services (Mervyn, et al., 2006).

3.2.5.4 Taxonomy of CD information framework types

In this session, we will introduce the CD information scenarios first, and then make a description of our reference CD information frameworks. Further applications would be interpreted in chapter four.

1. CD information framework scenario

CD framework can be classified according to participants and scale. As mentioned earlier in section 3.2.1, we may take participants involved in a CD as either designer or non-designer. The CD scale can be separated as within the same corporation or cross different one. In the former one, two cases are covered in the following session. In the designer-to-designer case (Figure 3.51), the elementary CD information framework should include the five fundamental elements we mentioned earlier in this section, namely Data repository, Web browser, Process diagram tool, Interface mapping/integration tool, and Dynamic UI (User Interface) Generation. Besides, we shall have a role of system administrator to monitor and mediator all the progressing process of CD, and to ensure smooth operation of interfaces and application systems.

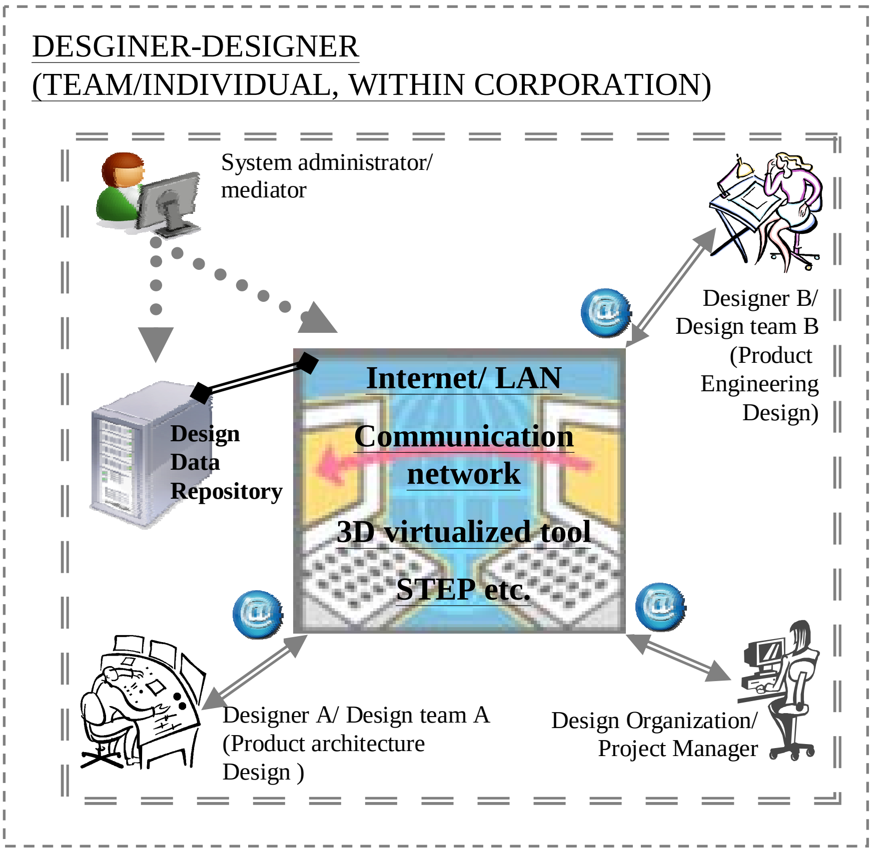


Figure 3.51 scenario of designer-to-designer CD information framework within corporation

(Source: original)

In the designer-to-non-designer case (Figure 3.52), we can see the more hierarchical structure of CD, which can be regarded as the evolving one of designer-to-designer. Built above the elementary information structure among all designers, we still have to add in more figures such as marketing supporters, corporation communicators, engineering consultants etc. in a more complex but radical situation. In the sub-level of information framework, the designers (R&D part in Figure 3.52) may be the case in Figure 3.51. The difference of the role of system administrator played in this case falls on its capability to control both repository and platform system but above the hierarchy architecture.

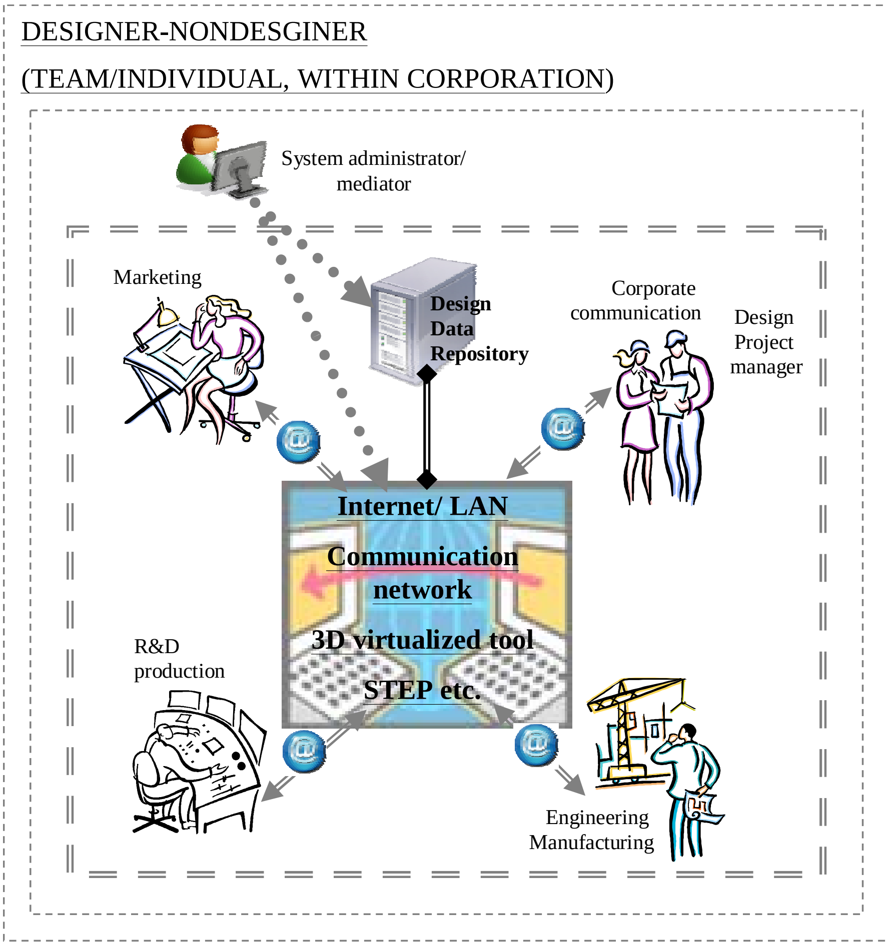


Figure 3.52 scenario of designer-to-non-designer CD information framework within corporation (Source: original)

Once the CD takes place cross two individual organizations, which may be cross simply companies or even with different industry characteristics, the third-party guaranteed repository should be put into the framework for justice and fairness. It can be a repository run by the third-party or be the platform provided by the third-party and run by the system administrators of CD cooperators. (See Figure 3.53)

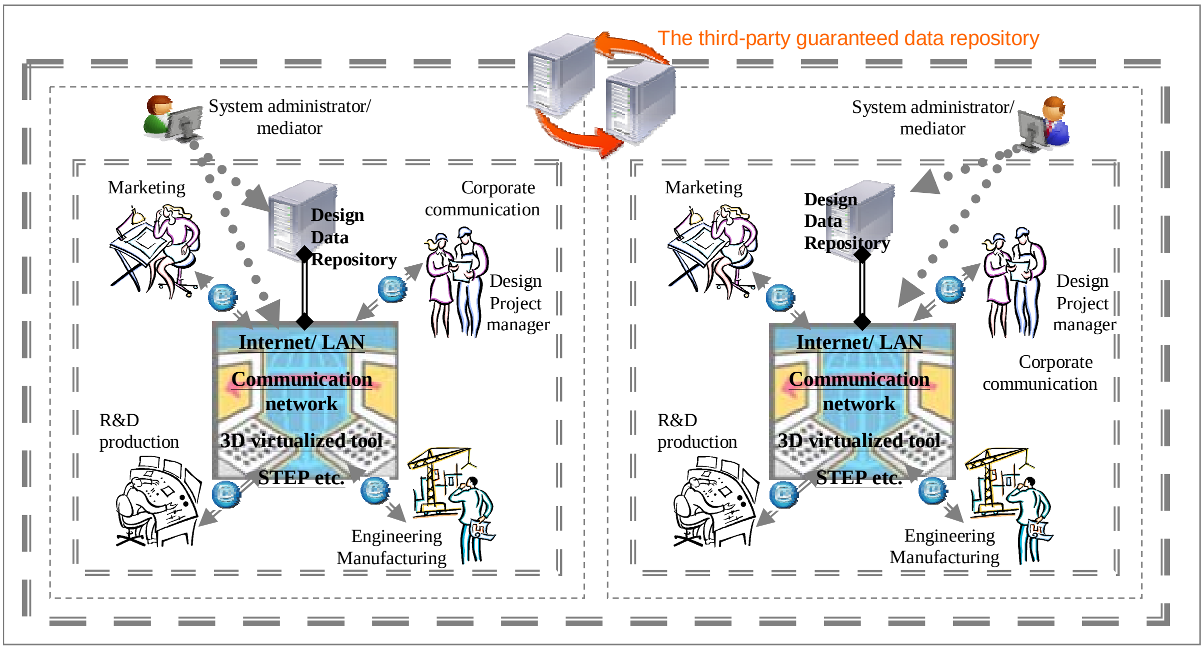


Figure 3.53 scenario of CD information framework cross corporation (Source: original)

2. Reference CD information frameworks

We use the organization and participants structures as our interpreting settings. CD can take place in three circumstances, namely mono-participant CD, multiple-participant CD, and network CD.

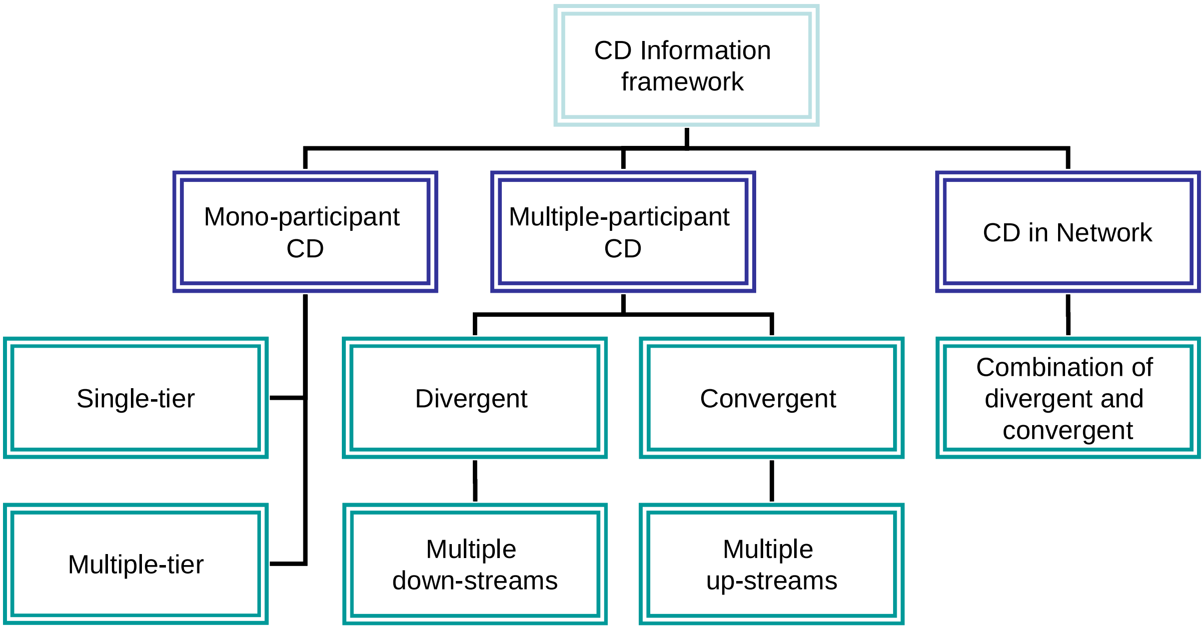


Figure 3.54 the CD Information framework (Source: original)

The first one represents participants involved in each tier of CD is one-to-one relation such as supplier-buyer or one-tier supplier to second-tier supplier, and so on. Under this kind of case, sub-scenarios can be separated into single-tier and multiple-tier. (See Figure 3.55 & Figure 3.56)

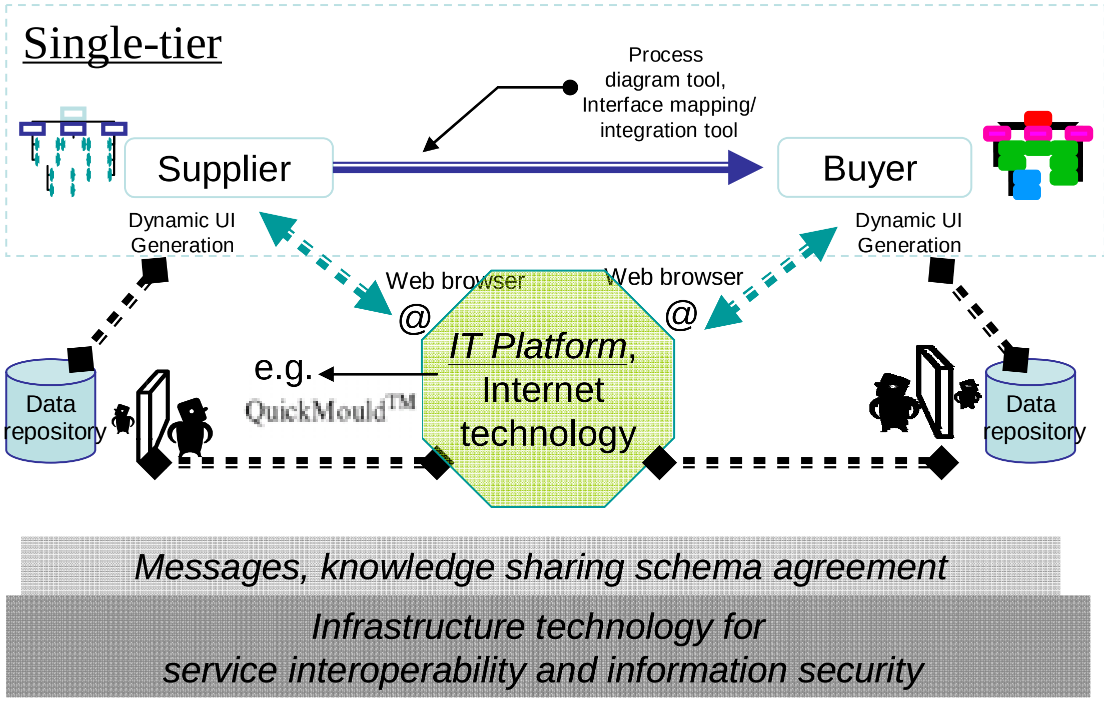


Figure 3.55 mono-participant CD: single-tier relationship (Source: original)

Both of the scenarios contain the five elements to form a CD information framework. In Figure 3.55, cooperators communicate through the IT platform, or with the support of Internet technology. Each of them possessing their own data repository, and have the right and authority to decide how far do they want to and have to share their information and knowledge with each other. With the support of dynamic UI, they can apply process diagram and interface mapping tools to transform the demands of product specification or terms from buyer to supplier for instance. A situation of information framework is expended into multiple-tier in Figure 3.56.

