



Design and integration of multi-disciplinary interfaces : method and modelling language for mechatronic systems engineering

Chen Zheng

► To cite this version:

Chen Zheng. Design and integration of multi-disciplinary interfaces : method and modelling language for mechatronic systems engineering. Mechanics [physics.med-ph]. Université de Technologie de Compiègne, 2015. English. NNT : 2015COMP2241 . tel-01451708

HAL Id: tel-01451708

<https://theses.hal.science/tel-01451708>

Submitted on 1 Feb 2017

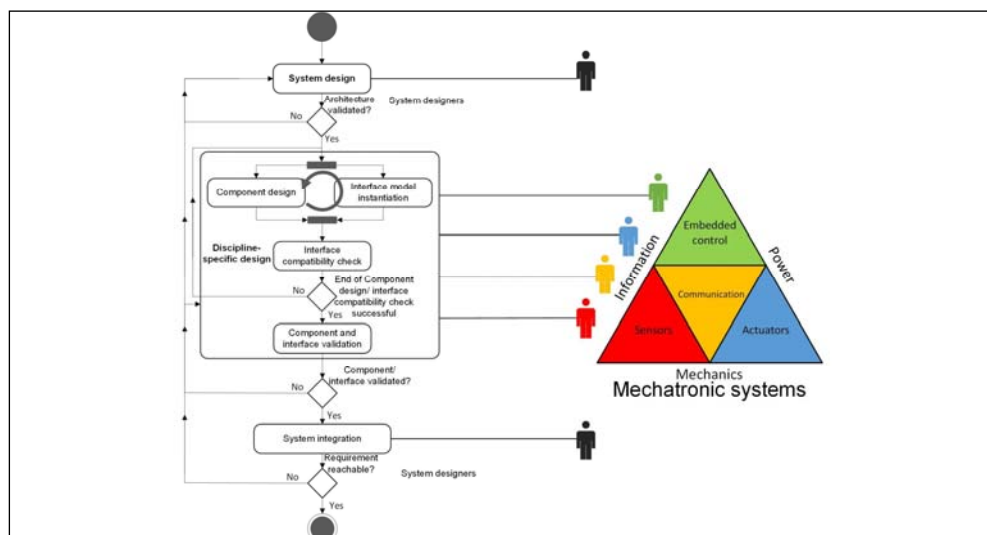
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Par **Chen ZHENG**

Design and integration of multi-disciplinary interfaces : method and modelling language for mechatronic systems engineering

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 14 décembre 2015
Spécialité : Mécanique Avancée

D2241

N°d'ordre :

Ecole doctorale n°71 : Science pour l'Ingénieur

THÈSE DE DOCTORAT
pour obtenir le grade de docteur délivré par

L'UNIVERSITE DE TECHNOLOGIE DE COMPIEGNE

Spécialité Mécanique Avancée

Présentée le 14 décembre 2015 et soutenue publiquement par

Chen ZHENG

**Design and Integration of Multi-disciplinary Interfaces: Method and
Modelling Language for Mechatronic Systems Engineering**

Membres du jury :

RIERA Bernard, Professeur des Universités, Université de Reims Champagne Ardenne,
rapporteur

YAN Xiu-Tian, Professor, University of Strathclyde, *rapporteur*

BRICOGNE Matthieu, Enseignant Chercheur, Université de Technologie de Compiègne,
examineur

EYNARD Benoît, Enseignant Chercheur HDR, Université de Technologie de Compiègne,
directeur de thèse

LE DUGOU Julien, Enseignant Chercheur, Université de Technologie de Compiègne,
codirecteur de thèse

GIRARD Philippe, Professeur des Universités, Université de Bordeaux, *examineur*

HEHENBERGER Peter, Associate Professor, Johannes Kepler University Linz,
examineur

LAMARQUE Frédéric, Professeur des Universités, Université de Technologie de
Compiègne, *examineur*

Invité : **NOPPE Eric**, Enseignant Chercheur, Université de Technologie de Compiègne

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES.....	5
LIST OF TABLES.....	8
LIST OF ACRONYMS.....	9
ACKNOWLEDGEMENT	10
THESIS STRUCTURE.....	12
SCIENTIFIC CONTEXT AND RESEARCH QUESTIONS.....	14
□ 1.1 CONTEXT	14
1.1.1 <i>Concept and history of mechatronics</i>	14
1.1.2 <i>Development trends of mechatronics engineering</i>	16
□ 1.2 SCIENTIFIC PROBLEMS AND RESEARCH OBJECTIVES.....	24
1.2.1 <i>Scientific problems</i>	24
1.2.2 <i>Research objectives</i>	25
□ 1.3 RESEARCH QUESTIONS	26
1.3.1 <i>Research questions related to the design process</i>	26
1.3.2 <i>Research questions related to the design data</i>	28
1.3.3 <i>Relationship between the two main research questions</i>	29
STATE OF THE ART.....	31
□ 2.1 PRODUCT DESIGN	31
2.1.1 <i>Product Lifecycle Management</i>	32
2.1.2 <i>Systems Engineering</i>	35
2.1.3 <i>Synthesis of two product design approaches: PLM and SE</i>	40
□ 2.2 DESIGN METHODS FOR MECHATRONIC ENGINEERING	41
2.2.1 <i>Sequential design process</i>	41
2.2.2 <i>V-model and its variants</i>	43
2.2.3 <i>Hierarchical design model</i>	49
2.2.4 <i>Assessment of existing design methods for mechatronic engineering</i>	50
□ 2.3 PRODUCT MODELS FOR MECHATRONIC ENGINEERING	52
2.3.1 <i>STEP (STandard for the Exchange of Product)</i>	52
2.3.2 <i>CPM (Core Product Model)</i>	54
2.3.3 <i>PPO (Product-Process-Organisation) Model</i>	56
2.3.4 <i>Assessment of existing product models for mechatronic engineering</i>	57

TABLE OF CONTENTS

□ 2.4 ONTOLOGY-BASED PRODUCT MODELLING APPROACHES FOR MECHATRONIC ENGINEERING	60
2.4.1 <i>Basic concept of ontology</i>	60
2.4.2 <i>Current research works on ontology-based product modelling approaches</i>	60
2.4.3 <i>Assessment of existing ontology based product modelling approaches</i>	62
□ 2.5 SUMMARY AND ASSESSMENT OF STUDIED RESEARCH WORKS	63
□ 2.5 RESEARCH APPROACHES.....	64
PROPOSITIONS.....	66
□ 3.1 MULTI-DISCIPLINARY INTERFACE MODEL	66
3.1.1 <i>Introduction of LEGO Mindstorms used for training the basics of mechatronic engineering</i>	67
3.1.2 <i>Interface classification</i>	71
3.1.3 <i>Interface model</i>	82
3.1.4 <i>Interface compatibility rules for design of mechatronic systems</i>	88
3.1.5 <i>Synthesis of multi-disciplinary interface model</i>	92
□ 3.2 DESIGN METHODOLOGY BASED ON MULTI-DISCIPLINARY INTERFACE MODEL	93
3.2.1 <i>General design process</i>	93
3.2.2 <i>Design phases</i>	94
3.2.3 <i>Achievement of macro level and micro level collaboration</i>	105
□ 3.3 SUMMARY	107
CASE STUDY	108
□ 4.1 DEMONSTRATOR DEVELOPMENT	109
4.1.1 <i>Introduction of 3DEXPERIENCE Platform</i>	109
4.1.2 <i>Implementation of the design methodology</i>	110
4.1.3 <i>Implementation of the multi-disciplinary interface model</i>	112
□ 4.2 3D MEASUREMENT SYSTEM	117
4.2.1 <i>System design sub-process</i>	118
4.2.2 <i>Discipline-specific design sub-process</i>	123
4.2.3 <i>Synthesis of case study</i>	131
□ 4.3 SUMMARY	132
CONCLUSIONS AND FUTURE WORK.....	135
□ 5.1 SUMMARY ON THE RESEARCH QUESTIONS.....	135
□ 5.2 RESEARCH CONTRIBUTIONS TO DESIGN OF MECHATRONIC SYSTEMS	139
□ 5.3 FUTURE WORK	141
REFERENCES.....	145
LIST OF PUBLICATIONS	159

TABLE OF CONTENTS

ABSTRACT	160
----------------	-----

70

RESUME.....	164
-------------	-----

List of Figures

FIGURE 1.1 HISTORICAL DEVELOPMENT OF MECHANICAL, ELECTRONIC AND MECHATRONIC SYSTEMS (ISERMANN, 2007)	16
FIGURE 1.2 DETAILED DESCRIPTION OF MECHATRONIC SYSTEM (CRAIG & MARCHI, 1996)	16
FIGURE 1.3 MECHANICAL SYSTEMS INTEGRATING ELECTRONICS IN INTERACTION WITH INFORMATION AND POWER (SCHÖNER, 2004)	18
FIGURE 1.4 DIFFERENT INTEGRATION LEVELS IN MECHATRONIC SYSTEMS (PENAS, PLATEAUX, CHOLEY, & RIVIERE, 2009)	21
FIGURE 1.5 RELATIONSHIP BETWEEN THE RESEARCH QUESTIONS	32
FIGURE 2.1 CONCEPTUAL PLM SYSTEM ARCHITECTURE (RACHURI ET AL., 2005)	35
FIGURE 2.2 SYSTEMS ENGINEERING PROCESS (LEONARD, 1999)	38
FIGURE 2.3 SE DESCRIBED BY ANSI/EIA632 (MARTIN, 2000)	39
FIGURE 2.4 SE DESCRIBED BY IEEE 1220 (DORAN, 2004)	40
FIGURE 2.5 ISO/IEC 15288 PROCESS MODEL (ISO/IEC 15288, 2002)	41
FIGURE 2.6 SEQUENTIAL DESIGN PROCESS (ALVAREZ CABRERA ET AL., 2010)	44
FIGURE 2.7 V-MODEL (DEPARTMENT OF TRANSPORTATION, 2007)	46
FIGURE 2.8 VDI GUIDELINE 2206 (VDI 2206, 2003)	47
FIGURE 2.9 DEVELOPMENT PROCESS BASED ON SE AND RFLP (KLEINER & KRAMER, 2013)	48
FIGURE 2.10 UML CLASS DIAGRAM OF THE CORE PRODUCT MODEL (FENVES ET AL., 2006)	57
FIGURE 2.11 INTERFACE FEATURE OF EMBEDDED SYSTEM MODEL (ZHA ET AL., 2005)	58
FIGURE 2.12 PRODUCT MODEL CLASS DIAGRAM (NOËL, ROUCOULES, & TEISSANDIER, 2005)	59
FIGURE 2.13 PROCESS MODEL DEVELOPED DURING THE IPPOP PROJECT (NOWAK ET AL., 2004) ...	60
FIGURE 3.1 MECHATRONICS AND LEGO	71
FIGURE 3.2 EXAMPLES OF LEGO TECHNIC PIECES (DIMENSIONSGUIDE, 2011)	71
FIGURE 3.3 PROGRAMMING BLOCKS OF LEGO MINDSTORM	72
FIGURE 3.4 POWER SUPPLY OF LEGO MINDSTORM	72
FIGURE 3.5 SENSORS OF LEGO MINDSTORM	73
FIGURE 3.6 ACTUATORS OF LEGO MINDSTORM	73
FIGURE 3.7 EV3 BRICK OF LEGO MINDSTORM	74
FIGURE 3.8 EXAMPLE OF GEOMETRIC INTERFACE	77
FIGURE 3.9 EXAMPLE OF ENERGY INTERFACE	78
FIGURE 3.10 EXAMPLE OF CONTROL INTERFACE	78
FIGURE 3.11 EXAMPLE OF DATA INTERFACE	79
FIGURE 3.12 EXAMPLE OF C-I-C	80
FIGURE 3.13 EXAMPLE OF C-I-E	81
FIGURE 3.14 EXAMPLES OF C-I-I AND I-I-I	82
FIGURE 3.15 EXAMPLES OF I-I-E	83
FIGURE 3.16 EXAMPLE OF AN INTERFACE CREATING POSITIVE EFFECTS AND UNINTENDED SIDE-	