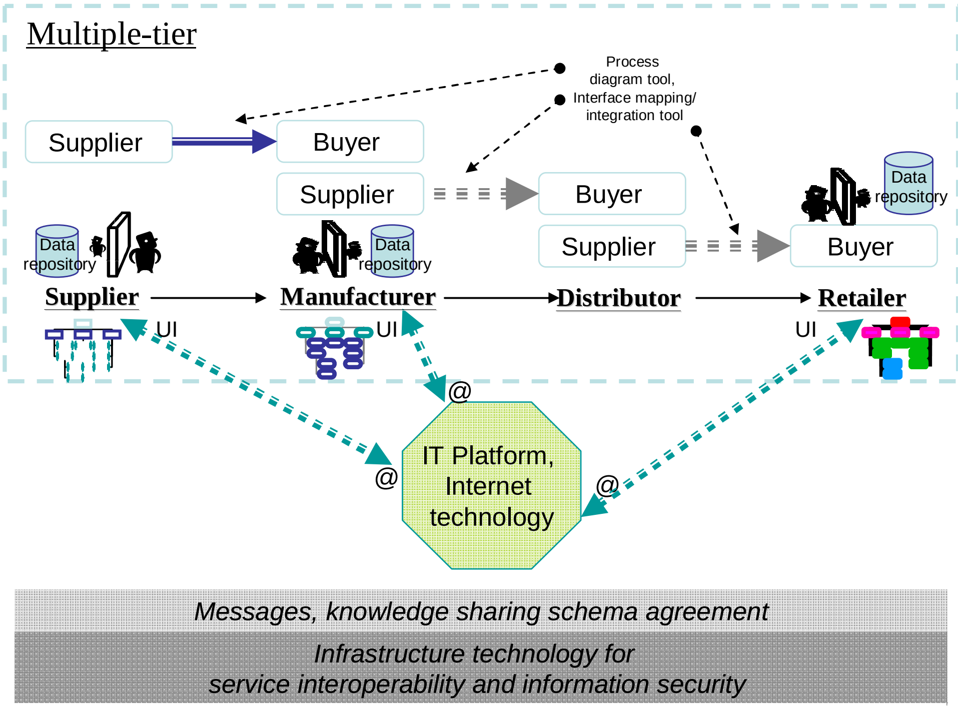


Figure 3.56 mono-participant CD: multiple-tier relationship (Source: original)

The second scenario is multiple-participant CD, and is divided into divergent and convergent ones. The former represents multiple down-streams cooperators, while the latter represents multiple up-streams cooperators (Figure 3.57).

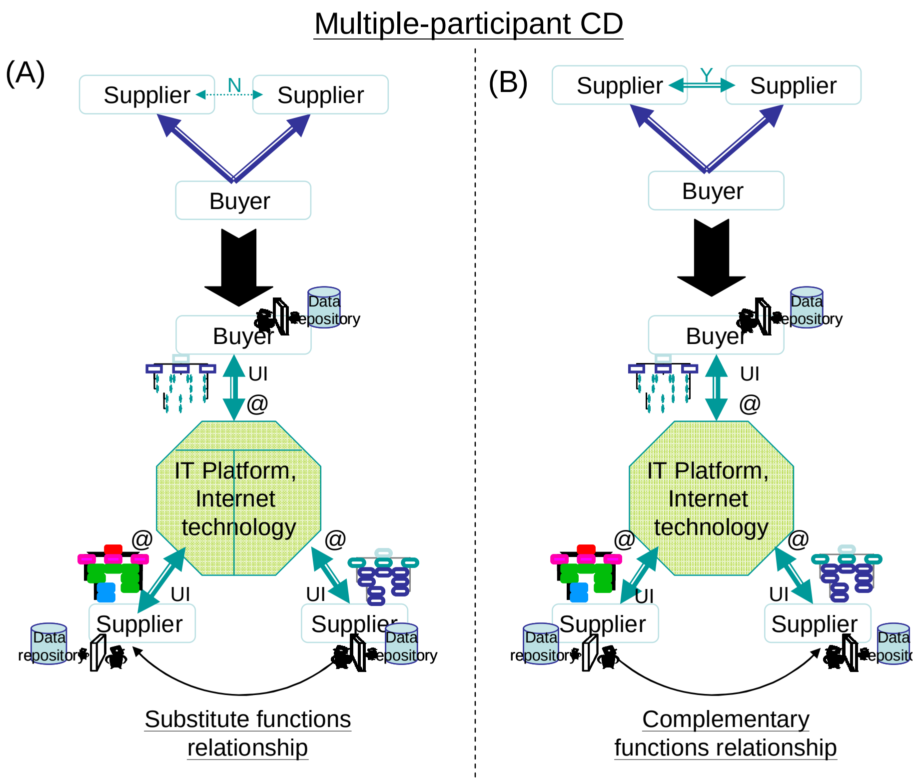


Figure 3.57 Multiple-participant CD information framework: Multiple up-streams (Source:

original)

One buyer may have multiple CD activities processed concurrently, or even make many up-streams cooperators to cover the same design task/engineering. Under this kind of situation, we can tell there are two main circumstances. One is permission of interacting and interchange information with the other up-streamer (see Figure 3.57-B), and the other is let each of them process separately, and the buyer (primary actor here) responds for the final and connecting works (see Figure 3.57-A).

Case of Figure 3.57-A is tempt to fall on the substitute functions relationship of the up-streams. Each of the suppliers can only connect to the authorized access of the platform, while the primary actor the buyer controls the over all picture of the CD. If it is the case of Figure 3.57-B, all entities involved in the CD has the same hierarchical status and the authority to share information equally through the media (Internet Technology (IT) platform). This kind of CD is prone to be the relationship with complementary functions of the up-streamers. The CD tasks issued by primary to the two secondary are highly correlated with one another. The design tasks are maybe in the sequential relations or simply complementary functions requirements need to be fulfilled.

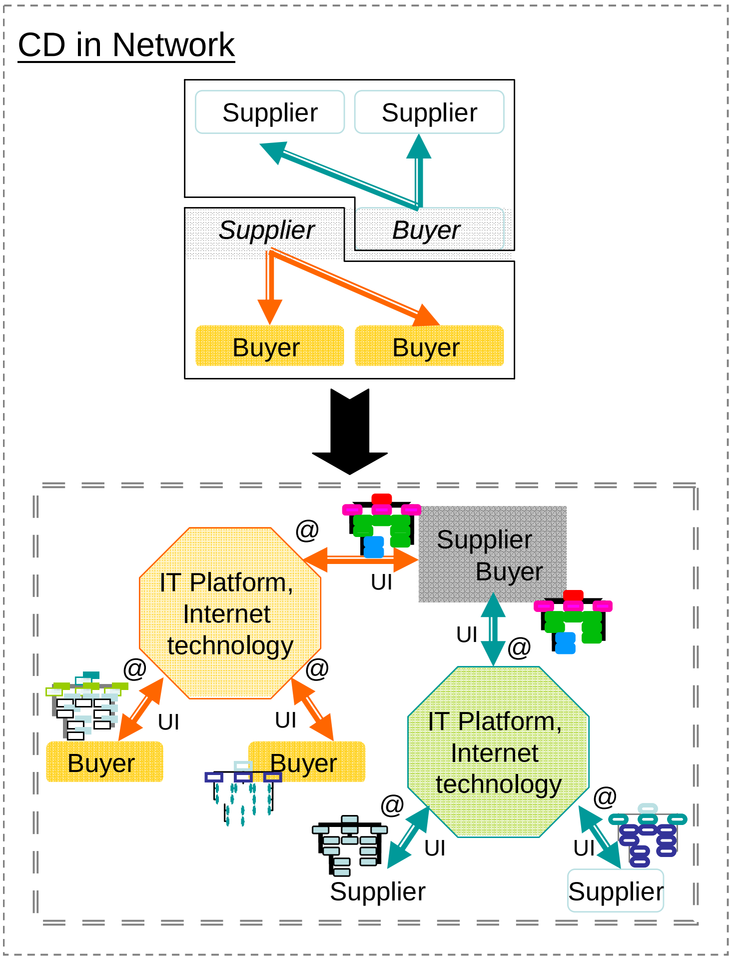


Figure 3.58 information framework of CD in Network (Source: original)

The last one is the information framework of network CD (See Figure 3.58), which is the combination of the multiple up-streams and down-streams cases. The middle layer has dual identifications as being the supplier and the buyer at the same. CD under this scenario should allocate two separated information sharing media (IT platform, Internet Technoogy) to up-stream and down-stream. Other elementary settings and types choices would follow statements mentioned earlier in this session.

We will introduce further information reference frameworks by its applying collaboration scope in details in chapter four.

3.3 Design stages

Design process can be defined as networks of information transformations from one state to another, the state of information refers to the amount and form of that information that is available for design decision-making (Panchal, et al., 2006). Stages/processes of product design and development can be basically defined as process of Idea/Requirement, Concept Design (CD), Design Build (DB), System Integration (SI), Product Validation (PV), Manufacture Validation

(MV), and Mass Production (Ramp/MP) (DoIT，2005). Ulrich and Eppinger (2008) also have similar view on stages/processes of product design and development, and they make a definition as six steps planning, concept development, system-level design, detail design, testing and refinement, and production ramp-up. Three significant types of design processes proposed by Holt (1990):

1. The analytical design process: used when there is little uncertainty about the alternatives, and the outcome is only a modification of something already exists.
2. The iterative design process: which is best suited to medium-risk projects such as radical improvements and adopted innovations.
3. The visionary design process: in which the problem cannot be defined precisely and is, perhaps vague at best.

Brigitte (2003) gives a more precise viewpoint on activities of design which is taken as a creative process, and divides it into three main phases: an analytic stag f widening the observation field, a synergic stage of idea and concept generation, and a final stage of selecting the optimal solution. The creative process corresponds to five phases, each of which has a different objective and correspondents to the production of more and more elaborate visual outputs (see Table 3.8).

Table 3.8 the design process (Source: Brigitte, 2003)

Phases	Objectives	Visual outputs
0. investigating	Idea	Brief
1. research	Concept	Visual concept
2.exploration	Choice of style	Roughs of ideas, sketches Roughs of presentation Reduced-scale model
3. development	Prototyping details	Technical drawings Functional model 3D mock-up for visual correctness and working capabilities
4. realization	Test	Documents of execution Prototyping
5. evaluation	Production	Illustration of the product

Cooper (1998) introduces the STAGE-GATE^{TM} process of design into six stages ideation, preliminary investigation, detailed investigation, development (money gate), testing and validation, and market launch. Brigitte combines theories of process proposed by Cooper, and Ulrich & Eppinger into following results (see Table 3.9).

Table 3.9 The STAGE-GATETM process of design (Source: adapted from Cooper, 1998)

	Phase/ gates	STAGE-GATE™	Team key activities	Design in NPD (Ulrich & Eppinger)
Stage 1.	Begin step 1 Stop or Go on to:	Ideation	Initial screening	Planning / exploration •Consider product platform and architecture •Assess new technologies and needs
	Begin step 2 Stop or Go on to:	Preliminary investigation	Market assessment Technical assessment Business assessment	Concept development
Stage 2.	Begin step 3 Stop or Go on to:	Detailed investigation	Market research •User needs and wants studies •Value in use studies Competitive analysis Concept testing Detailed technical assessment Manufacturing appraisals Detailed financial analysis	System-level design •Generate alternative architectures •Define major sub-systems and interfaces •Refine industrial design
	Begin step 4 Stop or Go on to:	Development (money gate)	Product development	Detail design •Define part geometry • choose materials •Assign tolerance •Complete Id documentation
Stage 3.	Begin step 5 Stop or Go on to:	Testing and validation	In-house product testing Customer test for products Market tests	Testing and refinement •Test of reliability, life, performance •Regulatory approvals •Implement design changes
	Step 6	Market launch.	Trial production Precommercialization business analysis Production start-up Market launch	Production ramp-up •Evaluate early production output

Before determining stages of product design, it is in what kind of topic domain affecting the criteria used for defining design process stages that we are going to discuss about.

Most of researches are focus on new product development (NPD) or topics about rapid production, which refer to general product development design process, while some are focusing on the specific domain of industrial design. For instances, Liu (2006) provides us the process framework on domain of industrial design with not only the flow of all the contents during design processing, but also the documentation of stage output is also defined precisely. Stages of industrial design process defined by Liu are stage of exploration (demand collection and analysis), planning (projects proposals, etc.), design (definition of part specifications, 3D-vector modeling, product structuring design, etc.), engineering (product modeling, product fabrication, Engineering

Certification Request 、 Engineering Change Notification, etc.) and implementation, which can corresponds with other researches. The only difference of Lui's model is the domain specific application of industrial design. In this thesis, we focus on the general CD cases but specific industrial design to cover a more comprehensive scope of the CD issues. (See Figure 3.59)

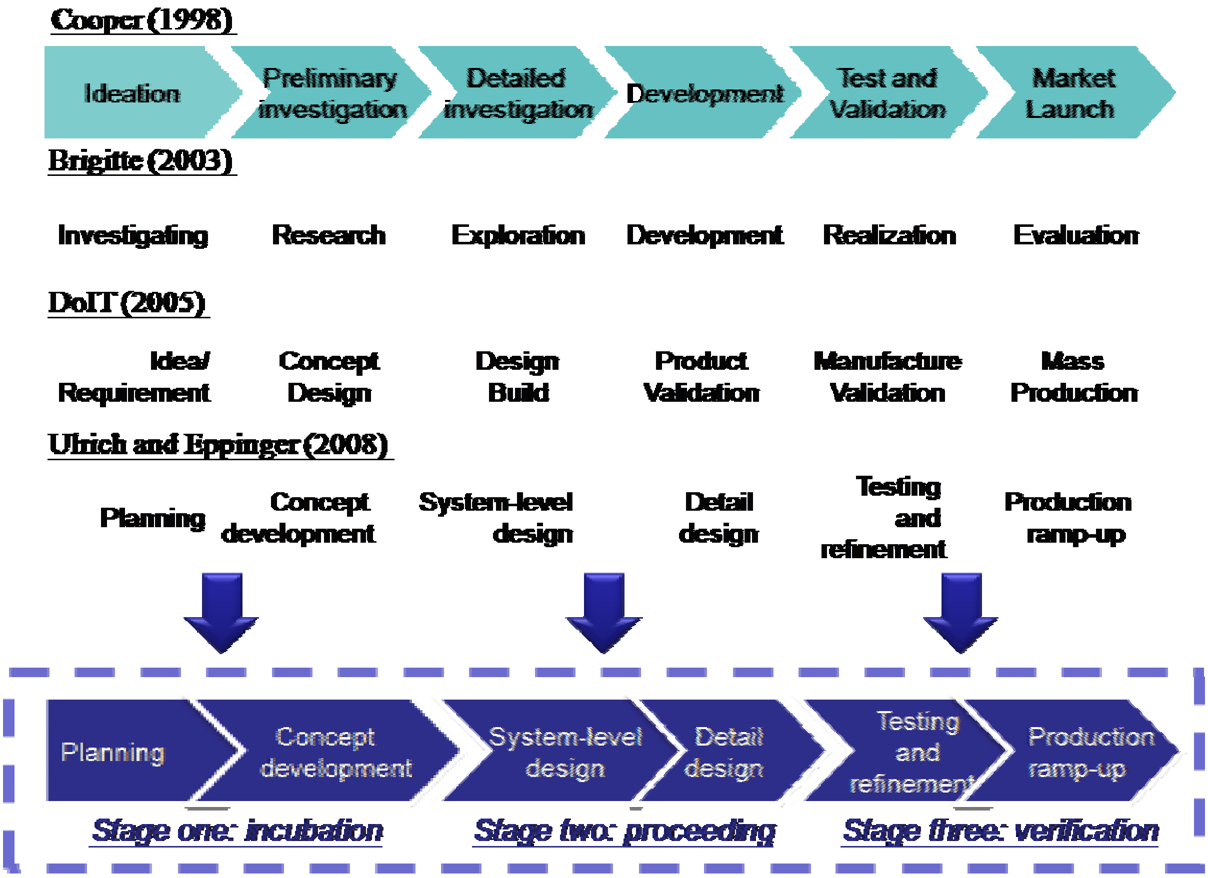


Figure 3.59 Collection of product design stages (Source: original)

In conclusion, to make a comprehensive definition of product design process of CD, we basically based on process proposed by Ulrich and Eppinger (2008), which is also acknowledged as the standard process of product design and development adopted in practical world, and condense the six steps into three major stages:

3.3.1 The first stage of CD: incubation stage

In the first design stage, collaboration is basically focusing on the preparation stage ahead of the precise designing practical tasks before next stage. Thus issues of this part are primary about the coordination related to CD project/process planning and product concept developing containing design tasks such as identification of user needs, technical factors, the diverse requirements of the operating environment, product exploration, and concept development. Detailed tasks include initial design idea collection, information pooling from present market, confirmation of customer needs, assessment of new technologies and needs, assessment of market demands, application for forming the design project, gaining permission for triggering following activities, consider product platform and architecture, and proposals of product concept designing. Keinoneon and Takala (2006) demonstrate the concepting design process consists of a series of stages, which as Figure 3.60 shows below, containing information gathering, brainstorming, scenario creation, concepts, formalization, evaluation, and final integration with project planning.