LIST OF FIGURES

EFFECTS SIMULTANEOUSLY	84
FIGURE 3.17 UML CLASS DIAGRAM OF INTERFACE AND ITS CLASSIFICATION.....	84
FIGURE 3.18 THREE TYPES OF PORTS EXISTING IN THE DIFFERENT INTERFACE CONFIGURATIONS	86
FIGURE 3.19 UML CLASS DIAGRAM OF PORT	89
FIGURE 3.20 UML CLASS DIAGRAM OF INTERFACE	90
FIGURE 3.21 SOLUTIONS TO SOLVE INCOMPATIBLE INTERFACES	95
FIGURE 3.22 EXAMPLE OF SOLUTIONS TO SOLVE INCOMPATIBLE INTERFACES	96
FIGURE 3.23 GENERAL DESIGN PROCESS: AN EXTENDED V-MODEL	98
FIGURE 3.24 CONSISTENCY BETWEEN DIFFERENT DESIGN PHASES DURING SYSTEM DESIGN PROCESS.	
102	
FIGURE 3.25 CONSISTENCY IN THE THREE DESIGN PHASES FOR THE LEGO ROBOT	104
FIGURE 3.26 UML ACTIVITY DIAGRAM FOR THE DISCIPLINE-SPECIFIC DESIGN SUB-PROCESS BASED ON MULTIDISCIPLINARY INTERFACE MODEL.....	108
FIGURE 3.27 INSTANCE OF THE INTERFACE BETWEEN THE RECHARGEABLE BATTERY AND THE EV3 BRICK	109
FIGURE 4.1 VPM FUNCTIONAL LOGICAL EDITOR	114
FIGURE 4.2 COMPONENT REPRESENTATION IN 3DEXPERIENCE PLATFORM	117
FIGURE 4.3 ENVIRONMENT REPRESENTATION IN 3DEXPERIENCE PLATFORM	118
FIGURE 4.4 INTERFACE REPRESENTATION IN 3DEXPERIENCE PLATFORM	119
FIGURE 4.5 REPRESENTATION OF PORT AND ITS PARAMETERS IN 3DEXPERIENCE PLATFORM	120
FIGURE 4.6 REQUIREMENTS SPECIFICATION USING WORD, ENOVIA AND 3DEXPERIENCE PLATFORM (KLEINER, 2013)	115
FIGURE 4.7 INTERFACE COMPATIBILITY RULE AND THE TEST RESULT	116
FIGURE 4.8 PRINCIPLE OF THE MEASUREMENT MODE: (A) ACTIVE MODE 1 (B) ACTIVE MODE 2 (C) PROJECTED PATTERN (D) DEFORMED PATTERN	122
FIGURE 4. 9 FUNCTIONAL AND ARCHITECTURAL MODEL FOR 3D MEASUREMENT SYSTEM	126
FIGURE 4.10 REPRESENTATION OF PATTERN PROJECTION SUB-SYSTEM	127
FIGURE 4.11 INTERFACE (I1.1) BETWEEN THE INDUSTRIAL ENVIRONMENT (E1) AND THE IMAGE GUIDE (C1.13)	128
FIGURE 4.12 IMAGE GUIDE DATA SHEET	128
FIGURE 4.13 INSTANCE OF THE INCOMPATIBLE INTERFACE I1.1	129
FIGURE 4.14 SOLUTION 1 DEALING WITH THE INCOMPATIBLE INTERFACE I1.1	130
FIGURE 4.15 INSTANCE OF THE COMPATIBLE INTERFACE I1.1	130
FIGURE 4.16 INTERFACE (I1.12) BETWEEN THE DMD (C1.12) AND THE IMAGE GUIDE (C1.13)	130
FIGURE 4.17 DMD DATA SHEET AND MAXIMUM IMAGE CIRCLE DIAMETER	131
FIGURE 4.18 INSTANCE OF THE INCOMPATIBLE INTERFACE I1.12	131
FIGURE 4.19 SOLUTION 2 DEALING WITH THE INCOMPATIBLE INTERFACE I1.121	132
FIGURE 4.20 INSTANCE OF THE COMPATIBLE INTERFACE I1.1211	132
FIGURE 4.21 SOLUTION 2 DEALING WITH THE INCOMPATIBLE INTERFACE I1.122	134

LIST OF FIGURES

FIGURE 4.22 PART OF 3D MEASUREMENT SYSTEM	134
---	-----

List of Tables

Table 1.1 Main research questions and their sub-questions	31
Table 2.1 Details of the evaluation for design methods	53
Table 2.2 Assessment of the design methods regarding needs of multi-disciplinary collaboration	54
Table 2.3 Details of the evaluation for product models	61
Table 2.4 Assessment of the product models regarding needs of multi-disciplinary integration	62
Table 2.5 Assessment of the ontology based product modelling approaches regarding needs of multi-disciplinary collaboration	65 Table
3.1 Main limitations of previous classifications	76
Table 3.2 Five types of interfaces	80
Table 3.3 Mapping between main entities of multi-disciplinary interface model and their equivalent entities in other product models	92
Table 4.1 Requirements, functions, components and design teams/standard component	125

List of Acronyms

AFIS:	Association Française d'Ingénierie Système
AP:	Application Protocol
BDD:	Block Definition Diagram
CAD:	Computer Aided Design
CAX:	Computer Aided x
CCD:	Charge-Coupled Device
CPM:	Core Product Model
CPS:	Cyber-Physical Systems
CVS:	Concurrent Versions System
DMD:	Digital Micro-mirror Device
E-CAD:	Electronic/Electrical Computer Aided Design
EDM:	Engineering Data Management
EES:	Electrical/Electronic Engineering Solutions
EIA:	Electronic Industries Association
EMC:	ElectroMagnetic Compatibility
E-PDM:	Electronic/Electrical Product Data management
ESM:	Embedded System Model
FBS:	Function–Behaviour–State
GUI:	Graphical User Interface
HW:	Hardware
IBD:	Internal Block Diagram
IGES:	Initial Graphics Exchange Specification
INCOSE:	International Council on Systems Engineering
IoT:	Internet of Things
IPPOP:	Integration of Product, Process and Organisation for improvement of engineering Performance
IT:	Information Technology
KBE:	Knowledge Based Engineering

ACKNOWLEDGEMENT

M-CAD: Mechanical Computer Aided Design

MPD: Modular Product Development

M-PDM: Mechanical Product Data management

NIST: National Institute of Standards and Technology

OCL: Object Constraint Language

PCA: Printed Circuit Assemblies

PDM: Product Data management

PLC: Product Life Cycle

PLM: Product Lifecycle Management

PPO: Product-Process-Organisation

PSRL: Product Semantic Representation Language

SCM: Software Configuration Management

SE: Systems Engineering

SoS: System of Systems

STEP: STandard for the Exchange of Product

SW: Software

UML: Unified Modelling Language

XML: Extensible Markup Language

Acknowledgement

When I am writing of the last sentence of this thesis, I realise that this piece of work is much more than my one person's effort. Without the support and help from my thesis supervisors, my family and my friends, I would not have arrived at this point. First, I would like to give my sincere gratitude to my supervisors, Professor Benoît Eynard and Associate Professor Julien Le Duigou, for the continuous support of my PhD study and research. Their patience, motivation, immense knowledge, and their inspiring guidance during my thesis deserve the most appreciation and respect in my

ACKNOWLEDGEMENT

heart. It is really a great experience working with them. With their abundant research experience, they show me how to become an independent researcher.

Second, I would like to thank Associate Professor Matthieu Bricogne for his help and guidance throughout the three-year research. I benefited greatly for having useful opinions and comments in weekly meetings, which greatly help me for conducting the thorough research in the field of mechatronic systems engineering.

Third, I address my acknowledgments to Professor Frédéric Lamarque, Associate Professor Laurent Petit, Associate Professor Erwan Dupont and Ph.D student Yingfan Hou for their support of the case study on the 3D measurement system. All their constructive comments and advice that have enriched my work, the different exchanges of ideas and interpretations, the time devoted to interviews and their professional experiences and knowledge in the domain of design of system mechatronic systems.

Fourth, I express thanks to the reviewers, Professor RIERA Bernard from the Université de Reims Champagne and Professor YAN Xiu-Tian from the University of Strathclyde, for agreeing to bring my work thesis. It is an honour for me to extend my thanks to the members of the jury for the time and attention they devoted to reading this work and their criticism of it.

Fifth, I would not fail to acknowledge to the Department of Mechanical Design Engineering of the Laboratory Roberval at the University de Technologie de Compiègne (UTC), where the work presented in this thesis was conducted. I would further like to give my gratitude to the financial support from the program of China Scholarships Council (CSC).

Sixth, I would like to say, I really enjoy working in the friendly atmosphere in this department. I wish to take this opportunity to express my appreciation and thanks to all the students, professors, secretaries and staff for their friendly spirit. Especially, I would like to thank all the members of I101, they are Marina Bruneau, Fabien Michelin, Christophe Danjou, Fabien Mahut, Cong Cuong Pham and Jinhua Xiao. Finally, I would like to express special thanks to my parents and my girl friend, who provide me eternal love and supports. They encourage me to explore knowledge and teach me to be a man with responsibility. With all the love and faith, I believe that tomorrow is going to be better.

Thesis structure

The topic of this thesis is on design and integration of mechatronic systems. Mechatronic system is considered as the resulting integration of electrical/electronic system, mechanical parts and information processing. Therefore to enable a systematic design process of mechatronic systems with high integration level, the so-called “multi-disciplinary integration” is required. The interest of the thesis concerns how designers can achieve such multidisciplinary integration during the design process of mechatronic systems. The structure of the thesis is organised as follows:

Chapter 1 introduces the context of the research, including the concept, the historical development and the technology trends of mechatronic system. Considering the technology trends of mechatronic systems, in order to achieve a more integrated mechatronic system, the research objectives are then pointed out. However, two scientific problems - process-based problems and design data-related problems are considered as the barriers which hinder the designers to achieve the multi-disciplinary integration. According to these scientific problems, the research questions are put forward.

Chapter 2 reviews relevant literature, which sets the stage for the need of this thesis research. Considering the research objective proposed in Chapter 1, this chapter focuses on the current studies on process models and product models, which are respectively considered as the potential solutions to address the process-based problems and design data-related problems. The evaluation results of the current studies shows that the multi-disciplinary interfaces in the mechatronic systems should be further studied and developed.

Chapter 3 focuses on the propositions on multi-disciplinary interface. In this chapter the approach of multi-disciplinary interface modelling is detailed to deal with the design data-related problems. A design methodology based on the multi-disciplinary interface modelling approach is then introduced in order to solve the process-based problems.

Chapter 4 proposes a demonstrator based on 3DEXPERIENCE Platform to demonstrate the feasibility the propositions. A 3D measurement system is adopted as

SCIENTIFIC CONTEXT AND RESEARCH QUESTIONS

a case study to show how the propositions can be applied during the design process and to validate the expected improvements.

THESIS STRUCTURE

Chapter 5 summarises the research work of the thesis. It concludes the thesis with the contributions and limitations of the propositions. The future research directions are finally pointed out.

Scientific context and research questions

Before concentrating on the research propositions, the thesis firstly provides the context by introducing the concept, the history and the technology trends. The research objectives will be then pointed out by regarding to nowadays technology trends. However, some problems still exist in current design of mechatronic systems. Considering the problems, the research questions will be finally pointed out.

□ 1.1 Context

1.1.1 Concept and history of mechatronics

Mechanical systems have been used to generate motion, transfer forces and torque for many decades. The control systems have been gradually involved to manipulate the commanded variables of the mechanical systems (Isermann, 2007). This kind of evolution allows fulfilling much more functions compared to a pure mechanical system or pure control system. But all this could still be just called an automated system; mechatronics means more than this. Figure 1.1 summarizes the development from basic mechanical systems of the 19th century to mechatronic systems in the 1980s.

The term Mechatronics originated at the Yaskawa Corporation from the combination of mechanics and electronics in 1969. After the 1970s, the meaning of mechatronic has been broadened to include software and computation (Carrier, Ohline, & Kenny, 2011). Nowadays, mechatronics becomes the synergistic integration of mechanical parts with electrical/electronic systems and information processing (Tomizuka, 2002). Thus there are several definitions of mechatronics as a scientific discipline, but one of the most accurate definitions could be - “the synergistic integration of mechanical engineering with electronics and intelligent computer control in the design and manufacturing of industrial products and processes” (Kyura & Oho, 1996). An iconic description of such mechatronic system, multi-disciplinary design is presented in Figure. 1.2.