Figure 3.60 Concepting design process (Source: adapted from Keinoneon and Takala, 2006)

Following the concepting process above, they illustrate the activity layers of product concepting for us for better understanding of activities involved. Basically, the planning and concepting stage composed of three sub-categories namely background research, concept generation, and concept evaluation, and the details concerned of the three accordingly, as shown in Figure 3.61.

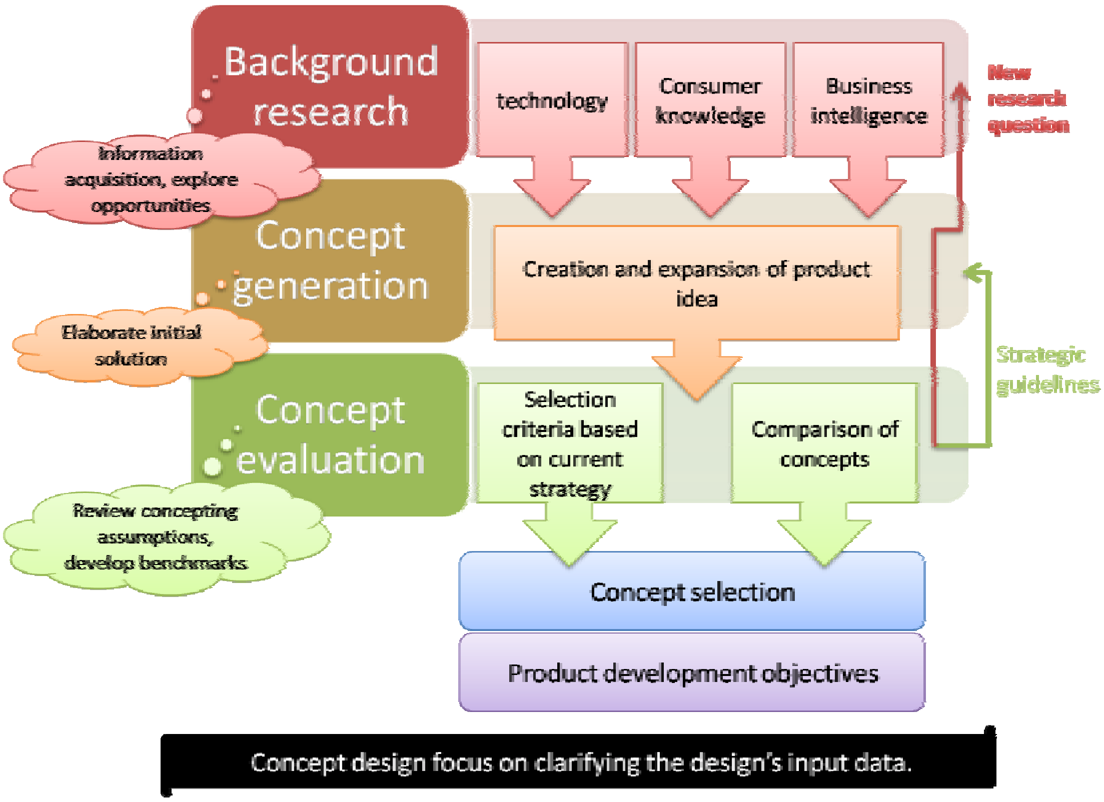


Figure 3.61 the activity layers of product concepting (Source: adapted from Keinoneon and Takala, 2006)

3.3.2 The second stage of CD: proceeding stage

In the second one, subjects being focused of this stage are primarily about the specified design tasks regarding product structure and architecture. CD issues of this stage would be the most complex one for coordination is related to the design activity itself. Combining propositions recent researches, second stage contains major design tasks such as product material and technology defined, new product design specifications, system-level design, detailed designs, resource allocation, selected concept design confirmation, current product analysis and market survey, and so on.

Conflicts and arguments involved of stage two may fit in the scenario proposed by Liu et al. (2006). They define the components of the design argumentation may include stakeholders, requirements, conflicts, design issues, parts, alternatives, arguments, ands decisions as shown in Figure 3.62. Dealing with collaboration issues under this stage, consideration of CD configuration can be divided as two perspective as design-in and spec-in. (see Hsu, 2004; Chen, 2008)

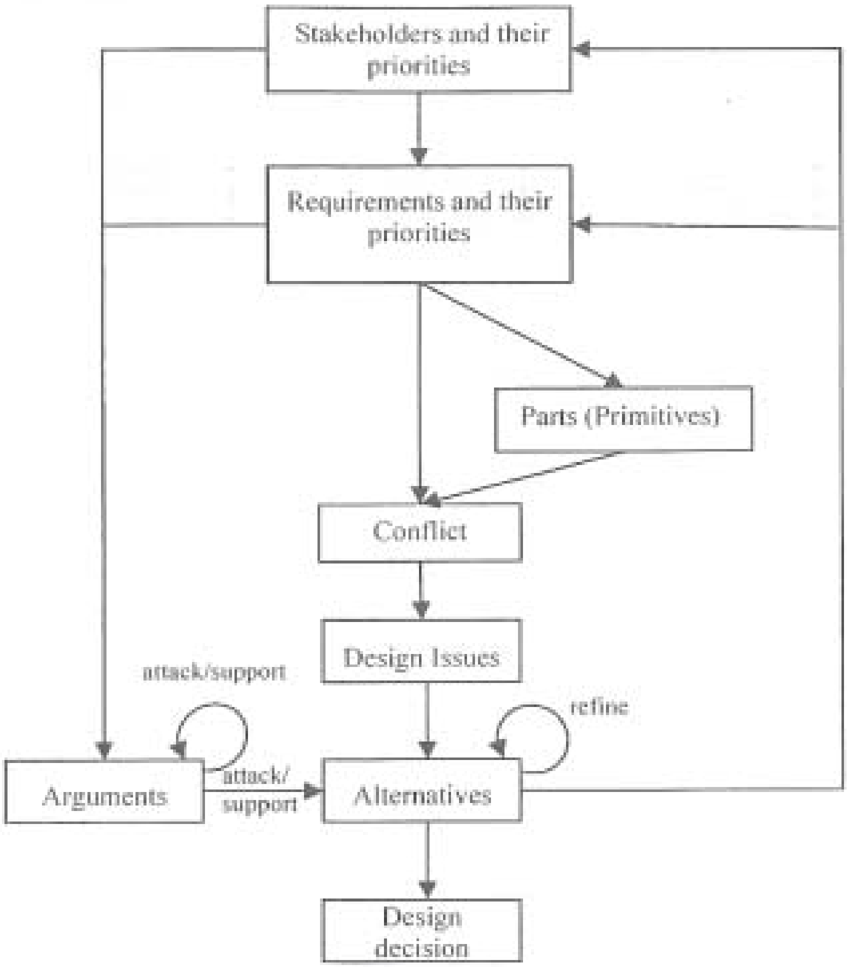


Figure 3.62 Framework for design argumentation (Source: Liu et al., 2006)

3.3.3 The third stage of CD: verification stage

In the third stage, major activities involve pre-empt production, manufacturing validation, marketing experiments, prototyping production, product evaluation, product refinement, and so on. This is the last stop along the product design process, but does play an important role to fill up the existing gaps between design and manufacturing stages. To facilitate the integration of different experts and enhance the efficiency of the iterative phases, prototypes are used as cost-efficient visual models. The use of virtual prototypes is especially in the early phases of product development, enable time- and cost-efficient decision making (Bullinger et al., 2007). The function served by the activity prototyping in this stage is significant to obtain the purposes described in last session. However, prototyping can be categorized into physical and digital two kinds (Bullinger et al., 2007).

Physical prototyping is well-known for the name rapid prototyping (RPT) for making it possible to produce physical artifact directly from CAD model without any tools. The most common technique today, like stereo-lithography

(STL), selective laser sintering (SLS), solid ground curing (SGC), and fused deposition modeling (FDM), are mainly used to produced design or geometrical prototypes. To accelerate the development processes, technical and functional prototypes are of great importance. Rapid tooling offers the possibility to build functional prototypes, and it is possible to build tools rapidly and inexpensively for prototypes parallel to the product development process.

However, physical prototypes are often time and cost intensive and thus need to be reduced to a minimum. By the combining of CAD technologies, rapid prototyping, virtual reality, and reverse engineering, prototypes can be produced faster and more cheaply than before. The digital demonstration allows early modification and optimization of the prototype. Furthermore, it leads to a cost-saving increase in the variety of prototypes. Also, faults concerning fabrication or the product itself can be detected in the early development phase and thus be eliminated without great expenditures.

An important component of digital prototyping is the digital mock-up (DMU), a purely digital test model of technical product. The objective of the DMU is the current availability of multiple views of product shape, function, and technological coherences. This forms the basis on which the modeling and simulation (testing) can be performed and communicated for an improved configuration of the design. The primary digital design model is called the virtual product. The idea is to test the prototype regarding design, function, and efficiency before producing the physical prototype. An enormous advantage of the DMU is the shortening of iteration cycles. Employing the DMU considerably reduce the time-to-market.

3.4 Collaborative design scope: Three scope levels and Contents

The last parameter coping with CD is the collaboration scope determined. Dealing with scope of CD, we put it into three segments cross-functional CD, cross-company CD, and cross-industry CD.

3.4.1 Three scope levels

In cross-functional CD, which is also the most common one faced, represents CD activities take place within the collaboration scope of one individual enterprise. CD may occur within the design collaboration team composed of designers only, or among multiple-discipline departments/ functions containing designers and non-designers. In general, key components of cross-functional CD may contain PM, Designer, project supporters (non-designers), System Administrator/coordinator in essence. In Figure 3.63, if we take the four segments as departments from planning to ramp-up production, then CD activities between department 1 and department 3 is one exampled of cross-functional CD.

In cross-company CD, we will deal with issues about CD activities occurred between two enterprises which may have relation with one another, or completely individual. To illustrate the scenario more easily in chapter four, we would follow the definition of actors of CD as introduced in section 3.2.1; following sessions will be discussed with the roles of Primary actor and Secondary actor (see section 3.2.1 & 3.2.5).

Liu et al. (2006) define the main characters of CD with scope of cross-company should contain groups of designers, manufacturers, suppliers, and customer representatives, which can be seen as the expended case of cross-functional CD in extent and scale. We should pay attention to the situation that the actors involved in this scope of CD are counted by "groups" rather by individual. Sections and scenarios discussion provided in chapter four would basically follow the view used here, which in two ways: relationship of vertical integration and horizontal integration.

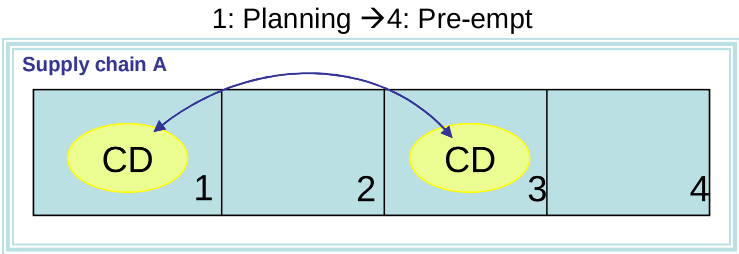
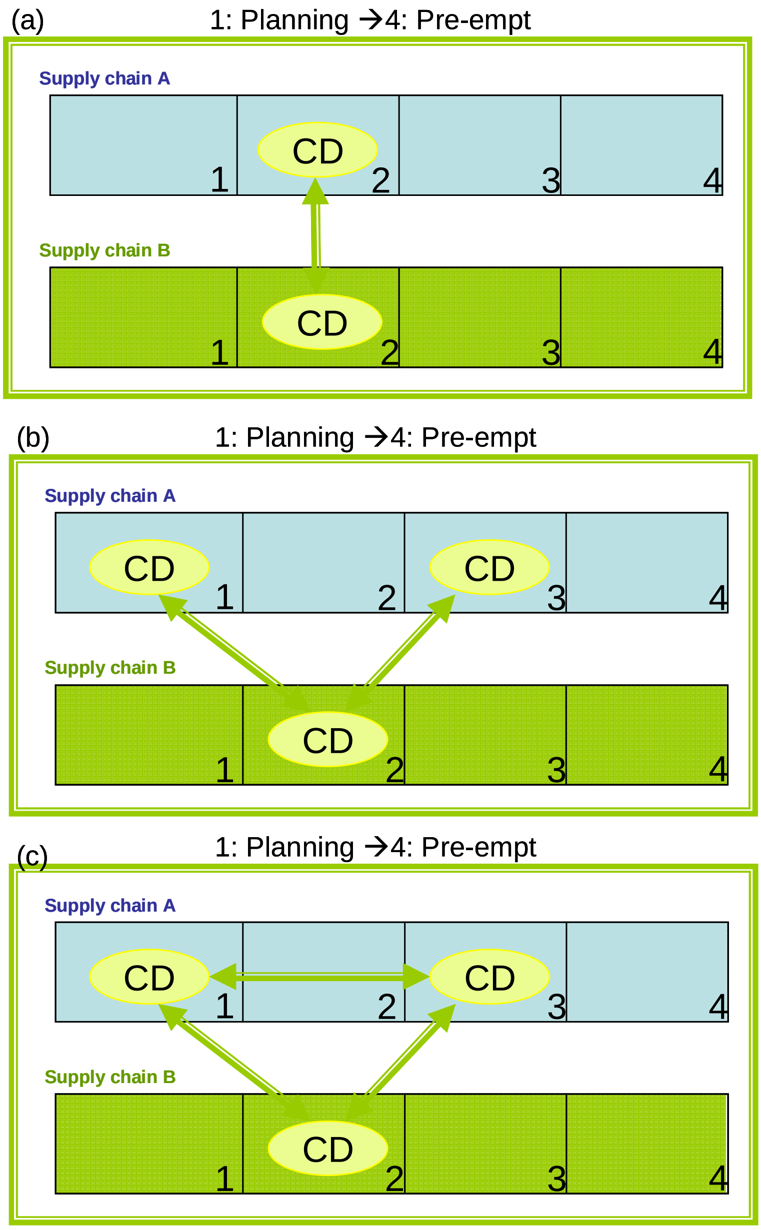


Figure 3.63 Example of cross-functional CD (Source: Original)



* Legend: number 1~4 stands for enterprise/ company of product planning to ramp-up production

Figure 3.64 Example of cross-company CD (Source: Original)

In view of vertical integration, we put it in two cases. One is that CD is carried out by two enterprises within the same supply chain (see also Figure 3.64); members of CD may be responsible for different tasks of the supply chain such as CD of two cooperators with the relationship of ODM and OBM. The other one is that CD is carried out by two enterprises of two unrelated supply chains. In this case, CD can be executed by members of the same function with different supply chain (see Figure 3.64-a.), or by members of complementary functions of two different chains (see Figure 3.64-b.). In more complicated situation, members of CD may include actors of same and complementary functions from both the same chain and different chains (see Figure 3.64-c.).