SCIENTIFIC CONTEXT AND RESEARCH QUESTIONS

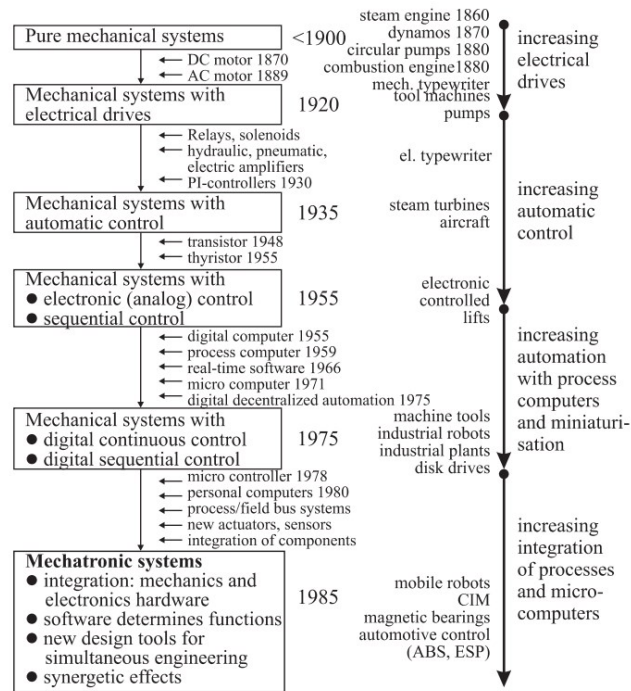


Figure 1.1 Historical development of mechanical, electronic and mechatronic systems

(Isermann, 2007)

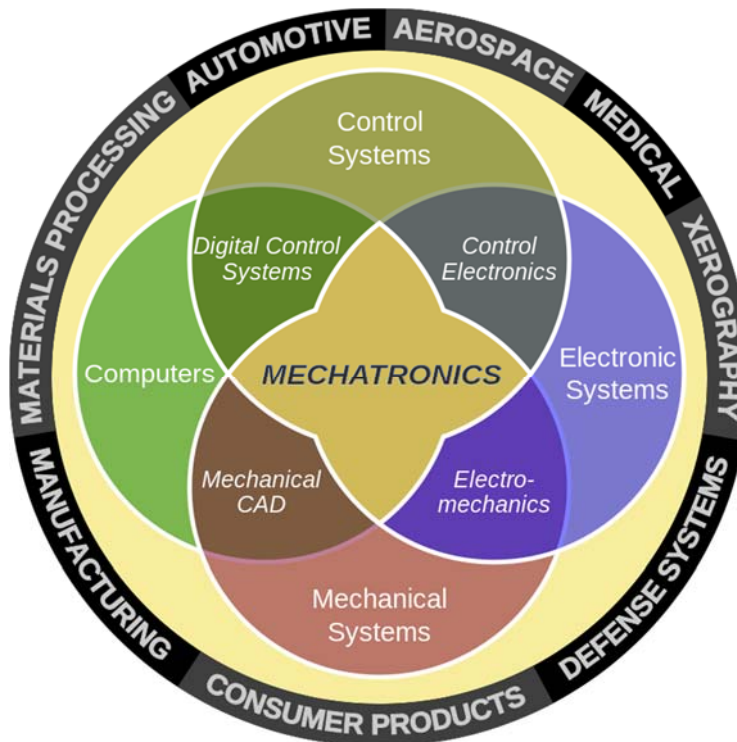


Figure 1.2 Detailed description of mechatronic system (Craig & Marchi, 1996)

With the development of technology, other disciplines (e.g. optics, hydraulics, pneumatics, etc.) are involved in the development of mechatronic systems

(Gausemeier, Frank, Donoth, & Kahl, 2009). The application domain of mechatronic systems have been accordingly enlarged, which leads to modern smart products.

By now the concept and the historical development of mechatronic system have been introduced. This historical development indicates that the mechatronic systems become more and more integrated. According to the definition proposed by (Kyura & Oho, 1996), the design of mechatronic systems should focus not only on the synergistic integration for the products, but also on the integration of the several involved disciplines during the design processes. Therefore much more attention should be paid to such two kinds of integration in the future. The development trends of mechatronics engineering related to the product and the process will be introduced in next sub-sections.

1.1.2 Development trends of mechatronics engineering

The above sub-section introduces the basic concept of mechatronic systems and its development history, which reveals the two kinds of integration relevant to the design of mechatronic systems-integration related to the product and the process. This sub-section will present the development trends of mechatronics engineering from the two types of integration. Considering the technology trends of mechatronic systems, the research objectives are then pointed out.

1.1.2.1 Integration related to the product

The previous discussion indicates that the mechatronic systems become more and more integrated. The integration related to the product will be introduced in this sub-section. According to Bricogne, the integration related to the product can be divided into functional integration and physical integration (Bricogne, 2015).

With the development of technology, more and more functionalities have been integrated into mechatronic systems. Figure 1.3 from (Schöner, 2004) presents such evolution, the different involved engineering disciplines and the overlaps between them. First, Actuators (A), represented on the right angle of the triangle, are added. They are in charge of managing actuation forces and speed. It can be regarded as the first combination of electronic and mechanical disciplines. To supply power to these actuators, external power is needed and generally provided by electrical engineering disciplines. The integration related to the product of mechatronic systems allows fulfilling more functions in more compact systems compared to a pure mechanical system or pure electronic system. “But all this could still be just called an automated

system; mechatronics means more than this” (Schöner, 2004). Mechatronic systems are the resulting product of a global concurrent engineering or integrated design process (Sohlenius, 1992; Tichkiewitch, 1994). This aspect will be detailed in the section, relative to process integration. Second, the Embedded control (E), comes “with the goal of an automatic or more reproducible process” (Schöner, 2004). On the top angle of the triangle, embedded systems, considered as the overlap between the electronic and software disciplines, have been gradually included in modern mechatronic systems for information processing (Marwedel, 2011). Third, the Sensors (S), on the left angle, allow the system to get detailed information about the status of the system and to fulfil correctly its functions to the various environmental conditions. It is considered as the overlap between the mechanical and IT disciplines. Finally, the Communication (C) is now considered as the central part of the system, especially for distributed systems.

It allows integrating the sub-system into the whole product/system.

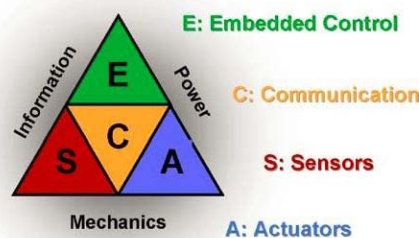


Figure 1.3 Mechanical systems integrating electronics in interaction with information and power (Schöner, 2004)

Led by the development of several technologies, mechatronic systems are influenced by some new development trends. For instance, with the support of information processing and cyber technologies, the information and communication are much more closely integrated into mechatronic systems. During the 1990s, this trend shifted towards information processing being associated with personal computers processing (Marwedel, 2011). The convergence of information resources has happened and the entire physical world itself is gradually becoming a type of information system in the real time (Parwekar, 2011). Nowadays we are experiencing the fourth industrial revolution named information revolution (Hermann, Pentek, & Otto, 2015). In this context, the term Industry 4.0 was manifested for the first time at the

Hannover Fair with the presentation of “Industry 4.0” initiative (Jazdi, 2014). Based on the information and communication technologies, various types of systems, such as embedded systems, Cyber-Physical Systems (CPS), Internet of Things (IoT),

System of Systems (SoS) etc., have attracted the attention from both academia and industry. By making use of the above systems, Industry 4.0 provides a scenario in which human can monitor physical processes, create a virtual copy of the physical world and make decision.

The embedded system, as introduced before, can then be considered as the link between the cyber world and the physical systems. However, the strong link to physics has been stressed even more by introducing the term “Cyber-Physical Systems”. CPS are defined as “integrations of computation and physical processes” (Lee, 2008). CPS begins to focus on the integration of knowledge and engineering principles across the computational and engineering disciplines (network, control, software, human interaction, learning theory, as well as electrical, mechanical, chemical, biomedical, material science, and other engineering disciplines) to develop new CPS science and supporting technology (Baheti & Gill, 2011).

Another concept influencing mechatronic field is Internet of Things, which is quite similar to CPS. Firstly proposed by Kevin Ashton in a presentation in 1998 (Weber & Weber, 2010), IoT is defined as “a world where physical objects are seamlessly integrated into the information network, and where the physical objects can become active participants in business processes. Services are available to interact with these ‘smart objects’ over the Internet, query their state and any information associated with them, taking into account security and privacy issues” (Haller, Karnouskos, & Schroth, 2009). Over the IoT, CPS communicate and cooperate with each other and humans in real time (Hermann et al., 2015). However, the frontier between CPS and IoT has not been clearly identified since both concepts have been driven in parallel from two independent communities, although they have always been closely related (Koubaa & Andersson, 2009). CPS can be considered as an opportunity and a backbone to promote IoT concept and associated technologies. They are really closed to mechatronic systems, even if the focus is more on the software parts of the system, whereas mechatronic systems are traditionally more focused on the hardware parts of the system (Bricogne, 2015).

One example of research towards the implementation of IoT is the field of “Intelligent product” (Hribernik, Ghrairi, Hans, & Thoben, 2011). (McFarlane, Sarma, Chirn, Wong, & Ashton, 2003) defines the intelligent product as “a physical and information based representation of an item which possesses a unique identification”. It is able to communicate efficiently with its environment, can collect and/or store data about itself, deploys a language to display its requirements, features and etc. It is capable of contributing to or making decisions relevant to its own destiny.

Mechatronic system can then be considered as a platform supporting the development of IoT/Intelligent product.

The above introduction of CPS and IoT indicates that there has been a growing interest in a class of systems, rather than a single complex system. Such class of systems is sometimes called Federation of Systems (FoS) or System of Systems (SoS) (Jamshidi, 2009). Like CPS, System of Systems (SoS) can also represent the new vistas for various applications, such as aerospace, manufacturing, military and so on (Samad & Parisini, 2011). (Carlock & Fenton, 2001) define SoS as “large-scale concurrent and distributed systems that are comprised of complex systems”. The principal differences between a thing being either a “system” (e.g., a mechatronic system) or a SoS are on the nature of a system’s composition (Boardman, 2006). The multi-disciplinary nature of SoS also requires close collaboration among several disciplines (Samad & Parisini, 2011). In summary, there is a tendency towards mechatronic systems which are envisioned as building blocks for the design of system of systems.

Above discussion indicates that more and more innovative functionalities have been integrated into mechatronic systems. For example, the integration of cyber technology that makes the products Internet-enabled improves innovative services to achieve, among other things, Internet-based diagnostics, maintenance, operation, etc. in a cost-effective and efficient manner (Jazdi, 2014). As a result, from the simple automated systems in which mechanical and electronic components are combined together, through the modern smart products in which various disciplines have been involved, to CPS, IoT and SoS into which cyber world has been integrated, new functionalities have been provided and become much more integrated than before thanks to the development of technology.

Besides the functional integration discussed previously, the physical level of mechatronic systems has become increasingly integrated as well. Several levels of physical integration of mechatronic systems related to product exist (Figure 1.4). The first kind of integration is called “separated components”. In this case, components are designed separately and are just incorporated in the same system thanks to cable. The second level of integration corresponds to the concept of “joined components”. The mechanical parts will be designed in order to place the electrical and/or the electronic parts in juxtaposition with each other. Distances between components have been reduced. The third kind of integration is called “included”: electronic components are spread out into the whole system, but this kind of integration does not achieve a “real” integration. Finally, the ultimate integration level is the “merged” components: electronics is integrated as close as possible to the

SCIENTIFIC CONTEXT AND RESEARCH QUESTIONS

mechanical and electrical components. Parts are gathered in a consistent and functional manner and mechanical parts can also be used as signal transmitter. The contributions of this integration are various:

- Physical integration: spatial and weight optimisations,
- Functional integration: detection, communication, control/information processing allow the system to provide new functionalities and to be more reliable.



Figure 1.4 Different integration levels in mechatronic systems (Penas, Plateaux, Choley, & Rivière, 2009)

The development trends, as presented before, related to mechatronic systems show that the design of mechatronic systems has to integrate various disciplines. This complexity is linked to the increasing integration level and the wider range of collaborators involved. However, such integrated design is often hampered by lack of uniform interpretation of product modelling languages and terminologies, leading to rework when discrepancies are discovered (Bock, Zha, Suh, & Lee, 2010). Designers geographically and organizationally distributed in global economy worsen these problems (Shen & Barthès, 2008).

Product modelling yields product data models drawn from a wide variety of source as its results. A product model database which is generated by product modelling approach during the product development process should be able to support all the concerns of the whole product life cycle. Here the term product development process refers to those stages or phases from an initial concept to a proven prototype of the product (Krause et al., 1993). Therefore the product model can be also considered as an effective support for the design of mechatronic systems. Previous work in product modelling approach can be generally divided into two parts, model-based product (or product model in the thesis) and ontological techniques (Bock et al., 2010). Product models have been used much longer than ontological techniques and numerous product models have been developed, such as ISO 10303 (STEP, STandard for the Exchange of Product model data) (ISO 10303, 1994), the National Institute of Standards and Technology (NIST) Core Product Model (CPM) and its extensions

(Fenves, Foufou, Bock, & Sriram, 2006), and Product-ProcessOrganisation model (PPO) (Noël & Roucoules, 2008). They provide engineering-oriented language and the meaning of the language is not strictly specified. Therefore the product models define a syntactic data representation (Abdul-ghafour, Ghodous, & Shariat, 2012).

Ontology is initially proposed as an important and natural means of representing real world things, or things intended for the real world (GómezPérez, Fernandez-Lopez, & Corcho, 2004). It has been recently applied to product modelling. Compared with product model, ontology defines a semantic data representation (Abdul-ghafour et al., 2012). It can give cleaner meaning to product models by interpreting them as physical things, rather than linguistic constructions such as modelling languages used by product models. Unlike the product model, the ontology is not tied to modelling languages for ease of use in the engineering community. Therefore the designers should give ontological meaning by making use of ontology languages for individual, physical objects or occurrences of behaviour when using it for product modelling (Bock et al., 2010). Efforts that combine ontology and modelling languages of product model have been made by current studies (Chungoora et al., 2013; Gil & Martin-Bautista, 2012; Sun, Fan, Shen, & Xiao, 2012).

The above introduction of the two product modelling approaches - product model and ontology introduces that both of them can be used as an effective support for the integration related to the product during the design of mechatronic systems, because they propose a natural means to help the designers to access, store, serve and reuse the design-related data during the design process. And meanwhile, the possibility of creating a standard representation of the design-related data which can be understood by the designers of different disciplines is also indicated by the nature of product model and ontology.

In this section, the integration related to the product has been presented as a great challenge for the design of mechatronic systems. Product modelling approach has been also introduced and is considered in this thesis as an effective support to the integration related to the product. However, in order to achieve the multi-discipline based integrated design of mechatronic systems, the design process has to be called for the coordination of the design teams of different disciplines. In other words, the collaboration among different expertise and disciplines during the design process of mechatronic system plays a key role to ensure that the results of their efforts are successful, especially to obtain an integrated system. Therefore, the second type of integration which has been discussed in next sub-section is related to the design process.

1.1.2.2 Integration related to the process

The integration related to the process requires a high level collaboration from both the designers and the various involved disciplines during the design process. The design of mechatronic systems calls for multi-disciplinary collaboration, the designers of which cannot master all the knowledge needed for the design, and sometimes the designers are separated in different areas and even different countries. That is the reason why collaborative and concurrent approaches for the design of mechatronic systems are so complex and so challenging. Such challenges will generally lead to a poor integration level. From the traditional sequential design (Pahl, Beitz, Feldhusen, & Grote, 2007) to the concurrent engineering (Li, Zhang, & Chen, 2001) numerous design methodologies have been proposed to organise the design process in order to achieve a high level multi-disciplinary integration during the design process of mechatronic systems. The historical development of mechatronic systems discussed in Section 1.1.1 indicates that the mechatronic systems become more and more integrated thanks to the convergence of technological diversity of from different disciplines. To keep up with such development trends of mechatronic systems, some new development methodologies have been proposed and are still under the study of academia and industry.

Agile development methods¹ were seen initially as software engineering methods in which requirements and solutions evolve through collaboration between self-organising, cross-functional teams (Highsmith, 2002). They were initially viewed as best suited to small and non-critical projects with co-located teams (Abrahamsson, Conboy, & Wang, 2009). Therefore how to apply the agile development methods to the design of mechatronic systems is still a great challenge for the designers (Bricogne, 2015). On the one hand, the design of mechatronic systems is often considered as a large-scale project and the designers are often separated in different areas. On the other hand, the design process of mechatronic systems always requires a regulated environment in which the design process is expressed and adapted by careful tailoring. The constraints of current agile development methods about small projects and colocated teams have been addressed by several research studies on agile adoption by large teams (Cao, Mohan, Xu, & Ramesh, 2004; Kahkonen, 2004) and in distributed environment (Boland & Fitzgerald, 2004; Kircher, Jain, Corsaro, & Levine, 2001). However, the regulated environments required by the design of

¹ <http://agilemanifesto.org/>

mechatronic systems and the agile development methods are often seen as fundamentally incompatible. How an agile approach can be implemented successfully in a regulated environment is still under discussion (Cawley, Wang, & Richardson, 2010; Fitzgerald, Stol, O'Sullivan, & O'Brien, 2013).

Specific methods have been proposed according to the principles of agile development. How to use them during the design process of mechatronic systems have attracted more and more attention and research. Scrum, one specific method of agile development, is originally suggested for “managing product development project” and mainly used for software development projects (Boehm, 2002). However, Scrum has been extended and some examples in literature show that design methods based on Scrum have been provided to solve the multi-disciplinary problems existing in the design of mechatronic systems (Cooke, Maarten Bonnema, & Poelman, 2012; Grimheden, 2013; Stelzmann, 2011).

Increasing the integration of mechatronic systems and decreasing time for development and cost reduction require a lean development process. Lean development is rooted in the Toyota Production System from the 1950s (Womack, Jones, & Roos, 2007). The core ideas of lean development method are to eliminate waste, achieve quality first time, and focus on problem solving (Poppendieck & Poppendieck, 2003). Studies have been carried on to apply the lean development during the design process of mechatronic systems (Elezi, Graebisch, Hellenbrand, & Lindemann, 2011; Hellenbrand & Lindemann, 2011; Jönsson, 2004).

The development trends of design processes for mechatronic systems proposed by the design methodologies discussed previously imply that the designers have been moving increasingly closer to the users (or customers) of their products. In other words, not only the collaboration of designers from different disciplines during the design process become more and more integrated, but the involvement of users (or customers) during the design process is becoming more and more important as well. The terms Co-design is proposed to describe such participatory design. Co-design, in a broader sense, refers to the creativity of designers and people not trained in design working together in the design development process (Sanders & Stappers, 2008). Various tools and methods of co-design have been proposed, such as Crowdsourcing, which is the act of taking a task traditionally performed by a designated agent and outsourcing it by making an open call to an undefined but large group of people (Howe, 2008), and Personas, which provide task scenarios describing how the users interact with the design (Long, 2009). Generally speaking, in co-design, the

boundaries of the roles are blurred. The user is granted the position of expert of his/her experience, and plays a large role in knowledge development, idea generation and concept development (Hribernik et al., 2011).

As discussed before, in order to achieve a high-level integration related to the process, the designers should focus on both the “coordination and the synchronisation of the different disciplines, specific development processes, activities, tasks and results across all fields” (Abramovici & Bellalouna, 2007) and “complex coherences and interactions between the disciplines” (Abramovici & Bellalouna, 2007). Furthermore, according to (Abramovici & Bellalouna, 2007), attention should be paid to “the comprehensive integration, configuration, change and release management across all disciplines”. The two previous sub-sections present the two types of multi-disciplinary integration during the design of mechatronic systems. The first type concerns the integration related to the product of mechatronic systems. This integration allows fulfilling more functions in more compact systems compared to a pure mechanical system or pure electronic system. The second type focuses on the integration related to the process. A well-organised collaborative and concurrent design of mechatronic system is required during the design process. By analysing the development trends of mechatronic systems, the scientific problems and the research objectives will be pointed out in next sub-section.

□ 1.2 Scientific problems and research objectives

1.2.1 Scientific problems

As depicted previously, in order to achieve the multi-disciplinary integration for the design of mechatronic systems, attention should be paid to the two types of integration: a design process with high-level multi-disciplinary collaboration and a mechatronic system with high-level functional and physical integration. The former raises “Process-based problems” while the latter brings up “Design data-related problems” (Abramovici & Bellalouna, 2007). According to (Abramovici & Bellalouna, 2007), “Process-based problems” are linked to the fact that the coordination and the synchronisation of the “different disciplines, specific development processes, activities, tasks and results across all fields is not sufficiently supported”, but also to the fact that complex “coherences and interactions between the

disciplines are considered in a late development phase”. Furthermore, “comprehensive integration, configuration, change and release management across all disciplines is little or barely supported” (Abramovici & Bellalouna, 2007).

The second kind of difficulties encountered during design of mechatronic systems, called “Design data-related problems” (Abramovici & Bellalouna, 2007), is related to the edition and management tools heterogeneity. For example, mechanical designers use Computer Aided x (CAx) applications to support the product development process. The data generated are generally stored in Mechanical Product Data Management (M-PDM) systems while electrical and electronic designers use Electrical/Electronic Engineering

Solutions (EES) to create data which are stored in Electronic/Electrical PDM (E-PDM). Software designers use development solutions to create source code that is managed thanks to Software Configuration Management (SCM) or Concurrent Versions System (CVS) systems. This heterogeneity in terms of product data, data models and data formats leads to several problems that can be summarised as no adequate multi-disciplinary integration of product data (Bricogne et al., 2010).

However, neither academia nor industry has yet provided explicit solutions to solve such two kinds of problems. Considering the two kinds of problems existing in the design of mechatronic systems, the research objectives will be presented in next subsection.

1.2.2 Research objectives

The two kinds of scientific problems related to the multi-disciplinary design of mechatronic systems discussed before reveal our research objectives in this subsection. They can be divided into two parts.

The first research objective is relevant to the design data-related problems. As discussed before, both the functionality and the physical size of mechatronic systems become increasingly integrated and compact respectively and different disciplines have been gradually integrated into the design of mechatronic systems. However, due to the heterogeneous design data coming from different disciplines during the design process, lack of uniform interpretation of design-related data often hamper such integrated design. Engineers distributed geographically and organizationally in global economies worsen these problems (Shen & Barthès, 2008). As a result, the first research objective is to provide a standard representation to link the design-related data created by different discipline during the design process. In other words, a

standard representation for the “interfaces” in mechatronic systems should be proposed. Such interfaces exist not only between sub-systems (or between sub-system and environment) designed by different disciplines, but also among the engineers who collaborate or coordinate by sharing information through formal or informal interaction. In order to achieve this objective, current product modelling approach should be further developed according to the mechatronic systems specificities.

The second research objective is relevant to the process-based problems. A new design method which can realise a better coordination and synchronisation of the different disciplines and the engineers during the design processes should be achieved. To be more precise, this new design method should not only describe the generic procedure for the design of mechatronic systems from the identification of all requirements on the total system to a uservalidated system, but also support each design phase where individual engineers can proceed and react in unforeseen situations and structure design sub-tasks.

In order to achieve the research objectives, the research questions will be pointed out in next sub-section.

□ 1.3 Research questions

As discussed before, in order to achieve a higher degree of multi-disciplinary integration for the design of mechatronic systems, the designers should focus not only on the synergistic integration for the products (i.e. integration related to the product), but also on the integration of the different involved disciplines during the design processes (i.e. integration related to the process). Therefore the main research questions addressed in this thesis are:

- Research questions related to the design process: How to achieve a multi-disciplinary and holistic process for the design of mechatronic systems?
- Research questions related to the design data: How to manage data issued from several discipline to support the multi-disciplinary integration during the design process of mechatronic system?

A series of sub-questions are listed below for each of the main research questions.

1.3.1 Research questions related to the design process

Two research sub-questions of design process can be identified hereafter:

How to support the macro level collaboration?