In view of horizontal integration, members of CD are not only belongs to different supply chains but also have different industry attributes which made them may not have complementary relationships. They come together may not for permanent cooperation but to capture the instant opportunities, which applying the concept of virtual enterprises. Take Figure 3.65 as an instance. CD occurred within two independent supply chains, and both are responsible for the ramp-up production stage, however, the one in chain C take the former 4-1 stage of design, while one in supply chain D participating in the latter 4-2 design stage. In this example, although they belongs to the same product function, cooperation of actors across supply chain still could happen for specializing on certain capability of the design stage.

1: Planning →4: Pre-empt

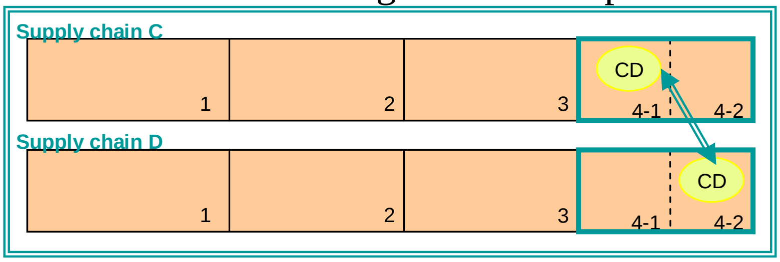


Figure 3.65 Example of cross-industry CD (Source: Original)

In cross-industry CD, we regard the model as the expended horizontal integration CD of cross-company. Take the biblical product City Storm, which is the CD achievement of Giant and DEM, as an example. Giant and DEM belong to different industry in the very beginning, but they cooperate with each other in the end under the CD environment to focus on their own core competence and end up very well (CD example of design and product assembly).

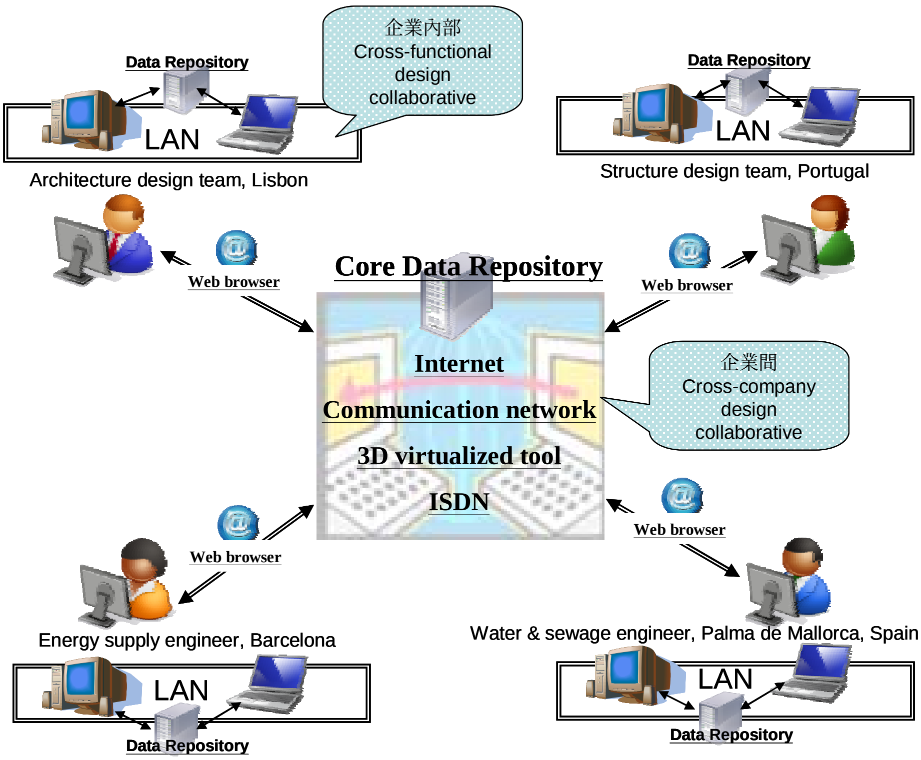
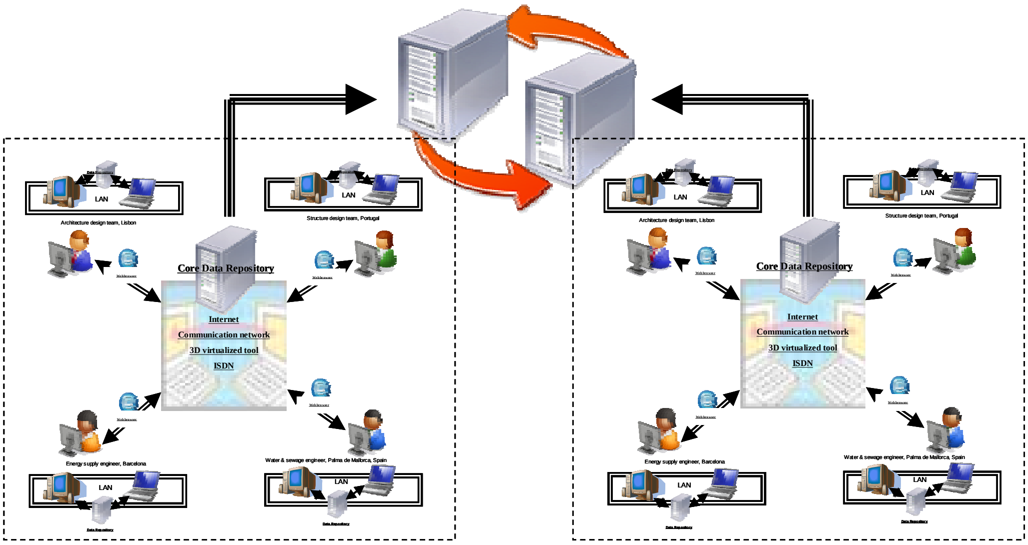


Figure 3.66 Communication reference framework (Source: Original)

We propose a communication reference framework to indicate the elementary components of a fundamental CD (Figure 3.66). In this framework, we can tell that there are four CD teams in this network, which are located around the world, and each of them is responsible for specific task of the CD project, which are architecture design structure design, energy supply engineering, and water& sewage engineering. To configure CD activity, each of the CD members should be equipped with technology and infrastructures mentioned in section 3.2.5 including data repository, web browser, process diagram tool, interface mapping/integration tool, and Dynamic User Interface (UI) Generation (application tools). The scenario described above is an example of CD taken with cross-company scope.

If we put the collaboration into a smaller scale in view of each CD team such as architecture design team in Lisbon, we can find out that collaboration may still exist within the process of architecture design tasks taken inside the corporate, which is the instance of CD with cross-functional scope. In this case, all the required elements forming a CD such as including data repository, process diagram tool, and so on, should still be equipped to enable proceeding of collaboration.



* Third-Party Database/ Repository

Figure 3.67 Communication reference framework: cross-industry collaboration (Source: Original)

In Figure 3.67, we illustrate the scenario of CD taken in place with cross-industry scope. Cross-industry collaboration can be regarded as the expended case of cross-company collaboration. The most significant difference between these two kinds of CD is the interfaces or collaboration platforms on which they communicate with each other. Reference model of cross-industry collaboration may like Figure 3.67, participants of CD which come from two different supply chain of diverse industry characteristics may collaborate with each other under the CD environment supporting by the third-party repository, which stands for certified criteria for communication among cooperators.

3.4.2 Issues of design process and CD scope

Dealing with configuring design processes, another affecting element is the collaboration scope involved. Following the framework of applying process level and coloration scope proposed by Panchal et al. (2006), we can have better understanding of different process configuration levels coping with corresponding CD scope applied. For example, as the scope increases from involving a single designer to teams and multiple organizations, the relevant detail of the design process changes from involving interactions among design variables to inter-organizational interactions (Panchal et al., 2006). Therefore, in different level and scope of CD, there would be corresponding design processes configured for cooperation.

However, we still have to put in minds that scopes of CD can take place concurrently within the same CD project among different levels of division of labor. Take above scenario for instance. The four individual distributed design teams, responsible for architecture, structure, infrastructure and energy supply engineering accordingly around the world, is the first tier of this CD project and fit in with the right-up corner of multiple organization with inter-organization interactions to the managerial level of design process configuring. However, there may exist the second or the third tier of CD projects involved inside each four teams. In these cases, the level of design process would top down to the designer level eventually dealing with interactions of design variables. (See Figure 3.68)

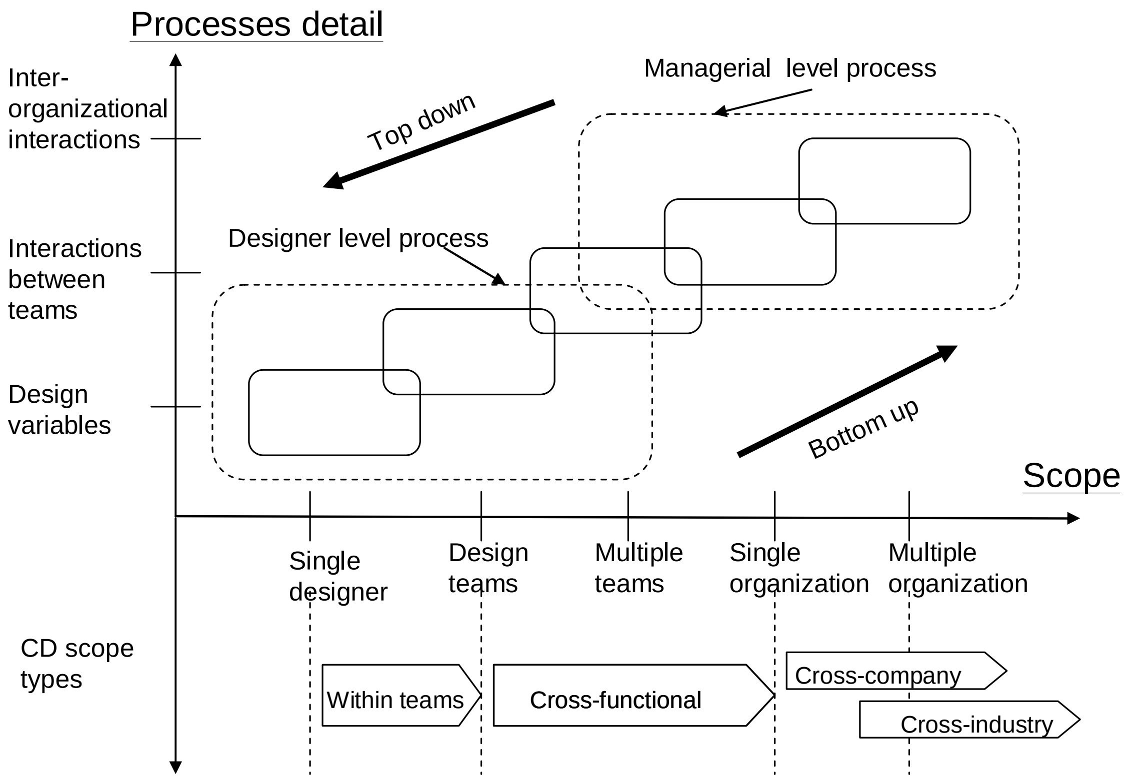


Figure 3.68 Processes represented at various levels of details (Source: adapted from Panchal et al., 2006)

CHAPTER 4 Collaborative Design Reference Models

Chapter four will present three CD models in view of collaboration scope from cross-functional within a corporation to CD in the manner of multiple organizations, including case of composing supply chain (SC) entities within the same industry or the case beyond SC with cooperating partnership built upon diverse industries.

Section 4.1 would elaborate situation of corresponding process and information models at the scope of cross-functional design collaboration. Section 4.2 will focus on the scope of cross-company design collaboration, and section 4.3 centers on the scope of cross-industry design collaboration.

4.1 Cross-functional design collaboration

4.1.1 Definition

Cross-functional design collaboration takes place when cooperation of multiple-disciplines teams or departments for New Product Design (NPD) or Product Design (PD) is necessary, and is limited to the scope within corporation for information sharing, communication activities, and process interacting, an so on. The cross-functional CD dimension is as shown in Figure 4.1.

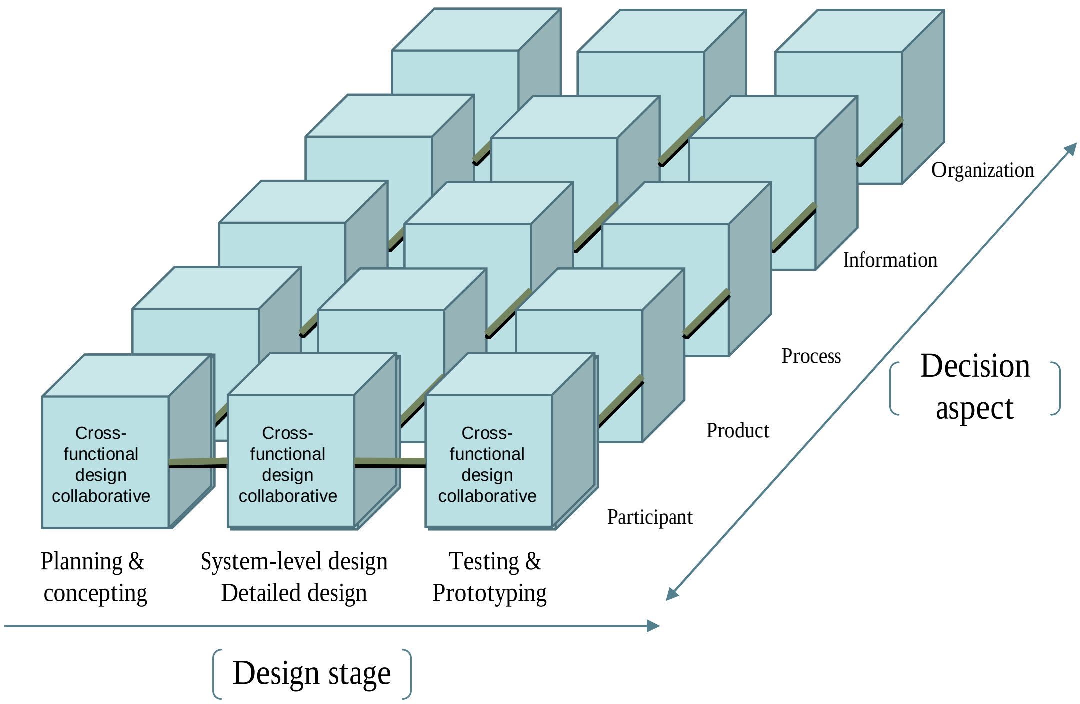


Figure 4.1 Cross-functional design collaboration architecture (Source: original)

4.1.2 Discussions

We elaborate the composing elements and situation occurred in this dimension with the proposed analysis framework illustrated in chapter three already.

4.1.2.1 Decision aspects

In the participant aspect, we can take all these cases in view of the features of actors involved or in the view of the participant taxonomy matrix we concluded in section 3.2.1. In the former manner, it is easily being realized that two cases are involved, namely designer-to-designer, and designer-to-NONdesigner. For general NPD, CD should be carried out with integration of multiple-disciplines departments such as product appearance design, design of materials applied, or even the modules of sub-systems. For example, though it seems there is no interaction between the design of transmission system and the air-conditioning system, and it takes for granted that these can be preceded by two unrelated team concurrently without collaboration. However, things turn out to be totally opposite, and here comes the case of cross-functional CD.