As mentioned by (Shetty & Kolk, 2010), mechatronic systems are often built from discipline homogeneous subsystems (mechanics, electrics/electronics and software). Concurrent engineering is a work methodology based on the parallelisation of design tasks carried out by different design teams (Sohlenius, 1992). It is of great importance as the design cost and development lead-time can be drastically reduced through the design tasks carried out at the same time (Wang, Shen, Xie, Neelamkavil, & Pardasani, 2002). Such concurrent engineering for the design of mechatronic system is carried on in a concurrent manner with a special focus on the subsystems and the interfaces among them (Wikander, Törngren, & Hanson, 2001). The macro level collaboration emphasises such discipline homogeneous collaboration from different design teams. It not only focuses on the assembly of the subsystems from different specific design disciplines, but pays special attention to the interfaces among them as well.

How to support the micro level collaboration

The design process of mechatronic system should also focus on the collaboration of the different designers, such as communication among designers, data sharing and exchange, etc. How to achieve the coordination of resources and designers has attracted increasing attention (Girard & Doumeingts, 2004).

The collaboration among the individuals is called in the thesis micro level collaboration. Traditionally, the micro level collaboration is often performed thanks to informal exchanges supported by e-mail, phone or regular meeting. In extreme collaboration, designers work physically together, i.e., at the same place, as long as they have to finish the task. Such traditional low-tech tools, like face-to-face discussion, whiteboards and flip charts, facilitate daily communication in several ways. On the one hand, designers can break out and start an instantaneous meeting as soon as a planned or unplanned issue arises. On the other hand, designers can become involved to prevent others from making uninformed decisions or immediately adapt their work to incorporate an unexpected result (Garcia, Kunz, Ekstrom, & Kiviniemi, 2004).

However, as discussed above, mechatronic systems have become more and more complex and integrated, and the designers are often geographically distributed. Therefore the low-tech collaboration tools such as face-to-face discussion are impossible for designers' daily communications. With the support of information technology, the tools and applications including graphical modelling, numerical simulations and analyses, networked support and standard product models, the

collaboration among the designers, which provides an instantaneous information sharing, can be consequently achieved. Product Data Management (PDM) systems, one example of integration tool connecting many different areas of product development, manage product-related information such as geometry, engineering drawings, project plans, product specifications, analysis results, bills of material, engineering change orders and many more (Eynard, Gallet, Nowak, & Roucoules, 2004; Xu & Liu, 2003). PDM can ensure that the right information is available to the right person at the right time and in the right form throughout the enterprise and many commercial PDM systems have been developed in recent years, such as

ENOVIA² belonging to Dassault Systèmes, Teamcenter³ belonging to Siemens, Windchill⁴ belonging to PTC and so on. But the detailed modelling method and the framework of those commercial PDM systems are seldom reported (H. Tang, Guo, Huang, Li, & Li, 2015).

1.3.2 Research questions related to the design data

Section 1.2.1 presents the challenges relevant to the design data-related problems. By analysing these challenges, two research sub-questions of design data can be identified:

How to describe the macro-level interface?

During the design process of mechatronic system, a great number of subsystems are developed by discipline-specific design teams. With the purpose of two subsystems (or components) defined by different design teams to be interconnected, there must be compatible interfaces in mechanical, electronic/electrical and software disciplines (Thramboulidis, 2005), which are called in this thesis macro level interfaces. Such interfaces describe the associations between subsystems, both to indicate how subsystems should be joined in the final product and provide high-level guidance for the disciplinespecific design teams (Bettig & Gershenson, 2010). The macro level interfaces can help the design teams to achieve the basics for integration of the subsystems. By comparing the description of macro level interface with that of macro level collaboration described previously, macro level interface envisions to be an

² <http://www.3ds.com/products-services/enovia/>

³ <http://www.plm.automation.siemens.com/>

⁴ <http://www.ptc.com/>

effective support for macro level collaboration during the design process of mechatronic design.

How to describe the micro-level interface?

The engineers need another kind of interface which allows them to exchange and share information or data during the design process of mechatronic system.

It intends to help designers to collaborate or coordinate by sharing information

through formal or informal interaction (Zha & Du, 2006). Such kind of interface, called micro level interface in this thesis, notifies the designers that their discipline-specific solutions have to be taken into account by other disciplines for their own solutions to manage conflicts between them. Because micro level interface has been built with a lot of information and included in many distributed computer systems to support the design process of mechatronic system, it fosters a better micro level collaboration. In summary, micro level interface can help engineers to have a well-organised concurrent engineering for the design process of mechatronic system, focusing on possible inconsistencies or poor integration.

1.3.3 Relationship between the two main research questions

The sub-questions of the two main research questions have been presented before. Table 1.1 shows the summary of the two main research questions and their sub-questions.

Table 1.1 Main research questions and their sub-questions

Main research questions	Sub-questions	Description
Questions related to the design process	How to support the macro level collaboration?	Discipline homogeneous collaboration which focuses on the assembly of the subsystems from different specific design disciplines and the interfaces among them
	How to support the micro level collaboration?	Collaboration of the different engineers, such as communication among designers, data sharing and exchange, design knowledge management and etc
Questions related to the design data	How to describe the macro-level interface?	Interface which describes the associations between subsystems and can help engineers to achieve the basics for integration of the subsystems

SCIENTIFIC CONTEXT AND RESEARCH QUESTIONS

	How to describe the micro-level interface?	Interface which allows engineers to exchange and share information or data from other disciplines and helps designers collaborate or coordinate through formal or informal interaction
--	--	--

In summary, design information can be classified into two categories: design process information and product information (Zha & Du, 2006). Macro level interface and micro level interface are considered as effective supports for macro level collaboration and micro level collaboration separately. Figure 1.5 shows the relationship between the sub-questions of macro-level collaboration and macro-level interface and micro-level collaboration and micro-level interface.

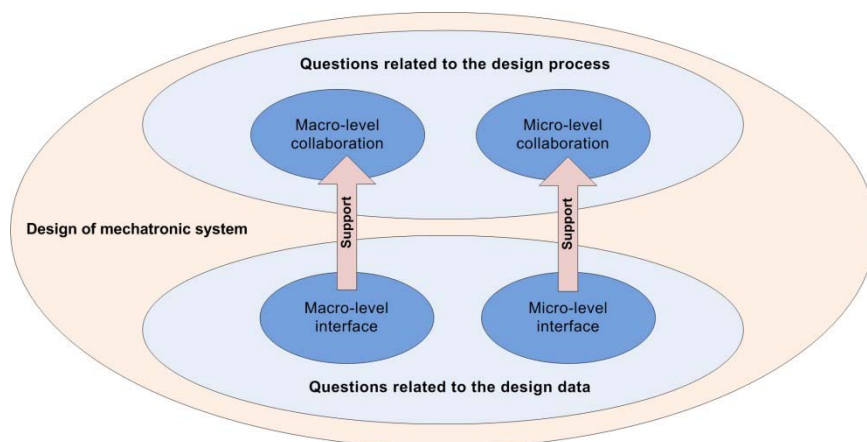


Figure 1.5 Relationship between the research questions

In order to position the research proposition relatively to current design methods and product models, next chapter will describe the state of the art followed by the research propositions.

State of the art

This chapter reviews current research works of mechatronics design. As above discussed, in order to achieve the multi-disciplinary design and integration of mechatronic systems, the design method and the product model are considered as the potential solutions to the main research questions presented in Chapter 1. In this chapter, two approaches, Product Lifecycle Management (PLM) and Systems Engineering (SE), are first reviewed. On the one hand, many design methods are considered as methodologies and supports of SE. On the other hand, various product models have been proposed to support Product Data Management (PDM) functions of PLM during the design process of mechatronic systems. Then, current research works on design methods and product models enabling mechatronic design and disciplinary integration in mechatronics are therefore presented. By analysing the evaluation results of design methods and product models, the research approaches will be introduced at the end of this chapter.

□ 2.1 Product design

In this section, two approaches dealing with multi-disciplinary collaborative design of product with high level integration are presented. The first approach is Product Lifecycle Management (PLM). It covers all activities performed along the product lifecycle (Terzi, Bouras, Dutta, Garetti, & Kiritsis, 2010). Various product models supporting PDM functions of PLM during the design process of mechatronic systems have been proposed. The second approach is System Engineering (SE), which focuses on the design and management of complex engineering systems over their life cycles (Blanchard, 2008). Based on the general systems engineering process, many design methods have been proposed. As a result, before reviewing the current studies on product models and design methods, we will firstly go over these two approaches.

2.1.1 Product Lifecycle Management

2.1.1.1 Definition of PLM

The concept of the Product Life Cycle (PLC) has been introduced since the 1950s in order to describe every phase of a product goes through, from the first initial requirement until it is retired (Stark, 2011). In the early 1980s, the Computer Aided Design (CAD) systems have appeared. Then the Engineering Data Management (EDM) and the Product Data Management (PDM) emerged in the late 1980s because the need to keep track of the increasing number of design data generated by CAD systems was recognised by the engineers in the manufacturing industries (Saaksvuori & Immonen, 2002). In order to fill the gap between the PDM and the enterprise business activities, during the 1990s, the concept of Product Lifecycle Management (PLM) was proposed. Different from the PDM system that mainly focuses on managing the product data, the PLM solution focuses on managing all the product-related knowledge throughout the different phases of the PLC (Ameri & Dutta, 2005). As defined by (Saaksvuori & Immonen, 2002), PLM is a “systematic, controlled concept for managing and developing products and related information”. It ensures “the management and the control of product process (product development, production and product marketing)” and provides “the order-delivery process, the control of product related information throughout the product life cycle, from the initial idea to the scrap yard” (Saaksvuori & Immonen, 2002).

2.1.1.2 PLM and PDM systems

PLM systems are tools that assist a company in the implementation of PLM concept (Rachuri, Fenves, Sriram, & Wang, 2005). The full PLM system functionality can be achieved by the specific components shown in Figure 2.1. These are: (1) an Information Technology (IT) Infrastructure; (2) a Product Information Modelling Architecture; (3) a Development Toolkit and Environment; and (4) a set of Business Applications. The IT infrastructure is the foundation that includes hardware, software, and Internet technologies, underlying representation and computing languages, and distributed objects and components. PLM systems form the top of the corporate software hierarchy and frequently implemented so that they depend on subsidiary systems for detailed information capture and dissemination (Rachuri et al., 2005).

Design of complex products, such as mechatronic systems, often requires the teams of designers from several disciplines and geographical distributed locations to work

together. The PDM functions of PLM are often used as an effective support enabling the collaboration during the design process of complex products. On the one hand, PDM is considered as a category of software that aims to store product data into a database. The information mentioned here includes CAD models, drawings and their associated documents (Eynard et al., 2004). On the other hand, PDM can also be considered as an integration tool connecting many different areas of product design, which ensures that the right information is available to the right person at the right time and in the right form throughout the enterprise (Xu & Liu, 2003).

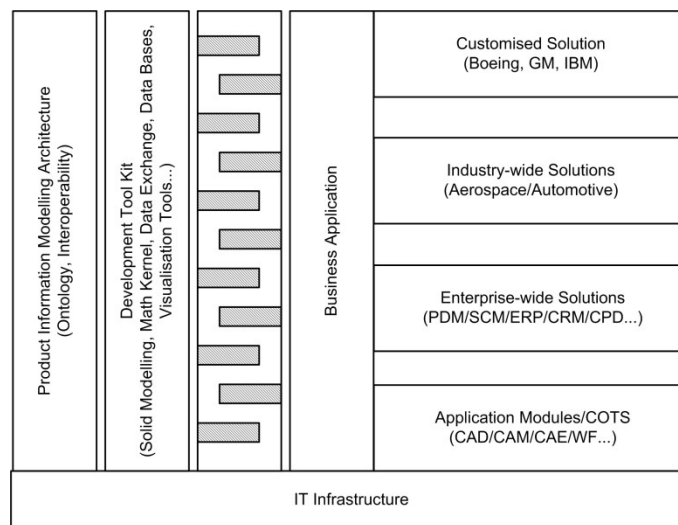


Figure 2.1 Conceptual PLM system architecture (Rachuri et al., 2005)

A typical PDM system possesses a basic set of functions and features. These may include a database, sets of user-directed functions and sets of utility functions. The database is considered as the foundation of a PDM system, so all kinds of product information can be stored in this database. In general, two types of data are stored in the data vault: the product data generated from various applications and the meta-data, which contains data about the PDM controlled information (Xu & Liu, 2003).