The designer-to-NONdesigner situation takes place to the extent above design tasks itself, but dealing with PD or the development project level. Therefore, participants should contain product design and development department, R&D, departments concerning product technology such as manufacturing and IT, project management teams, and marketing coping with customer needs and demand. In view of participant taxonomy matrix, the four types of players can be adapted to fit the characteristics of different departments. For instance, major player can refer to critical design team members, and compatible player is for flexible down-stream manufacturing lines.

In product aspect, all of the five types can take place, but general and routine NPD with derivative of existing platform or incremental improvement are the most common. Fundamentally new products, new product platforms, or even new product generation/ upgrade may cost more effort, and probably would need more cross-organization collaboration.

In the organization aspect, CD take place within the corporation is usually executed by project organizations, and adoption of the appropriate type of organization is in accordance to the corporate culture or the evaluation system of the company. The classification of project types is well-defined in section 3.2.4 of the four main types proposed by Ulrich and Eppinger (2008).

In information aspect, mono-participant CD can be take as cooperating relationship of two individual departments such as R&D and marketing, or architectural design and engineering design. In addition, multiple-tier mono-participant CD can occur in the case of three collaboration teams take the responsibility from mechanics testing design, product architectural design, to manufacturing parts in sequences. Multiple-participant CD, on the other hand, illustrates the situation such as two collaborative teams cover the same design tasks such as mechanics design or product structure design, and both respond to the same CD project team. CD in Network can be applied to explain the phenomena of complex CD with multiple teams or members of the same design task layer cooperate with each other to carry out the same goal or certain functions of the product. For instances, we may have three correlated cells to carried out different sub-function design such as material applied testing team, geometrics design team, and interfaces design team under the architecture design of the collaborative product design and development project.

4.1.2.2 Design stages aspect

Cross-functional CD can take place cross-stages at the same time for different departments and teams would be involved at different design stage, and this is as shown in Figure 4.2. In this case, the cross-functional CD collaboration happens when there is a need for the third stage of pre-empt production to inform the design team in the first stage of the production limits or specification etc.

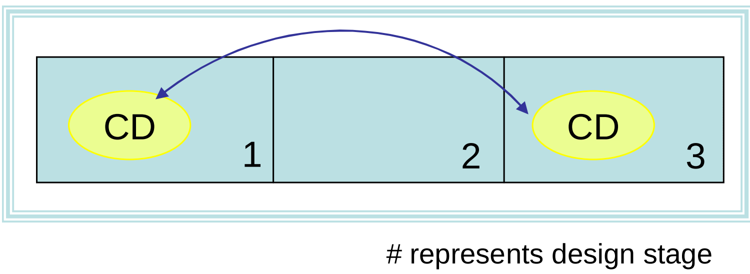


Figure 4.2 example of Cross-functional CD within design stages (Source: original)

However, Cross-functional CD can also take place within each design stage with various cooperators representatives. For example, in the first stage of planning and concepting, product design planning team and concepting team may have collaboration with one another to make sure the time table is fit. Other Cross-functional CD examples are like CD within a design team or project with diverse sub-team or sub-project, and CD within multiple-sites participants for performing different functions by each site within the same corporation.

4.2 Cross-company design collaboration

4.2.1 Definition

Cross-company CD can be taken as the extended one of cross-functional as we mentioned in last section, but not dealing with cases of CD cross industries. Cross-company CD still focuses on the information sharing mechanism, communication and interaction within activities. This kind of CD is the most common one we can see in empirical practices. CD of OEM, ODM with OBM is one kind of the instances. The cross-company CD dimension is as shown in Figure 4.3.

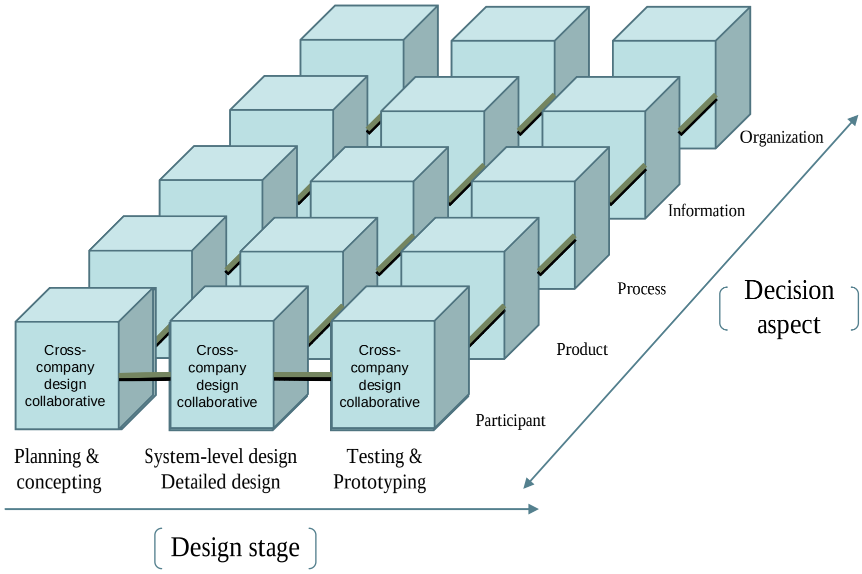


Figure 4.3 Cross-company design collaboration (Source: original)

4.2.2 Discussions

4.2.2.1 Decision aspects

In participant aspect, the demarcation of designer and non-designer still works well. Two main types of CD participant are designer-to-designer and designer-to-NONdesigner. In the former case, the involving participants can be classified by the CD participant classification matrix described in chapter three as shown in session 3.2.1. For instances, if the case of our target CD product is dominant technology required then niche player of possessing particular techniques would be included in. In this case, the primary actors, in the scenario is the role we play, may have less power to argue on the market for the niche player possessing things we are eager to order. Examples are like the up-streamers of DRAM industry, or the transmission corporation such as Shimano, whose products and technology is the most critical components of bicycle, to all ODM, even OBM such as Giant.

In product aspect, CD with complex and multiple-disciplines team member required is more easily to be cross-company one. Fundamentally new products and new product platforms are those cases. Considering product structure and product extensiveness, product of cross-company CD falls on to the product structure of integrate and not limited to the product extensiveness, which include specialized and general usage. For example, even like product with superior complex architecture as Airplanes, jet engine is not necessarily has to be implemented by CD. They can consider continue using existing technology or extended by present platform but with costing cross-company CD. However, even like software such as ERP package software or VISTA belongs to general products may still have the demand for CD to fit the high-risk properties. (See Figure. 3-6)

In organization aspect, we conclude for three main types of CD organization to choose from, namely project, mediatory, and functional as introduced in section 3.2.4.

With the topic of cross-company CD, we can take it into two directions in information aspect. One is in view of horizontal integration CD, and the other is vertical integration CD. The former one represents roles involved perform the same or complementary functions within the same tier. For example, CD carried out on the subject of system-level design, and three participants involved belongs to different individuals. In this case, the three participants can be three corporations around the world take the responsiveness of product structure design, architecture design, and system integration assessment accordingly. All of them perform the tasks of the system-level design jobs at the same design stage and tier (not up-stream and down-stream relationship) but collaborate with each other.

The latter one can apply the CD Information framework introduced in section 3.2.5., namely Mono-participant CD, Multiple-participant CD, and CD in Network. In mono-participant CD, simple cases can be taken into consideration. For instance, CD such as derivative of existing product platforms can be carried our within corporation or co-work with suppliers. If it is the latter case, then we have the Mono-participant single-tier cross-company CD. If the dyadic relation changes into two- or three-tier, then we will have a Mono-participant multiple-tier cross-company CD. If the number of up-streamers multiple, then we are in the case of Convergent multiple-participant cross-company CD. If it is combination of both multiple up-streamer and down-streamers, then here is the case of Network cross-company CD.

4.2.2.2 Design stages aspect

Cross-company CD can take place at the same design stage or cross different stages. Here we have three examples involving two companies to illustrate situations described above. In the case of horizontal CD at the same design stage (Figure 4.4), we have two cases. One is two participants belongs to two companies carry out design tasks of the same design stage (here is stage 2) concurrently and collaboratively. Figure 4.4-a is the general case that does not consider task sequence, while Figure 4.4-a-1 is the one of sequence relation of the two CD tasks, which means though they are collaborate with each other at the same stage, but the task of company B and A have interdependence relation.

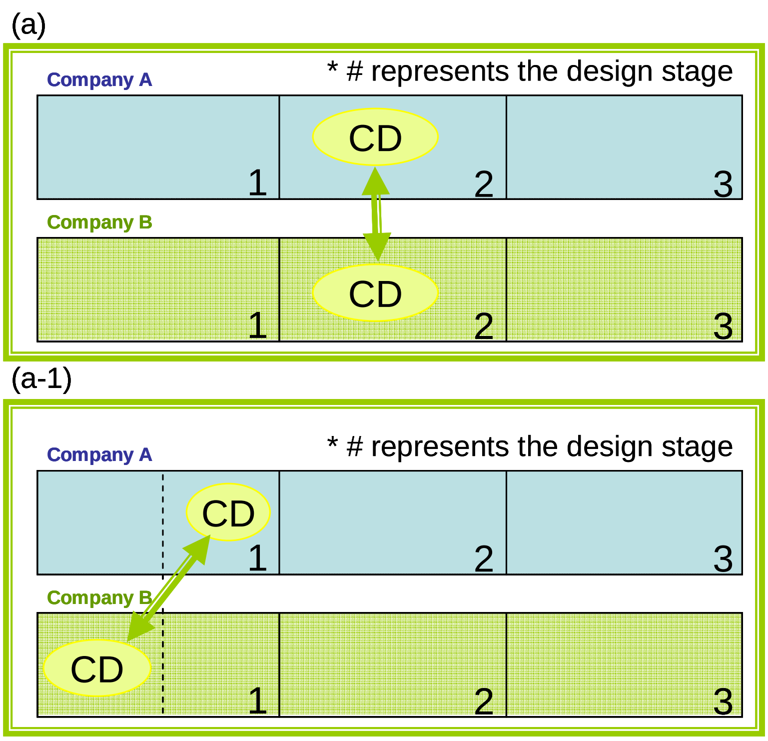


Figure 4.4 example of cross-company CD of the same stage (Source: original)

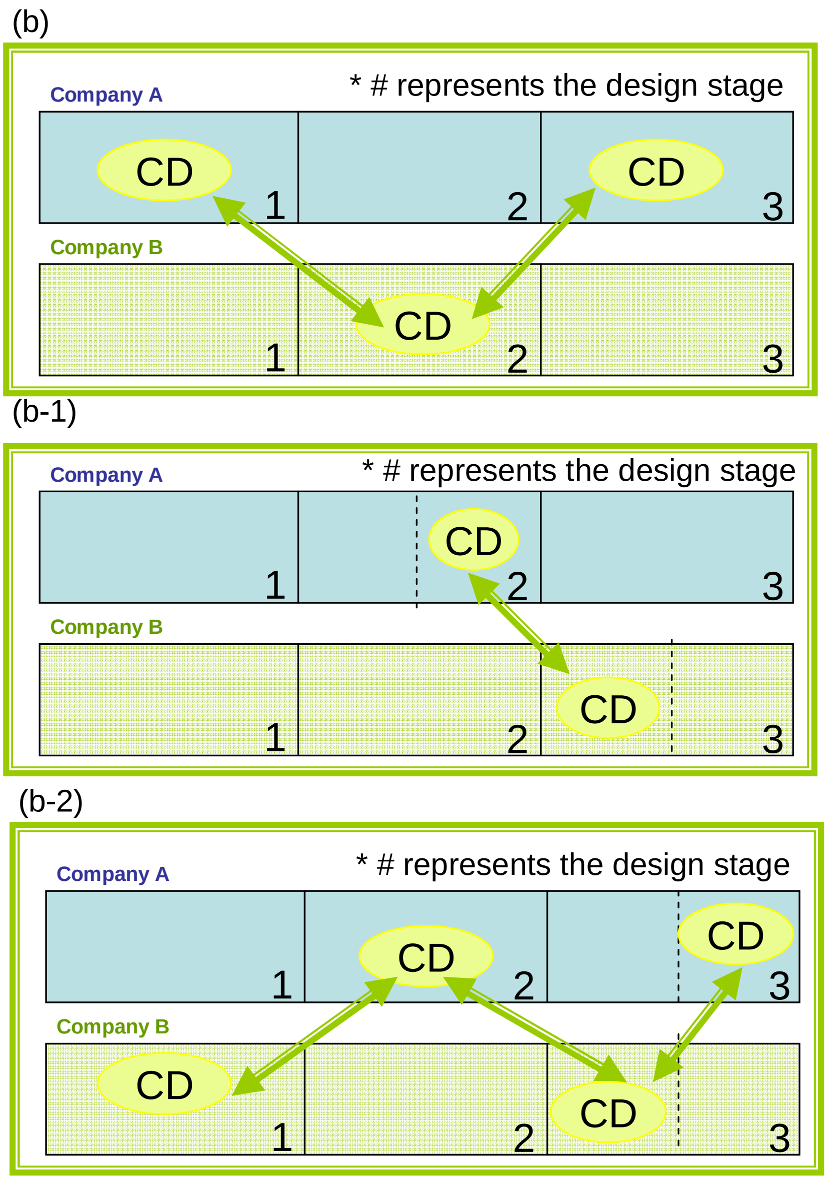


Figure 4.5 example of cross-company CD cross different stage (Source: original)

If it is the case of CD cross different stages, we conclude three scenarios as shown in Figure 4.5. Figure 4.5-b shows two participants involved to carry out three CD deign tasks at three stages individually. From stage one to three, the whole design process is complete the two and there is sequence relation among the tasks and stages. Participant of company A is responsible for tasks of stage one and third, here what we need to show attention is the two tasks may be carried out by two different teams but only of the same corporation. Therefore, although there are only two companies involved, three participants are the actual number to count, and there is no direct connection between stage one and three but the three collaborate with each other following the design stage sequence. Figure 4.5-b-1 shows the special case of cross two stages but have a sequence relation. Figure 4.5-b-2 combines the concept of Figure 4.5-b and Figure 4.5-b-1 showing that no interaction or information sharing among participants within the same company, but each two of them do have collaboration.

(c)

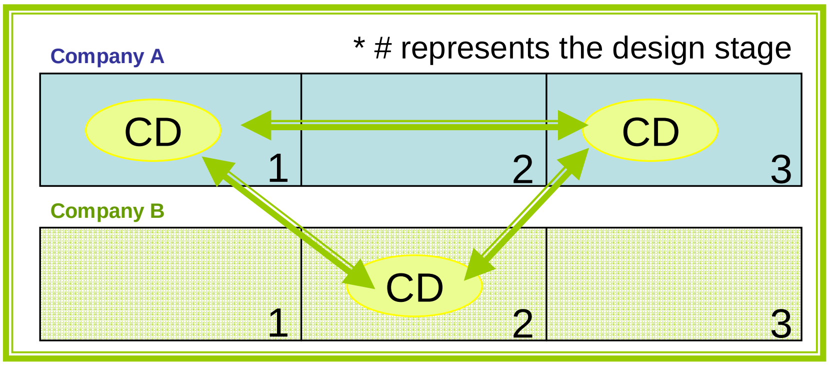


Figure 4.6 example of cross-company CD (Source: original)

Example in Figure 4 shows us the situation of considering fully communication of all participants involved even of the same corporation. Even CD is carried out through the whole product design and development stages by three individual design teams of two companies, concerning of information security and information sharing necessity, it is possible to interchanges information by two teams carrying tasks crossing different design stages within the same corporation. For example, company A may only outsource the system-level or detail-level design part to Company B, and have the concepting and prototyping part inside company A. To make sure the materials applied in concepting stage can work out well for prototyping stage, maybe it is reasonable to sharing information and communicates between design team of the first and third stage.

4.2.3 Case Study

We follow the CD environment provided by Chung and Lee (2002), and will give a brief introduction of the injection molding case. In session 4.2.3.2, the delivery of OA application will show how we can apply the framework under the cross-company CD scenario.

4.2.3.1 Scenario Introduction

Scenario proposed as shown in Figure 4.7 gives us the information used for OA as following:

(1) The participants include three parts: Customer Company, Injection

Company, and mold maker. The mold maker knowing as mold company consisting of four divisions, which are

(2) Product is basically the plastic parts ordered by the customer company, which would outsource the order to a mold company.

(3) Process includes the fundamental workflow dealing with order fulfillment and the part of collaborate with one another. Basic collaboration process follows the life-cycle: injection company get the order from Customer Company with product drawings and specifications. Considering cost and degree of difficulty, chosen Mold Company will get the outsourcing order with needed information such as specifications. Then inside the mold company, cross-functional CD takes place. Two CD scenarios occur under the situation of this case, design validity evaluation scenario and appropriate outsourcing company selection scenario. Both take place within Injection Company and mold company.

(4) Organization: three parts involved may be more like mediatory organization for Mold Company is the actual manufacturer and injection company play the mediatory one. However, the whole CD process is executed through project management approach.

(5) Information includes two parts, one is static one indicating the data, transformation, information flow, and knowledge involved. The other is dynamic one, suggesting information framework or mechanism of CD.

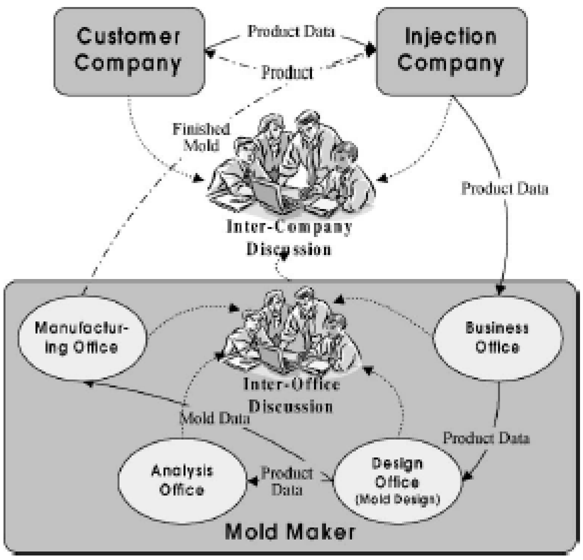


Figure 4.7 Workflow in injection mold production process (Source: Chung and Lee, 2002)

4.2.3.2 Application of OA

In the case of cross-company CD, we will follow the other two dimensions of OA, decision aspect and design stage aspect, to demonstrate how this framework proceeds. In the following sessions, we will depict the corresponding decision aspects to each design stage from incubation to verification stage under the scenario proposed by Chung and Lee (2002).

- Incubation stage
  - *Participants & Product*: in the first stage, participants involved are only customer company, Injection Company, and mold company. Partnership selection process may take place within each of the three ones. However, in this case, the product in prone to be product platform improved or upgrade, even would be routine new product (plastic parts used here in view of costumer company) development. Therefore, Injection Company here may play the compatible player to Customer Company, and the mold company may be the minor player injection one or even customer one if customer does involve in the selection process or have the authority.
  - *Process*: in product planning and concepting, although Mold Company does involved in the activity, real characters are only customer and injection for product concept development and specification definition. If it is the case of OEM, then injection company does gain the specific design drawing and specification information from Costumer Company. If it is ODM or OBM, the role of injection company would play more important the role as specification/code definer. In the latter case, communication between customer and Injection Company would be more frequent and intense.
  - *Organization*: in this stage, Mold Company does not play significant role yet in the CD activity. Two main participants, customer and Injection Company cooperate more like a functional organization, each play their duty well without dominating another one.
  - *Information*: as mentioned by Chung and Lee (2002), information exchange platform is formed by XML, and the CD system architecture follow web-based mechanism like Figure 4.13. In incubation stage, the information framework would follow Figure 3.56 with the situation of multiple-tier (here is two-tier) mono-participants CD.
- Proceeding stage
  - *Participants*: during the system-level and detail-level design, mold company play an important role in this case. Since the selection process has been executed before this stage, which is not a issue of this paper, here we have three main characters in this stage, namely customer, injection, and Mold Company, and the latter two have closer interaction. In view of customer, Mold Company may be the minor player which

only does the manufacturing part, while it would become a compatible player to Injection Company on the other hand.

- *Product*: products in this stage would be plastic parts manufactured by Mold Company with the specification and design drawing provided by Customer Company, but may be verified or trimmed by Injection Company who may have the authority given by Customer Company.
- *Process*: through CORBA and platform formed by XML, customer and Injection Company may cooperate with each other to analyze and verify the specification and design drawing delivered by Customer Company considering its manufacturability and difficulties.
- *Organization*: organization type of this case under the stage would fall on to the "mediatory organization" type, and the injection company plays the mediatory role.
- *Information*: In static part, technologies used include XML to share information among design participants and application programs, RAMDES (RApid Mold Design Expert System) for mold design tool, PCIA (Parameter Connectivity Information Administrator) for evaluating the validities of parameters used in different dimensions and companies, client module (see Figure 4.8) and sever module (see Figure 4.9). In addition, the information system does provide a display of design information served the similar functions as PDM or PLM system (see Figure 4.10). In the dynamic part, the CD system of this case follows Figure 4.11 and Figure 4.12, containing the four qualified factors mentioned in section 2.5.2, which are data repository supported by XML, 3D virtualized tool for product design among various clients, connecting system by PCIA, and ASP sever to verify CAD system customized for mold design. Besides, the communication mechanism would be as shown in Figure 4.13.

With the analysis framework of OA for CD, situation faced here will fall on to the case of "CD in Network" as shown in Figure 3.58. Injection Company plays the middle layer character facing both the layer of supplier and buyer. Although Chung and Lee (2002) does not deliver the detailed information regarding corresponding authority of each participant using the information sharing platform, the information gather from customer and mold company should be filtered by injection company for its own good, or for other confidential issues. Information exchange media (web-based platform or IT system) would follow the mechanism we provide as shown in Figure 3.58.

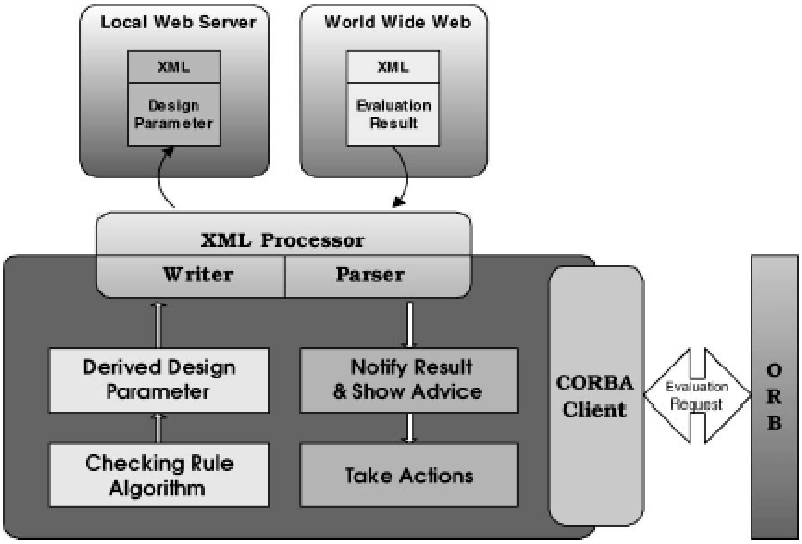


Figure 4.8 Client architecture (Source: Chung and Lee, 2002)

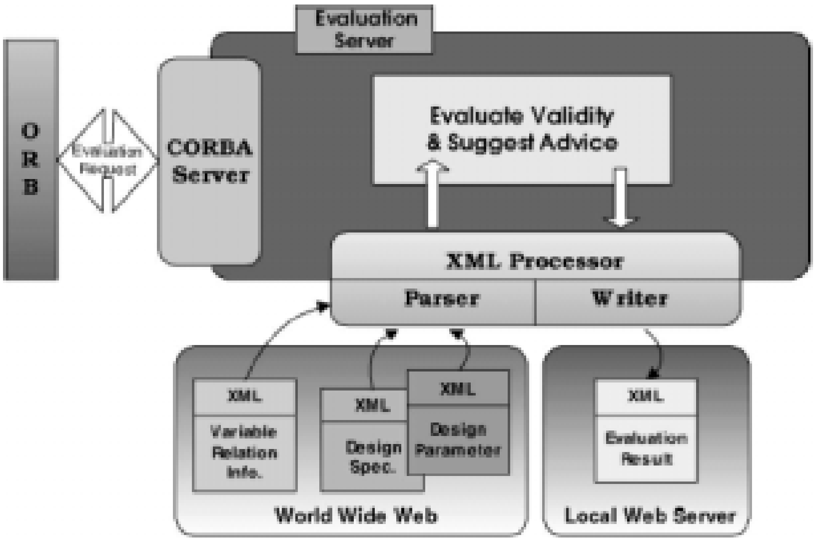


Figure 4.9 Server architecture (Source: Chung and Lee, 2002)

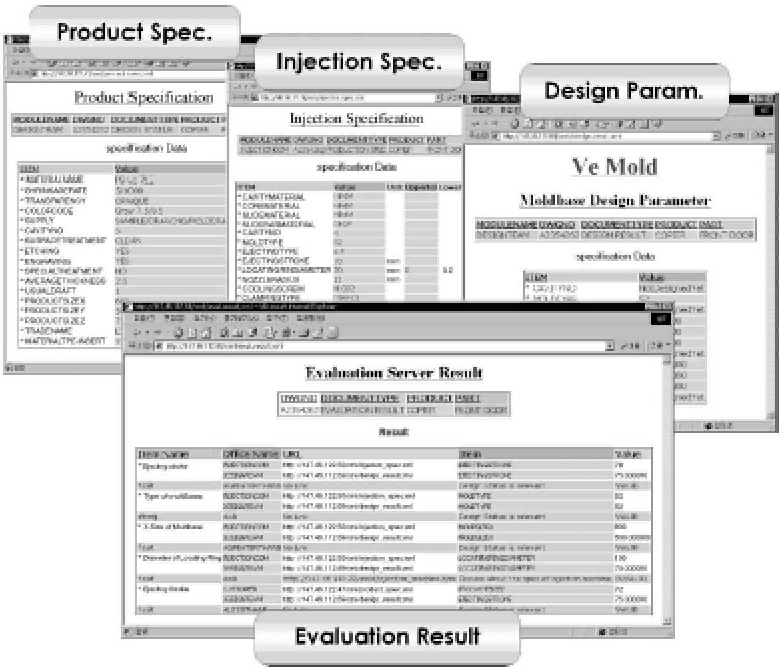


Figure 4.10 Display of design information (Source: Chung and Lee, 2002)

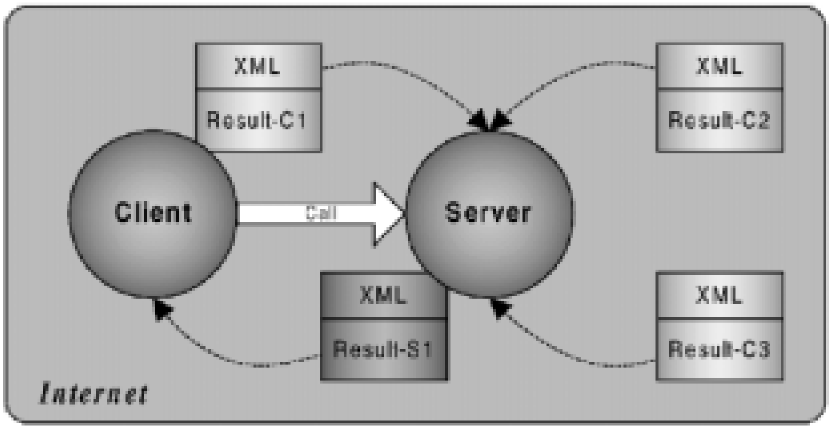


Figure 4.11 Basic collaborative design system model (Source: Chung and Lee, 2002)

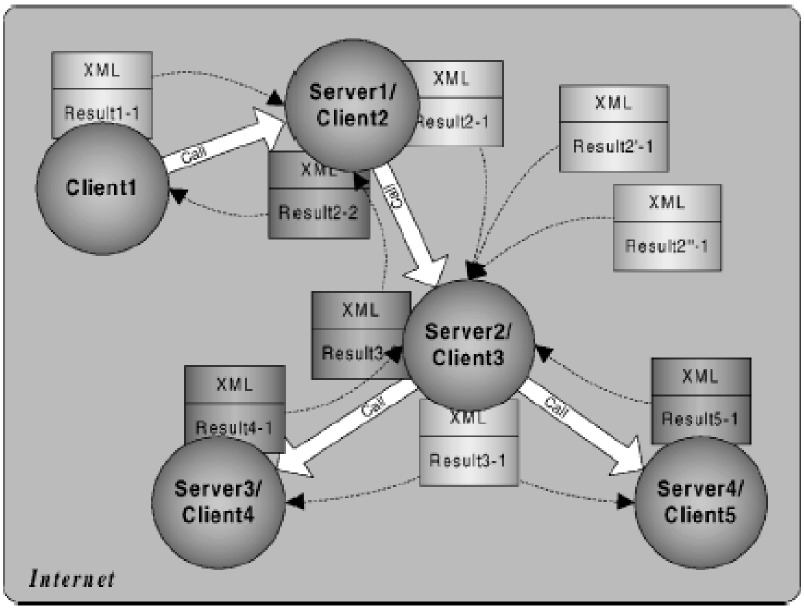


Figure 4.12 Extended system architecture (Source: Chung and Lee, 2002)

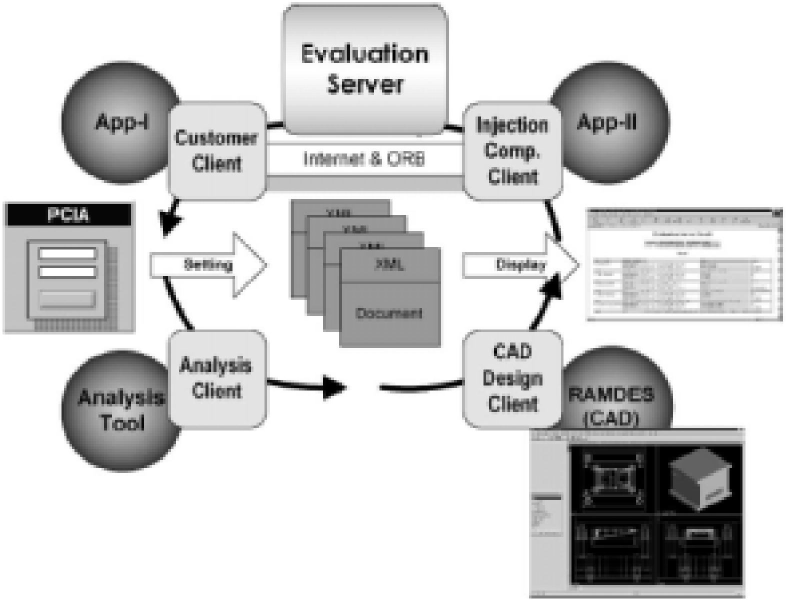


Figure 4.13 System architecture for injection mold productions (Source: Chung and Lee, 2002)

- Verification stage
  - *Participants:* in this case, participants of verification stage are more like cross-functional CD. Participants involved are basically divisions of Mold Company, and may include some analysis and design divisions of Injection Company. Manufacturing and business office of Mold Company may be the primary actors tackling with Injection Company. And Design and analysis office of Mold Company would support them

making mold production successfully. However, from another viewpoint, manufacturing and business offices play the secondary roles to support design and analysis divisions to fulfill the order from Injection Company.