2.1.1.3 Benefits of PLM systems for the design of mechatronic systems

Generally speaking, PLM systems can help the designers to achieve multiple advantages in terms of collaboration. The benefits for the design of mechatronic systems can be summarised as follows:

- Achieve the multi-disciplinary collaboration: during the early design phase of mechatronic systems, the PDM functions of a PLM system can help the designers from different disciplines to collaborate and identify the new

product and others currently in production. In other words, the PLM systems can lead to a collaborative development process of new product as well as improvements on existing products (Miller, 1998).

- Reduce the complexity of accessing the information: the PLM systems can simplify many day-to-day user operations by managing and automating routine tasks, such as searching for drawings, tracking approvals, and completing status reports (Liu & Xu, 2001). This improvement will greatly reduce the complexity of accessing the information during the design process of mechatronic systems.
- Reduce the lead-time and the cost for the design of mechatronic systems: because of the increasing complexity of information during the design process of mechatronic systems, the lead-time and the cost can be increased dramatically. The PLM systems help the designers to access, store, serve and reuse the information more effectively and efficiently, so the lead-time and the cost for the design of mechatronic systems can be greatly reduced (X. Tang & Yun, 2008).
- The design process of mechatronic systems indicates that a global control of the business process through all its lifecycle should be taken into consideration. In such multi-disciplinary context, how to manage the different evolutions of components in mechatronic systems becomes one of the main difficulties. In order to overcome this difficulty, it is necessary to keep the coherence and the information integrity as well as to allow a real collaboration between the different designers throughout the life cycle of the product. PLM systems seem the most appropriate support to help the designers to achieve this objective (Abid, Pernelle, Noterman, Campagne, & Amar, 2014). The first approach of product design - Product Lifecycle Management (PLM) has been presented in this section. PLM can help the designers achieve the collaborative design process of products because it covers all activities performed along the product lifecycle. The product model can be used as an effective support to the PDM function, one of the most important elements in the implementation of PLM. However, a product model implemented in the PDM function of PLM systems which can support the multi-discipline based integrated design of mechatronic system has not been fully developed (Zheng, Bricogne, Le Duigou, & Eynard, 2014a).

In this section the first approach dealing with multi-disciplinary collaborative design of product has been gone over. The product model implemented in the PDM function

of PLM systems seems to be a potential solution to the design data-related problems. Existing product models will be then surveyed in Section 2.2.2. Next section presents the second approach of products design by focusing on SE. Many design methods have been proposed based on the general system engineering process, which are all considered as the potential solutions to solve the process-based problems.

2.1.2 Systems Engineering

2.1.2.1 Definition of systems engineering

According to different background and personal experiences of engineers, SE may be defined in a numbers of ways. However, there is a common theme that deals with a top-down process, which is lifecycle oriented, involving the integration of functions, activities and organisations (Blanchard, 2008). The International Council on Systems Engineering (INCOSE) defines systems engineering as “Systems engineering is an interdisciplinary approach and means to enable the realisation of successful systems” (INCOSE, 2015). It focuses on “defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem”. As shown by Figure 2.2, the fundamental SE activities are Requirements Analysis, Functional Analysis and Allocation, and Design Synthesis - all balanced by techniques and tools collectively (Leonard, 1999). SE considers both the business and technical needs of all customers with the goal of providing a quality product that meets the user needs.

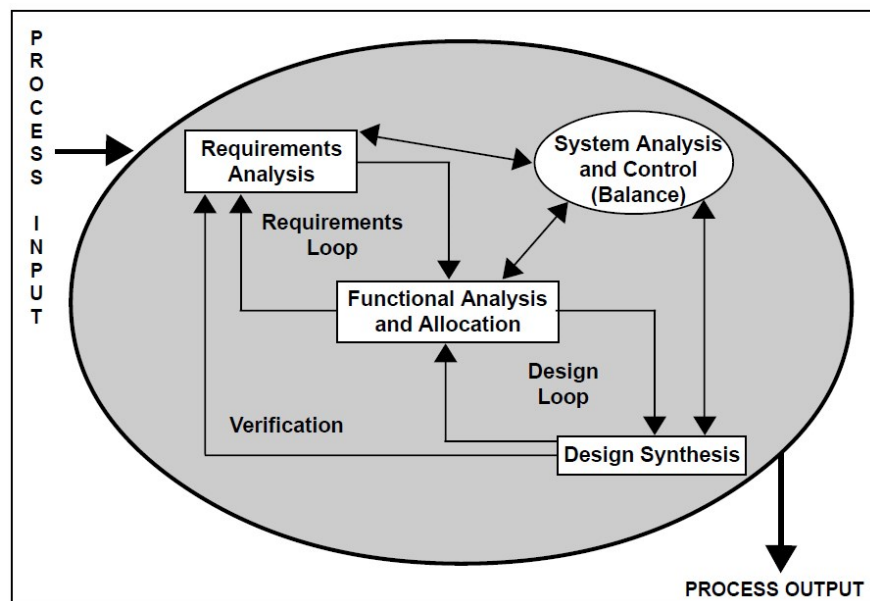


Figure 2.2 Systems engineering process (Leonard, 1999)

Generally speaking, SE describes a view of all the processes that have to be performed in order to engineer and develop a complex system, considering its complete lifecycle (Lardeur, 2006). Some international standards have been developed to describe the systems engineering process.

2.1.2.2 ANSI/EIA 632

The ANSI/EIA 632 was developed by the G47 Systems Engineering Committee of the Electronic Industries Association (EIA) and titled “Systems Engineering” in August 1998 and the full standard was expanded in scope to include all the technical processes for engineering a system (Martin, 1998). It is intended to be a higher level abstraction of the activities and tasks found in the intermediate standard version plus those other technical activities and tasks deemed to be essential to the engineering of a system (Martin, 1998). The focus of this standard is on conceptualizing, creating and realizing a system and the products that make up a system. Figure 2.3 shows the thirteen processes directly related to the technical aspects of engineering systems. It defines representative tasks and the expected outcomes associated with each one. There is not one process but a series of processes, in groups, with loops among them (Martin, 2000).

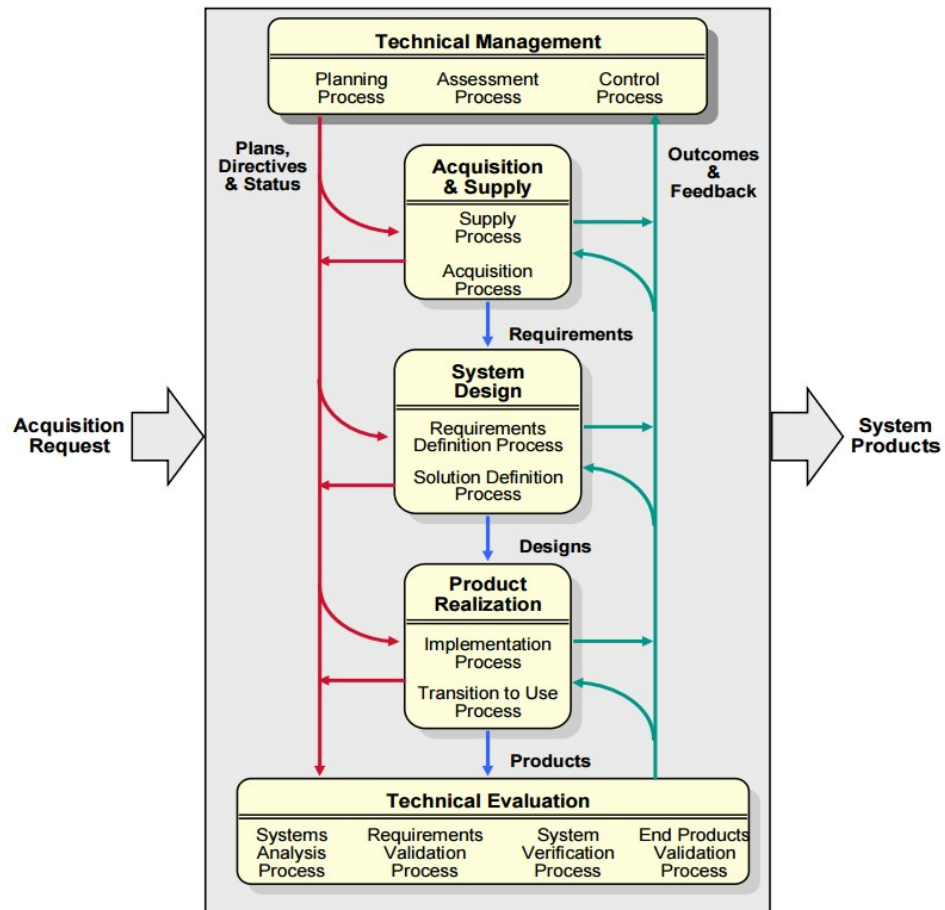


Figure 2.3 SE described by ANSI/EIA632 (Martin, 2000)

2.1.2.3 IEEE 1220

The second standard reviewed here is IEEE 1220, entitled “Application and Management of the Systems Engineering”. It provides the next-level-of-detail description of the systems engineering processes defined in EIA 632. Figure 2.4 outlines the systems engineering process proposed by IEEE 1220. It generally includes six processes as follows: Requirements Analysis, Requirements Validation, Functional Analysis, Functional Verification, Synthesis and Physical Verification. These processes are linked together via Control Processes consisting of Data Management; Configuration Management; Interface Management; Risk Management and Performancebased Progress Measurements (Doran, 2004).

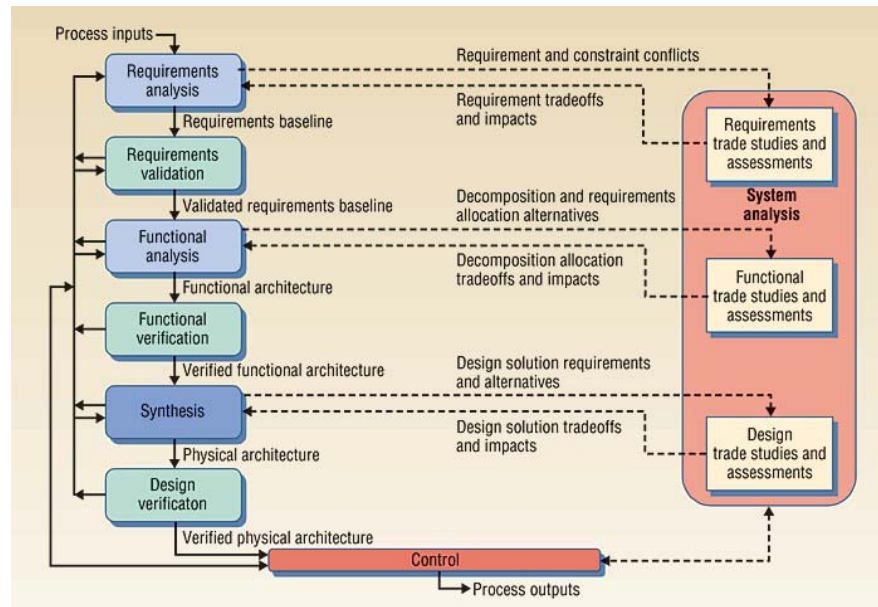


Figure 2.4 SE described by IEEE 1220 (Doran, 2004)

2.1.2.4 ISO/IEC 15288

The last international standard presented here is ISO/IEC 15288, which was proposed to ISO/IEC JTC1 by the U. S. National Body in 1994. From then on, ISO/IEC 15288 has been under conceptualization and development. It was ratified as an International standard in June 2002, and was published in November 2002 (Arnold & Lawson, 2004). ISO/IEC 15288 is applied to the full lifecycle of systems, including conception, development, production, utilization, support and retirements of system and to acquisition and supply (ISO/IEC 15288, 2002). Figure 2.5 shows the detailed process of ISO/IEC 15288, including Agreement Processes, Enterprise Processes, Project Processes and Technical Processes.

2.1.2.5 Synthesis of SE standards

Three SE standards have been reviewed above indicate they differ primarily in their depth and breadth of coverage. ISO/IEC 15288 is developed to cover from the beginning of design for complex system to the end of its service, but it only describes the general processes and no details of each process are involved. Therefore it has the greatest breadth but the least depth of coverage. IEEE 1220 only focuses on the systems' development and defines a systems engineering process, but the process is described more at the task or application level. Thus it has the greatest depth but the least breadth. ANSI/EIA 632 falls between ISO/IEC 15288 and IEEE 1220, it defines

the set of requirements for engineering a system and its level of details is below the process requirements described in IEEE 1220.