- *Product:* product here is basically the improved design objects or routine product development ones that should manufactured by manufacturing office of Mold Company and deliver it on-time.
- *Process:* divisions of Mold Company have to cooperate with each other to fulfill the request quantity and specification of the order from Injection Company. Concurrent engineering (CE) may be applied and communication mechanism would be delivered in following sessions.
- *Organization:* organization applied here may be general project management organization as concept proposed by Hayes (1988) and taxonomies proposed by Ulrich and Eppinger (2008), which are projects organization, functional organization, and matrix organization.
- *Information*: divisions within Mold Company would able to get on the same IT system or intranet platform to interchange information within different departments. Data repository such as product issues tracking system or PDM system and connecting system like intranet communication platform may provide a channel for information sharing and update among divisions. Information framework may follow Figure 3.57. Although CE does implement in the CD, certain sequential activities still take place within the CD activities. Therefore, within different authority given, analysis office and manufacturing office may only get the partial authority on the IT platform, for instance.

4.3 Cross-industry design collaboration

4.3.1 Definition

Cross-industry CD can be taken as the special case of cross-company CD illustrated in last section. Most of the cases focus on pursuing function-oriented collaboration partnership and is more like VE; once the CD project is accomplished the organization would be released. Usually, the primary actor of cross-industry CD tries to collaborate with the secondary for particular capability for whom can provide. The cross-company CD dimension is as shown in Figure 4.14.

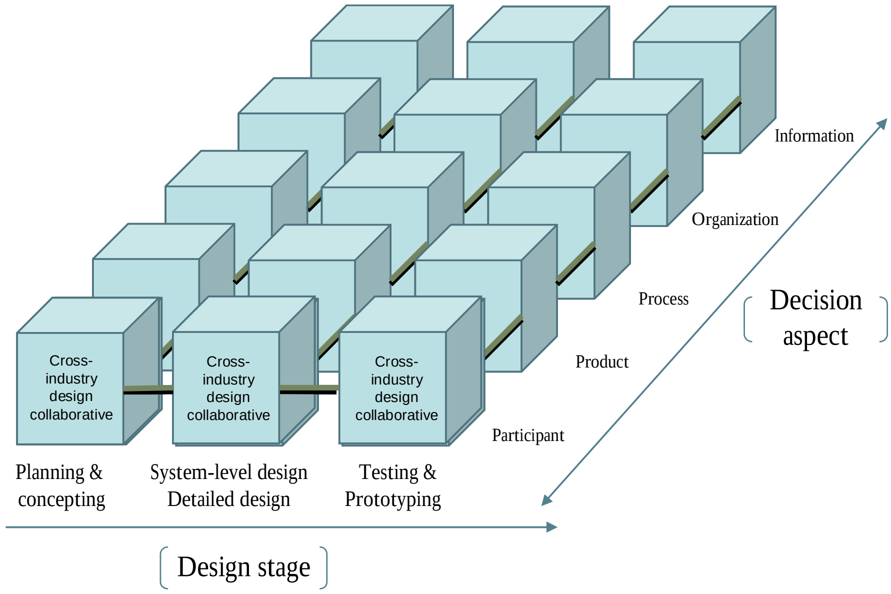


Figure 4.14 Cross-industry design collaboration (Source: original)

4.3.2 Discussions

4.3.2.1 Decision aspects

Following the five CD decision aspects, contents applied for cross-industry CD are typically the same as illustrate in section 4.2.2. What we want to emphasize is the individual functions each participant can provide. Although we focus on the collaboration of purely the domain of design, literally speaking, we can classify the function of each kind of design tasks play, and compress it into three main parts, namely pure design segment, manufacturing segment, and marketing segment. The segment of pure design we defined here is about the product concepting, physical product structure design, etc, which are more close to "design" itself. The portion of feasibility assessment design, DFX (design for X), and so on belongs to the manufacturing segment. The last one is the marketing segment representing the portion of CD planning design, designs for project execution, or for the supporting of administration system.

Following definition described above, the participant combination pool of cross-industry CD is as shown in Figure 4.15 and 4.16. The classification criteria are set to be how many tier is involved, and what the number of participant at each tier is.

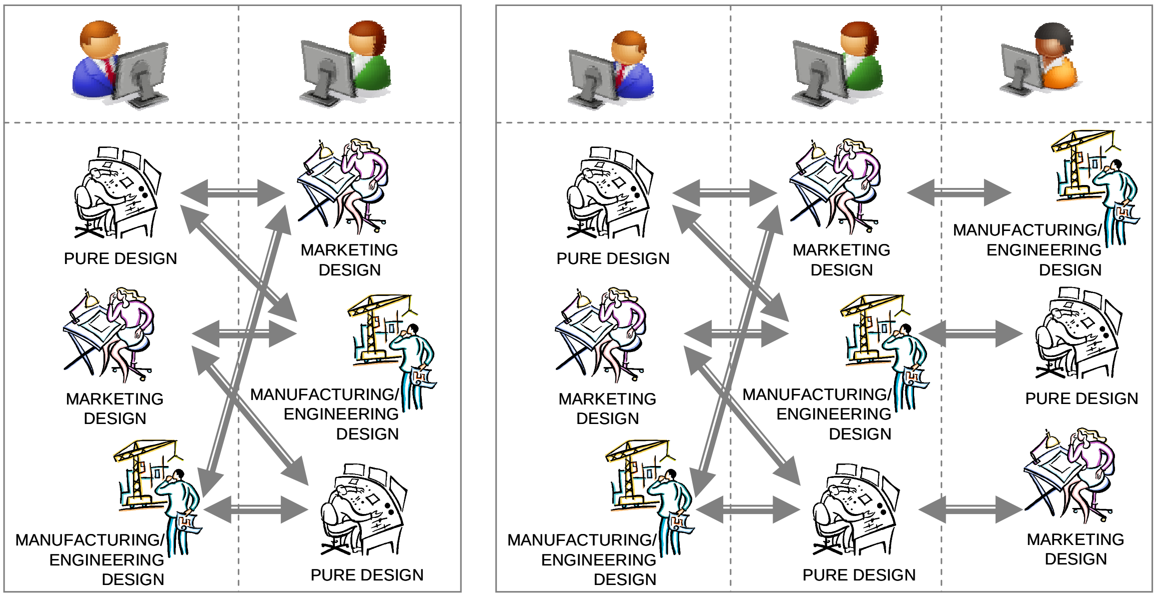


Figure 4.15 example of participant pooling with single-tier cross-industry CD (Source: original)

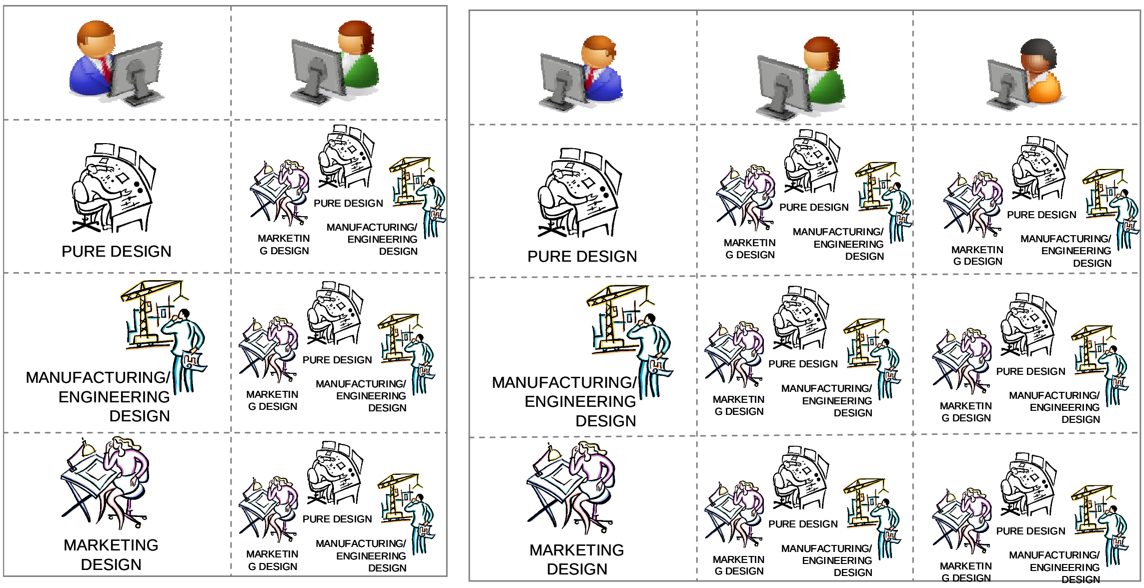


Figure 4.16 example of participant pooling with multiple-tier cross-industry CD (Source: original)

Typical empirical instance of cross-industry CD is the case of product named City storm by Giant, the top three bicycle brand in USA, and DEM inc., the design corporation. Giant used to be OEM for brand name bicycle Co., while now become ODM even OBM of the industry. However, case of NPD as City storm focusing on its characteristics of luxury goods, fine art but function-oriented product make Giant have to collaborate with professional design Co. such as DEM inc. cross two industries of bicycle and design.

The information frameworks and contents are almost the same as described in section 4.2.2, but only have to emphasize the information sharing platform or repository have to be run by the third-party or with the guaranteed of them for the sake of reliability and justice, which is the most significant difference between cross-company and cross-industry CD.

4.3.2.2 Design stages aspect

Cross-industry CD can also take place cross different design stages. In Figure 4.17 we show the examples of CD across two different industries at various stages, and the concept is the same with cross-company CD but have the participants coming from different industries.

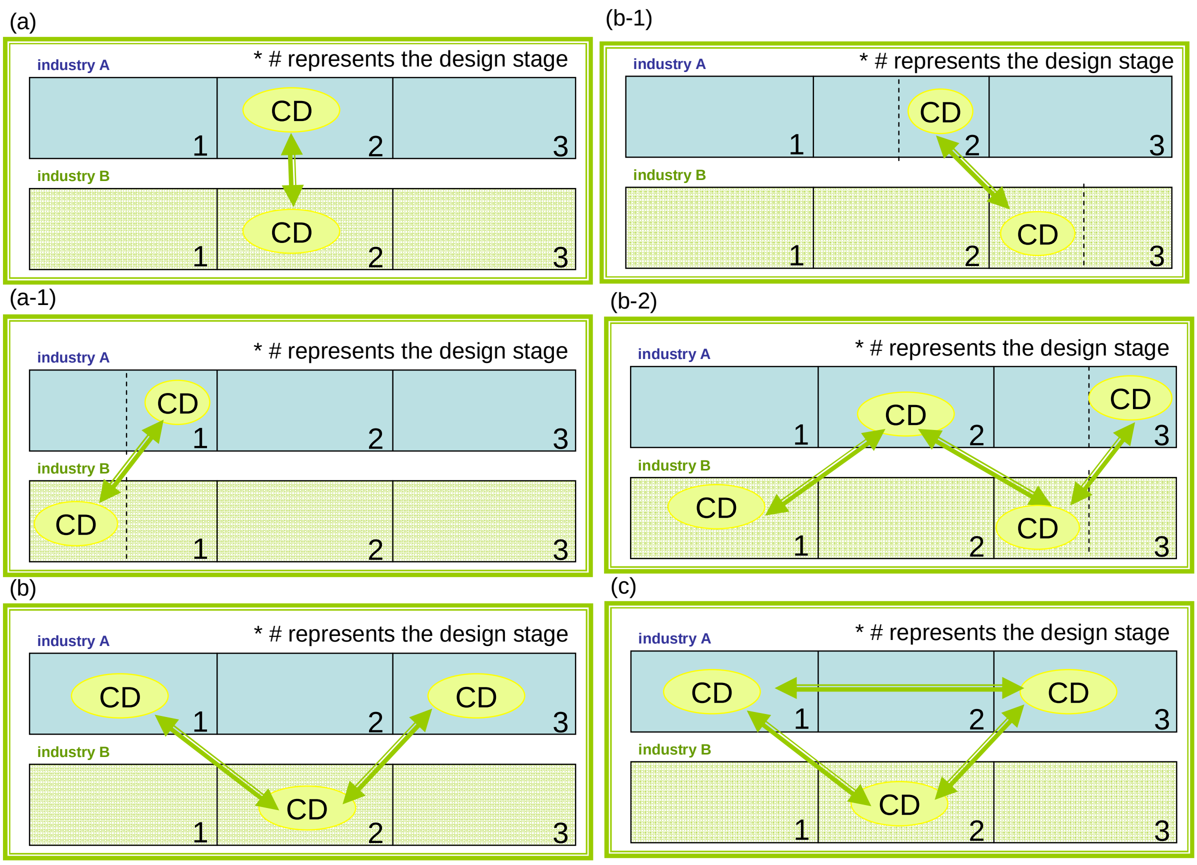


Figure 4.17 examples of CD types in view of design stages (Source: original)

CHAPTER 5 Conclusions and Future Research

This chapter reviews the work we proposed in this thesis, draws the main conclusions, and then describes the opportunities of future research.

5.1 Conclusions

We introduce the CD OA for issues of collaborative design, providing a new perspective in analyzing CD and dimensions needed to be considered when initiating a CD project. This framework helps decision-maker in dealing with implementing a CD project or activity. The CD cube framework also serves as a guideline map for software system developer or people involved in the design collaboration to figure out their own functions and current status of the group. Three dimensions included in the framework are:

- *Decision aspect*: five aspects include Participant, Product, Process, Organization, and Information.
- *Design stage*: basically divided as three parts
  - Stage one: Planning and concepting
  - Stage two: System-level design and detail design
  - Stage three: Testing and prototyping
- *Collaboration scope*: includes three types of situations
  - Cross-functional CD
  - Cross-company CD
  - Cross-industry CD

In chapter four, we provide a discussion of CD reference models by three different scopes to demonstrate how to use the framework in developing design collaboration activities and to specify the details of three various types of CD contents according to the collaboration applying scope.

However, there are still many technical and managerial issues and structural improvements left to be continued and proceed.

5.2 Future Research

Despite the collective contribution of the demonstration of the CD cube framework, there are many limitations of this framework. For examples, empirical practices and studies should provide more execution topics and managerial problems to be discussed, which is the lacking part of this article. Based on the structure and dimensions provided by the framework, applications can be developed to illustrate actual implementation situations. In addition, the applying CD model based on the framework should adapt to various industries and corporations for which the CD is carried out, and these further discussions mentioned above can all be regarded as the index of future research topics.