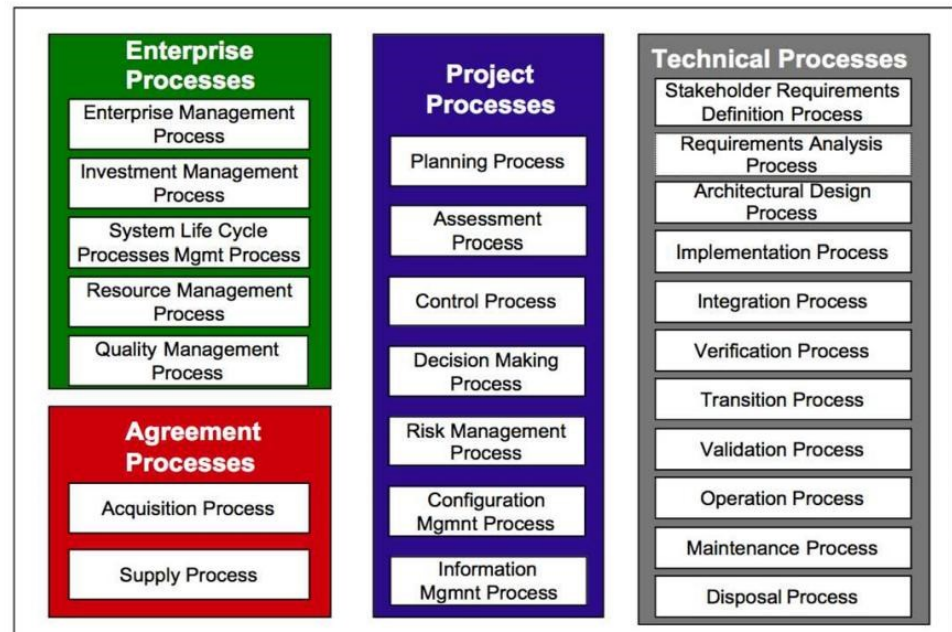


Figure 2.5 ISO/IEC 15288 Process Model (ISO/IEC 15288, 2002)

2.1.2.6 Benefits of SE for the design of mechatronic systems

SE focuses on defining customers' needs and required functionality early in the development cycle, so it brings about some benefits for the design of mechatronic systems as follows:

- The designers need an overall point of view for the whole mechatronic system during the early design phases. The system engineering views the system as a whole. It does not focus on the internal details of how one sub-system is to accomplish its object. This systems viewpoint means that the SE deals with the relationships of the system designed to its super-system (environment) and sub-system (Kapurch, 2010).
- SE can help the designers to ensure the traceability and consistency between each phase during the early design phases. For example, each function identified should be traceable back to a requirement, and each sub-system defined in the basic structure of a system must meet at least one functional requirement (Leonard, 1999).
- The architecture of mechatronic systems becomes more complex with the introduction of evolving new disciplines. Such complex systems always

require a multi-disciplinary design process. SE provides the multi-disciplinary effort required throughout the system design process to ensure that all design objectives are met in an effective manner (Blanchard, 2008).

In this section the SE has been introduced, the second approach for multidisciplinary collaborative design of product. Many design methods have been proposed based on the general system engineering process, but seldom of them can fully support the multi-disciplinary collaborative activities during the design process of mechatronic systems (Zheng, Le Duigou, Bricogne, &

Eynard, 2013). Existing design methods will be then surveyed in Section 2.2.1. By now, two approaches dealing with multi-disciplinary collaborative design of product with high level integration have been previously reviewed. A synthesis of the two product design approaches will be made in next subsection.

2.1.3 Synthesis of two product design approaches: PLM and SE

What have been reviewed imply that PLM and SE can be both used during the development process of mechatronic systems. However, the PLM concerns the entire lifecycle of systems, while the SE mainly focuses on system development phases (Bricogne, 2015).

Moreover, the systems engineering process is a top-down comprehensive, iterative and recursive problem solving process, applied sequentially through all stage of development (Leonard, 1999). On the contrary, the PLM structure all the information and data by making use of numerous multi-disciplinary models while maintaining the consistency among the models. Such approach is considered as “bottom-up” (Bricogne, 2015).

Although the scope and the way of organisation between the PLM and the SE are different, combining the two approaches to realize a systematic and synergetic approach is still possible. For example, the information or data developed during the systems engineering process can be collected in a standard form and integrated directly into the PDM functions of PLM.

This sub-section has reviewed the two product design approaches. These two approaches have been widely applied in nowadays design of mechatronic systems. Many design methods are used for SE, and various product models which support PDM functions of PLM during the design process of mechatronic systems have been proposed. In next sub-section the design methods for existing design of mechatronic systems will be reviewed.

□ 2.2 Design methods for mechatronic engineering

As above reviewed, SE has been proposed as a multi-disciplinary approach to enable the realisation of successful system since the late 1950s and early 1960s. Since 1970s, some system design methods, such as waterfall model (Royce, 1970), spiral model (Boehm, 1988) and V-model (Forsberg & Mooz, 1998) were widely used for design of complex systems. However, a design method specially adapted for design of mechatronic system was not put forward at that time. After the 1980s, the use of microcomputer technology and software determined functions were integrated in mechatronic system (Isermann, 2007). The continuously growing complexity of mechatronic system requires a more integrated design than ever. Therefore numerous mechatronic design methods have emerged to meet the need of collaboration during the design process of mechatronic systems. Derived from approaches such as the traditional sequential design (Pahl et al., 2007), the concurrent engineering (Li et al., 2001) or the much recent lean product development (Gautam & Singh, 2008), design methods have been adopted for mechatronic engineering, but these design methods still remain poor to support the technology integration and multi-disciplinary perspectives in mechatronic design. A non-exhaustive list of design methods is presented hereafter.

2.2.1 Sequential design process

The traditional approach for the design of mechatronic system is called sequential design process. In this design process, the main concerns of the mechanical view are reliability and technical performance of the system. The control view of the system is then designed and added to provide additional performance or reliability and also to correct undetected errors in the design. As the design steps occur sequentially, this approach is called sequential design model (Shetty & Kolk, 2010).

The principle of the sequential design process is that each new design task must be started when the previous one has been finished (Shetty & Kolk, 2010). For example, the mechanical design has to be “frozen” before proceeding to the design of control software (Alvarez Cabrera et al., 2010).

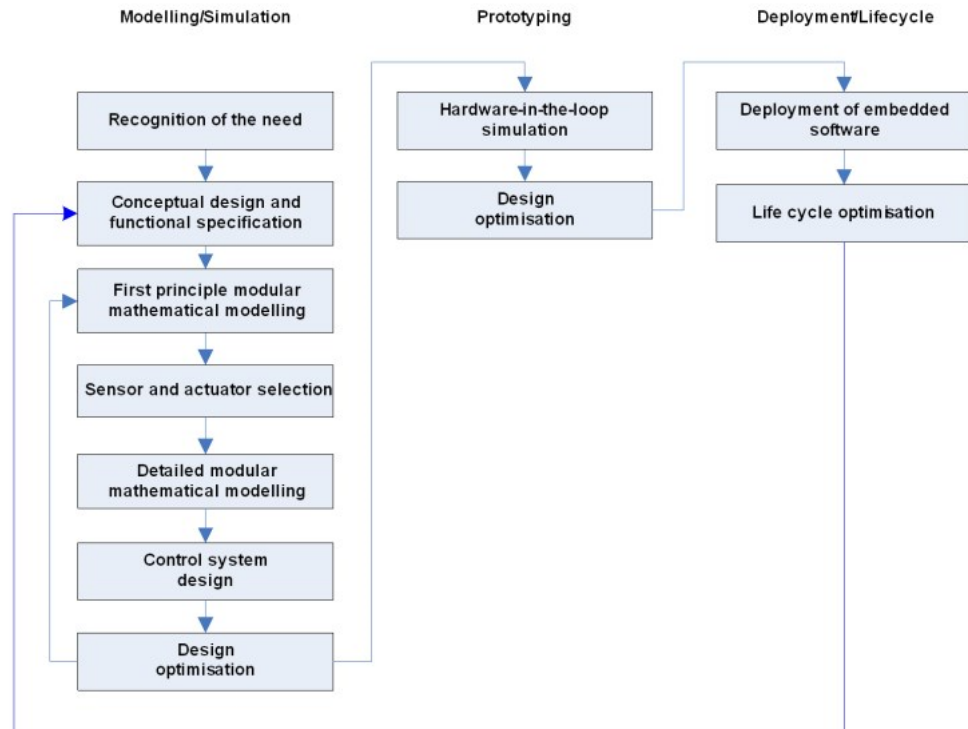


Figure 2.6 Sequential design process (Alvarez Cabrera et al., 2010)

Obviously, the sequential design process can help the executive manager to have a global view about the whole design. However, it is not suitable for modern industrial company any longer. First, the whole duration of the design process is very long since the design in each discipline has to be carried out one after another. Consequently, this approach usually does not lead to optimal overall behaviour. Second, the software plays a key role for the system's performance, so it must be considered during the whole design process. As the software design is often executed as the last task in the sequential design process, this process cannot reflect the importance of software in modern mechatronic design. Figure 2.6 shows that there are explicit links between subsystems (sensor and actuator, detailed modules, control system and etc.) during the design process. So the macro level collaboration can be performed in such design method. But it does not provide an effective support for the collaboration among different engineers. So the micro level collaboration has not been developed in this design method.

Thus, the sequential process leads to negative effects on further developments of the mechatronic systems. In order to solve the problems brought by sequential design process, several design approaches which allow concurrent design have been put forward. V-model, for instance, will be presented in the next section.

2.2.2 V-model and its variants

The V-model presents a general flow for the product development process. It starts with clarification of user's requirements. Once the requirements have been taken into account, they are then placed under project control (upper-left) and the V-model will end with a user-validated system (upper right). In order to arrive to the final product, each stage of the product definition should be tested (Forsberg & Mooz, 1998). V-model and its variants will be reviewed hereafter.

2.2.2.1 V-model

In the 1980s, V-model was widely used for SE. It provides a general flow for the product development process to the designers. Figure 2.7 shows the development process of V-model. During the High-level Design and Detailed Design phases on the axis of "Decomposition and Definition", subsystems of the system are identified and decomposed further into component. Requirements are allocated to the system components and interfaces are specified in detail. Therefore the design tasks for different subsystems can be executed in parallel. The main purpose of the axis of integration and recomposition is to validate each corresponding stage on the axis of decomposition and definition (Forsberg & Mooz, 1998).

The V-model defines an integrated design process, and a concurrent engineering approach has been achieved in this model. During the phase of implementation, the V-model simply divides the mechatronic system into software and hardware, ignoring the fact that the mechatronic system results of the combination of mechanics, electronics and software disciplines. For that reason the macro level collaboration is partially performed in V-model. The collaboration of different engineers has not been mentioned in this design method.

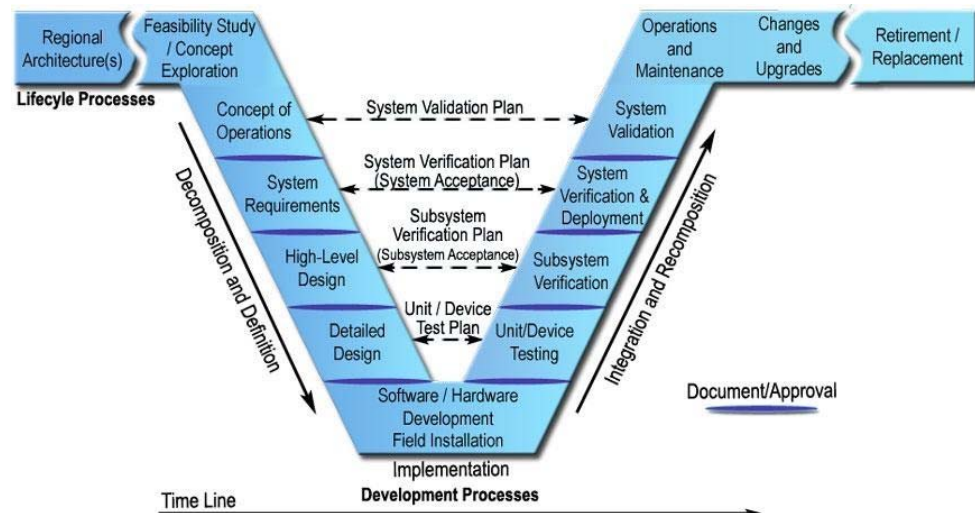


Figure 2.7 V-model (Department of Transportation, 2007)

Some variants of V-model which uses V-model as the macro level have been specially put forward for the design of mechatronic systems. The VDI guideline 2206 is an example of this V-model based method.