References

1. Abdalla, J. A., "A communication model for structural design objects II: Performatives and protocols," Advances in Engineering Software Vol. 37, 393–405, 2006.
2. Bidault, F., Despres, C., Butler, C., "New product development and early supplier involvement (ESI): the drivers of ESI adoption", International Journal of Technology Management, Vol. 15, No.1/2, 49-69, 1998.
3. Brazier, F. M. T., Langen, P. H. G. V. and Treur, J., "Modeling conflict management in design: an explicit approach," Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, Vol.9, No.4, 353-366, 1995.
4. Brelinghoven, A., "Cooperative concurrency control for design environment," IEEE, 308-313, 1995.
5. Brigitte, B. de M., *Design management: using design to build brand value and corporate innovation*, Allworth Press, Canada, 2003.
6. Britton, G., Beng, T. S. and Wang, Y., "Virtual concurrent product development of plastic injection moulds," Proceedings of the Institution of Mechanical Engineers (Part B) Engineering Manufacture, Vol. 214, Issue 2, 165-168, 2000.
7. Browne, J., Harhen, J., and Shivnan, J., Production Management Systems: An Integrated Perspective (2nd Edition) (Addison-Wesley Publishers Ltd), 1996.
8. Bullinger, H. –J., Warschat, J., Leyh, J., and Cebulla, T., "Management, planning, design, and control," *Planning and integration of product development*, Frauhofer Institute of Industrial Engineering, 1283-1294, 2007.
9. Camarinha-Matos, L. M., and Afsarmanesh, H., "A comprehensive modeling framework for collaborative networked organizations," Journal of International Manufacturing, Vol. 18, 529–542, 2007.
10. Chen, M. –K. (陳銘崑),"用協同設計帶動企業升級：提高營運價值的策略工具，" *CSD Review (中衛報告)*, , No.3，28-31, 2008.
11. Chen, Y. T. (陳衍村)，"設計鏈的績效衡量:要好設計也要好管理，" *CSD Review (中衛報告)*, No.3，46-50, 2008.
12. Chiu, M. –L., "An organizational view of design communication in design collaboration," Design Studies Vol. 23, 187–210, 2002.
13. Chryssolouris, G., Pappas, M., Karabatsou, V., Mavrikios, D. and Alexopoulos, K., "Leveraging design process related intellectual capital-a key to enhancing enterprise agility," *Collaborative Product Design and Manufacturing Methodologies and Applications*, W.D. Li et al. (ed.), Springer, 59-68, 2006.
14. Chung, J. and Lee, K., "A framework of collaborative design environment for injection molding," Computer In Industry Vol.47, 319-337, 2002
15. Cooper, R., *Product leadership*, Perseus Books, 1998.
16. Department of Industrial Technology, R.O.C.，全球協同設計創新:15 家導入協同設計電子化成功企業經驗分享，全球協同設計創新/示範性資訊應用開發計畫專案辦公

室編，Taipei，2005.
17. Dorothy, L, -B., “Wellspring of knowledge,” Harvard business school press, 1995.
18. Forsyth, D. R., *Group Dynamics*, 4th Edition, Belmont, CA: Thomson Wadsworth, 2006.
19. Freeman, C. and Perez, C., "Structural Crises of Adjustment, Business Cycles and Investment Behavior," in DOSI et al., eds., 38-66, 1988.
20. Gajski, D.D., and Peir, J.K., “Essential issues in multiprocessor systems,” Computer, Vol.18, No.6, 9-28, 1985.
21. Gasser, L., “Information and Collaboration from a social/ organizational perspective,” Nof, S. Y. (ed.), Information and Collaboration Models of Integration, 237-261, 1994.
22. Goranson, H. T., *The Agile Virtual Enterprise: Cases, Metrics, Tools*, Quorum Books, London, 1999.
23. Gibb, B. and Damondaran, S., *ebXML: Concept and Application*, Wiley, 2002
24. Hao, Q., Shen, W., Zhang, Z., Park, S. –W., and Lee, J. -K., “Agent-based collaborative product design engineering: An industrial case study,” Computers in Industry, Vol.57, 26–38, 2006.
25. Hayes R. H., Wheelwright, S. C., and Clark K. B., *Dynamic Manufacturing: Creating the learning organization*, The Free Press, New York, 1988.
26. Hsu, S. –M., “A Study of Collaborative Design Model for the Industry of National Defence Supply Chain,” (Master's thesis), Department of Industrial and Systems Engineering, Chung Yuan Christian University, Taiwan, 2004.
27. Holt, K., *The nature of design process, in handbook of design management*, Oakley, M. (ed.), Basil Blackwell, 195-198, 1990.
28. Huang, C. –W., “The Cross-platform Collaborative Viewing and Management System,” (Master's thesis), Department of Mechanical Engineering, National Taiwan University, Taiwan, 2002.
29. Huang, G. Q., and Mak K. L., “Modeling the Customer–supplier Interface Over the World-Wide Web to Facilitate Early Supplier Involvement in the New Product Development,” Proceedings of the Institution of Mechanical Engineers -- Part B -- Engineering Manufacture, Vol. 214, Issue 9, 759-769, 2000.
30. Hunter, I., Hassan, Gayoso, D., A., and Garas, F., The eLSEwise vision, development routes and recommendations. Engineering Construction and Architectural Management, 1999.
31. Jacobson, I., Chriterson, M., Jonsson, P. and Övergaard, G., *Object-oriented software engineering: a used case driven approach*, Addison-Wesley, Reading, Massachusetts, 1996.
32. Jacobson, I., Ericson, M., and Jacobson, A., *The object advantage: business process reengineering with object technology*, Addison-Wesley, Reading, Massachusetts, 1995.
33. Jagdev, H. S. and Thoben, K. -D., “Anatomy of enterprise collaborations,” Production planning and control, VOL. 12, NO. 5, 437- 451, 2001.
34. Jagdev, H. S., and Browne, J., “The extended enterprise: a context for manufacturing,”

Production Planning and Control, Vol.9, Issue 3, 216-229, 1998.

35. Jørgensen, H. D., Karlsen, D., and Lillehagen, F., “Product Based Interoperability – Approaches and Requirements,” Challenges in Collaborative Engineering (CCE'07) Coordination of Collaborative Engineering: State of the Art and Future Challenges, 11th-13th April, 2007.
36. Keinoneon., T., “Introduction to concept design,” *Product concept design --- a review of the conceptual design of products in industry*, Keinoneon., T. and Takala, R. (Ed.), Springer, 1-31, 2006.
37. Khanna, N., Fortes, J.A.B. and Nof, S.Y., A formalism to structure and paralyze the integration of cooperative engineering design tasks, IIE Transaction, Vol. 30, 1-15, 1998.
38. Khosla, P. K. and Brown, P., “A vision for an agile design environment,” Emerging technologies, ASME, 79-86, 1995.
39. Klein, M., “Integrated support for cooperative design coordination: managing processes, conflicts and memories,” *Information and Collaboration Models of Integration*, 435-459, 1994.
40. Kung, C. –C., “A Study of Molding Involvement Product Collaborative Design、Technological Innovation on Development Performance,” 2004.
41. Llerena, P. and Matt, M., “Technology policy and cooperation: A paradigmatic approach,” DRUID (Danish Research Unit for Industrial Dynamics) Working Paper No. 00-2, 1-34, 2000.
42. Liao, Y. –C., “A Study of Product Collaborative Design Model,” (Master's thesis), Taiwan, 2002.
43. Lin, C. –C., “Research on Collaborative Management Model and A Empirical Study,” (Master's thesis), Department of Industrial and Information Management, National Cheng Kung University, Taiwan, 2004.
44. Liu, X., Raorane, S. and Leu, M. C., “A web-based intelligent collaborative system for engineering design,” Collaborative Product Design and Manufacturing Methodologies and Applications, W.D. Li et al. (ed.), Springer, 37-57, 2006.
45. Liu, A., “協同作業模式應用於產品設計探討,” CB Newsletter, No.11, 1-35, Taiwan, 2003.
46. Liu, R., -F., *Design management: Theories, procedures and applications*, Taipei, 2006.
47. Lu, C. –A., “The Impact of Product Collaborative Design on competency of Research and Development and New Product Development Performance in the Technology Industry,” (Master's thesis), Executive Master of Business Administration, National Cheng Kung University, Taiwan, 2006.
48. Lu, Y. -C. & Luh Y. –P., “An Innovation of Collaboration Product Design System” (Master's thesis), Department of Mechanical Engineering, National Taiwan University, Taiwan, 2001.
49. Lu Y.-C. & Luh Y. –P., “An Innovation of Collaboration Product Design System” (Master's thesis), Department of Mechanical Engineering, National Taiwan University,

Taiwan, 2001

50. Luo, Y., “Cooperative design in building construction” ,Collaborative Product Design and Manufacturing Methodologies and Applications, W.D. Li et al. (ed.), Springer, 93-107, 2006.
51. Ma, Y. -S., Tang, S. -H. and Chen, G., “Leveraging design process related intellectual capital-a key to enhancing enterprise agility,” *Collaborative Product Design and Manufacturing Methodologies and Applications*, W.D. Li et al. (ed.), Springer, 109-134, 2006.
52. Mahesh, M., Ong, S.K. and Nee, A. Y. C., “A web-based framework for distributed and collaborative manufacturing,” *Collaborative Product Design and Manufacturing Methodologies and Applications*, W.D. Li et al. (ed.), Springer, 137-148, 2006.
53. Mcshane, S. and Travaglione, T.,”Organizational behavior” Australia: McGraw-Hill, 2003.
54. Msanjila, S. S. and Afsarmanesh, H., “Trust analysis and assessment in virtual organization breeding environments,” International Journal of Production Research, Vol. 46, No. 5, 1253–1295, 2008.
55. Mills, J. J. and Goossenaerts, J. 2001, “Toward information and Knowledge in product realization Infrastructures.” International J.P.T. Mo and L. Nemes (ed.) *Global Engineering, Manufacturing and Enterprise Networks*, Kluwer Academic Publishers, Boston, Dordrecht, London, 245-254, 2001.
56. Mills, J. J. and Goossenaerts, J.,” Towards Information and Knowledge in Product Realization Infrastructures,” Fourth International Conference on the Design of Information Infrastructure Systems for Manufacturing: Global Engineering, Manufacturing and Enterprise Networks table of contents, 245 - 254, 2001.
57. Müller, H. J., “Towards agent systems engineering,” Data and knowledge engineering, Vol. 23, 217-245, 1997.
58. Mervyn, F., Kumar, A. S. and Nee, A. Y. C., “Leveraging design process related intellectual capital-a key to enhancing enterprise agility,” *Collaborative Product Design and Manufacturing Methodologies and Applications*, W.D. Li et al. (ed.), Springer, 71-90, 2006.
59. Nelson, E. M., “Open architecture for the enterprise, Part 1: Architectural principles of open architecture: Be cheap, flexible, and interoperable in 7 easy steps,” IBM official websites, 2007.
60. Panchal, J.H., Choi, H. -J., Aellen, J. K., Rosen, D. and Mistree, F., “Leveraging design process related intellectual capital-a key to enhancing enterprise agility,” *Collaborative Product Design and Manufacturing Methodologies and Applications,* W.D. Li et al. (ed.), Springer, 1-32, 2006.
61. Panchal, J.H., Fernández, M.G., Paredis, C. J. J., Aellen, J. K. and Mistree, F., “Leveraging design process related intellectual capital- a key to enhancing enterprise agility,” *Collaborative Product Design and Manufacturing Methodologies and*

Applications, W.D. Li et al. (ed.), Springer, 201-228, 2006.
62. Papastavrou, J. D. and Nof, S. Y., “Decision Integration Fundamentals in Distributed Manufacturing Topologies,” IIE Transactions, Vol.24, No.3, 27-42, 1992.
63. Parunak, HVD. *Manufacturing experience with the contract net*. In: Huhns M, editor. Distributed artificial technology. Los Altos, CA: Pitman Pubs. /Morgan Kaufman, 285–310, 1987.
64. Robin, V., Sperandio, S., Blanc, S., Girard, P., “Interactions modeling between factors influencing performance of the design process,” International Conference on Engineering Design, ICED2005, Melbourne, 2005.
65. Robin, V., Rose, B., Girard, P., Modeling collaborative knowledge to support engineering design project manager, Computers in Industry, Vol.58, 188–198, 2007.
66. Roller, D., Mešina, -I. M., Dalakakis, S., “An Active Semantic Network for Rapid Prototyping,” Informatics and Algorithms, 12-13, 2002.
67. Rothwell, R., “Toward the fifth-generation innovation process,” International marketing review, Vol.11, Issues 7, 7-31, 1994.
68. Sääskilahti, M., Takala R., “Vision concept,” *Product concept design --- a review of the conceptual design of products in industry,* Keinoneon., T. and Takala, R. (Ed.), Springer, 177-195, 2006.
69. Shen, W., Hao, Q., Wang, S., Li, Y. and Ghenniwa, H., “An agent-based service-oriented integration architecture for collaborative intelligent manufacturing,” *Robotics and Computer-Integrated Manufacturing,* Vol. 23, No.3, 315-325, 2007.
70. Su, H. –J., “Modeling and Analysis of the Collaborative Product Development Process: Development and Application of Extended Design Structure Matrix,” (Master's thesis), Industrial Management Institute, National central university, Taiwan, 2004.
71. Tang, D., “Partnership development between product customer and tool and die supplier,” Proceedings of the Institution of Mechanical Engineers -- Part B -- Engineering Manufacture, Vol. 219, Issue 4, 365-376, 2005.
72. Tang, S. H., Ma, Y. S. and Chen, G., “A feature-oriented database framework for web-based CAx applications,” Computer-Aided Design & Applications, Vol. 1, 117-125, 2004.
73. Takala, R., Keinoneon, T., and Mantere, J., “Processes of product concepting,” *Product concept design --- a review of the conceptual design of products in industry,* Keinoneon., T. and Takala, R. (Ed.), Springer, 58-90, 2006.
74. Teece, D.J., “The market for know-how and the efficient international transfer of technology,” Annals of the American academy of political and social science, Vol.458, 81-96, 1981.
75. Teece, D.J., “Profiting from technological innovation: implications for integration, collaboration, licensing and public policy,” Research policy, Vol. 15, 285-305, 1986.
76. Thomas, J.P. and Baker, K.D., “Knowledge representation for design,” IEEE International Conference on SCM, vol.3, 1827-1832, 1992.

77. Thomson, A., Stone, A., and Ion, W., "Distributed team design in small- and medium-sized enterprises: How to get it right," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 21, 203–218, 2007.
78. Trappey, A. J. C., Hsiao, D. W., "Applying collaborative design and modularized assembly for automotive ODM supply chain integration," Computers in Industry doi:10.1016/j.compind, 2007.
79. Ulrich, K.T. and Eppinger, S.T., *Product design and development,* McGraw-Hill, 4th Edition, 2008.
80. Ulrich, K. T. and Eppinger, S. D., *Product design and development*, Chang, S. –W. and Day, D., 2nd Edition, Taipei, 2008.
81. Wang, C., K. Li, Ghenniwa, H., Shen, W. and Wang, Y., "Real time distributed stop floor scheduling: an agent-based service-oriented framework," *Collaborative Product Design and Manufacturing Methodologies and Applications*, W.D. Li et al. (ed.), Springer, 175-197, 2006.
82. Wang, H. F. and Zhang, Y. L., "CAD/CAM integrated system in collaborative development environment," Robotics and Computer-Integrated Manufacturing, Vol.18, 135-145, 2002.
83. Wang, C. –C., "The Study of Performance Measurement Indicator on Product Collaborative Design," (Master's thesis), Department of Industrial Engineering and Management, Yuan Ze University, Taiwan, 2005.
84. Wang, H. –C., "Applying Cognitive Work Analysis to Organize the Collaborative Design Team," (Master's thesis), Institute of Industrial Management, National Central University, Taiwan, 2006.
85. Witzerman, J. P. and Nof, S. Y., "Tool integration for collaborative design of manufacturing cells," International journal of production economics, 23-30, 1995.
86. Wu, J. –H., and Lin, H. H., Systems analysis and design: theory and application, Best-wise, 2nd Edition, Taipei, 2001.
87. Yang, Q.Z. and Zhang, Y., "Semantic interoperability to support collaborative product development," *Collaborative Product Design and Manufacturing Methodologies and Applications*, W.D. Li et al. (ed.), Springer, 255-284, 2006.
88. http://www.workflowpatterns.com/
89. http://www.cr1.dircon.co.uk/
90. http://en.wikipedia.org/wiki/Main_Page
91. http://www.npd-solutions.com/pdm.html (Kenneth Crow, DRM Associates)
92. http://en.wikipedia.org/wiki/Product_Data_Management
93. http://en.wikipedia.org/wiki/Product_lifecycle_management
94. http://www.product-lifecycle-management.com/ (Active Sensing, Inc.)
95. http://www.ibm.com/developerworks/library/ar-openarch1/ (Nelson, E. M., IBM)