2.2.2.2 VDI guideline 2206

The VDI guideline 2206 is a functional modelling methodology based on the V-model. The functional modelling methodology means that different methods are used to define a model of any system by capturing and processing the information about its purpose and the functions of its components to fulfil the purpose (Koch, Sprowitz, & Ströhla, 2006). The VDI guideline 2206 (VDI 2206, 2003) is developed and standardised by a VDI committee, a German engineers association. Different from the VDI/VDE 2422 used for the design of mechanical and electrical components separately, the VDI 2206 provides the first neutral guideline for the design of mechatronic system (Fotso, Wasgint, & Achim, 2012). The VDI guideline 2206 provides a useful frame for designing any kind of mechatronic system. It consists essentially of three elements (Bathelt, Jonsson, Bacs, Dierssen, & Meier, 2005):

- The V-model on the macro-level
- A general problem-solving cycle on the micro-level
- Predefined process modules for handling recurrent working steps in the development of mechatronic systems.

The VDI guideline 2206 divides mechatronic system design into four major phases, called “system design”, “domain-specific design”, “system integration” and “assurance of properties” (Figure 2.8).

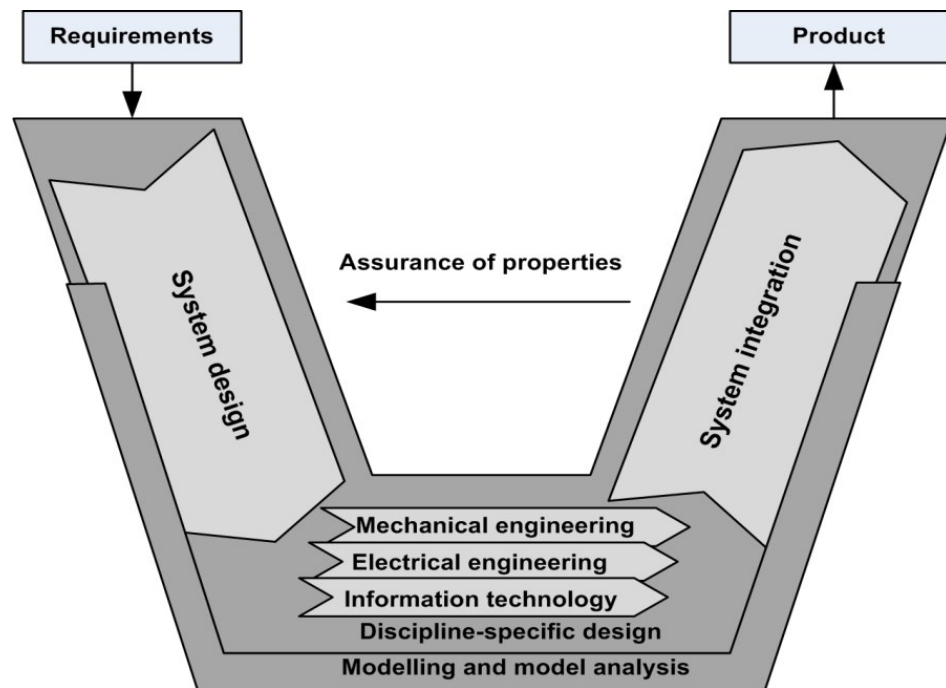


Figure 2.8 VDI guideline 2206 (VDI 2206, 2003)

The goal during the system design phase is to define a cross-discipline solution concept for the system. In this phase the overall function of the system will be divided into sub-functions.

The domain-specific design phase can be regarded as several parallel smaller design tasks. The results from the discipline-specific design are integrated to the complete mechatronic system in the phase of system integration. The purpose of the assurance of properties phase is to make sure that the results of the system integration fulfil the principal purposes defined during the phase of system design. If the system does not realise the functionalities required by the customers, the design process should be repeated.

The modelling and model analysis lasts from the system design phase to the system integration phase. During process of modelling and model analysis, modelling technique and CAx applications will be used. In order to meet the some special requirements of complex mechatronic system, several variants of VDI 2206 have been proposed. A product development process which focuses on the degree of mechatronic product maturity is proposed based on the principle of VDI 2206. In this variant, the mechatronic product is generally not produced within one macro cycle of V-model, but within many macro cycles as a continuous macro cycle. At the end of each macro cycle, a product with an increasing maturity, such as laboratory specimen, functional specimen and pilot-run product will be produced (Vasi & Lazarevic, 2008).

STATE OF ART

The VDI 2206 provides a practice-oriented guideline for design of mechatronic system. Compared with the V-model, it unifies the discipline-specific design more systematically (mechanical engineering, electrical engineering and information technology), but the specific design activities in every design phases have not been specified.

2.2.2.3 RFLP method

The RFLP approach is a specific V-model derived method particularly adapted to design of mechatronic system. It is supported by the 3DEXPERIENCE Platform⁵ software and can therefore be considered as a commercial approach.

In this method, the descending branch of V-model is divided into 4 views: Requirement engineering view, Functional view, Logical view and Physical view (Kleiner & Kramer, 2013).

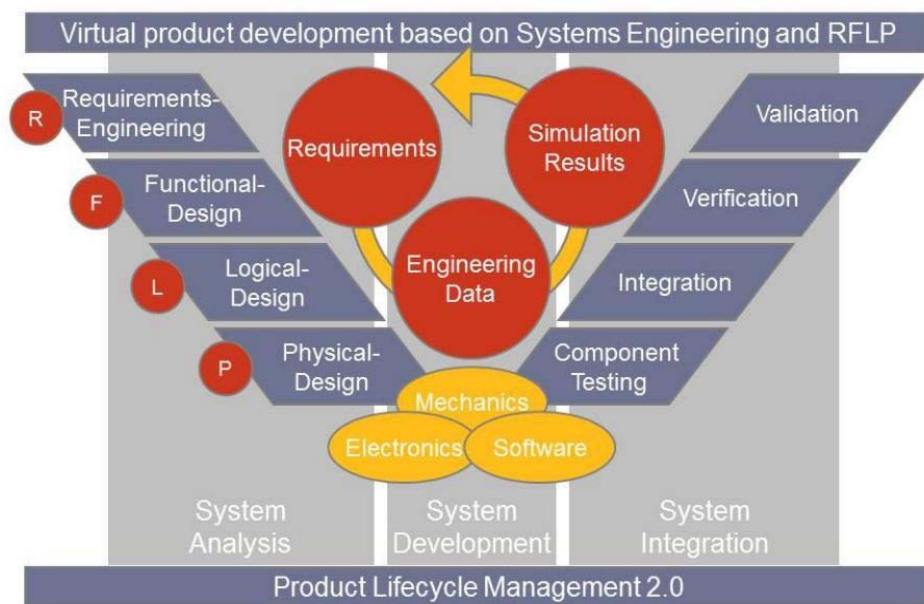


Figure 2.9 Development process based on SE and RFLP (Kleiner & Kramer, 2013)

⁵ <http://www.3ds.com/products-services/3dexperience/>

In the requirement engineering view, users' requirements are clarified. These requirements can be described according to the APTE⁶ method or the requirement diagram in SysML language⁷.

In the functional view, main functions of the mechatronic system are specified.

The SysML Block Definition Diagram (BDD) and Internal Block Diagram (IBD) or CATIA system workbenches can be used as the modelling tools in order to build the functional view.

In the logical view, the logical architecture of mechatronic system will be defined. Multi-disciplinary tools to model and carry out numerical analysis, such as Modelica⁸, Matlab/Simulink⁹ and bondgraphs¹⁰ can be used for the logical modelling of mechatronic system.

In the physical view, the components, the geometric of product (Mechanical Computer Aided Design, M-CAD), the schematic definition (Electronic/Electrical Computer Aided Design, E-CAD) and the source code (software) will be created. The 3D CAD applications and the analysis software, such as SIMULIA¹¹ can be used. In this view, the difficulty of multidisciplinary simulation is related to issues of interoperability between design and analysis, which will lead to difficulties to ensure multi-disciplinary optimisation (Lefèvre, Charles, Bosch-Mauchand, Eynard, & Padiolleau, 2014).

Nowadays, the RFLP method has been integrated into CATIA/ENOVIA v6. This CAx/PDM system provides functionalities for storing, sharing and exchanging certain types of data and information among the engineers of different disciplines, such as the data of M-CAD and E-CAD (Beier, Figge, Müller, Rothenburg, & Stark, 2013), but how to integrate the software source code remains a challenge.

VDI guideline 2206 and RFLP method have been presented in the sub-section above as two variants of V-model. Besides these two design methods, other variants have been proposed and will be presented in the next sub-section.

⁶ <http://cabinet-apte.fr/>

⁷ <http://www.omg.sysml.org/>

⁸ <https://www.modelica.org>

⁹ www.mathworks.com

¹⁰ <http://www.bondgraph.org>

¹¹ www.3ds.com/simulia

2.2.2.4 Other variants of V-model

A mechatronic system controlled by a PLC (Programmable Logic Controller) can be developed by another variant of VDI 2206. In this variant, because the information technology (developing the software for the PLC) and mechanical engineering (using CAD to design the geometry) has been identified as the major engineering domains for such mechatronic system, the domain

“electrical engineering” is neglected (Bathelt et al., 2005). (Casner, Renaud, Houssin, & Knittel, 2012) add an optimisation phase in V-model, which can optimise the prototype model of one mechatronic system. Some design methods which possess several passes of V-models have been proposed because a single V-model is understood as a generic procedure pattern and a complex mechatronic product will normally not be finished within one macro cycle (Gausemeier & Moehring, 2003). (Vasi & Lazarevic, 2008) believe that the product maturity, e.g. laboratory specimen, functional specimen and pilotrun product, should be taken into consideration during the design process, thus several V-models should be adopted in order to represent the product maturity. (Hofmann, Kopp, & Bertsche, 2010) also insist that a number of macro levels should be required for the design of complex mechatronic systems. They propose an additional V-model to represent the reliability information flow during the design process. (Gausemeier, Dumitrescu, Kahl, & Nordsiek, 2011) propose a 3-cycle-model. In this model, three V-models are used to represent the principle solutions, virtual product development and the virtual production process respectively. A W-model based on the V-model is proposed for the development of mechatronic systems. Two V-models are linked together to represent five design phases: “System analyzing”, “Specific solutions and dependency analysis”, “Virtual system integration”, “Model analysis and detailed development” and “System integration”. Central element is “Virtual system integration” defining the name giving “W”-shape (Barbieri, Fantuzzi, & Borsari, 2014; Nattermann & Anderl, 2013).

In summary, the V-model and its variants bring forth great benefits for proposing an effective way of representing a macro level collaboration for the design of complex systems. All the methods based on the V-model begin to pay a special attention to the collaboration of different design teams during the design process, but not all of them cover all the disciplines for the design of mechatronic systems (Bathelt et al., 2005; Kleiner & Kramer, 2013). Moreover, only RFLP method provides the possible solutions (i.e. modelling tools) for every design phase, other V-model based design methods seldom or never pay attention on the micro level collaboration in which

individual designers can structure design sub-tasks and proceed and react in unforeseen situations (Barbieri et al., 2014; Gausemeier et al., 2011; Gausemeier & Moehring, 2003; Hofmann et al., 2010; Nattermann & Anderl, 2013; Vasi & Lazarevic, 2008; VDI 2206, 2003).

Considering the need for micro level collaboration, the hierarchical design model will be introduced in next sub-section. Next sub-section will review the state of art of the hierarchical design model. The hierarchical structure proposed by hierarchical design model can be used as an effective support for the systems engineering process (including the requirements design, the functional design and the architectural design).

2.2.3 Hierarchical design model

Hierarchical design model helps the designers to describe product models from different viewpoints and guarantee the consistency of these models in the overall product development process (Hehenberger, 2014).

In the requirements specification phase, the requirements of a new mechatronic system should be analysed and a requirements specification (i.e. requirements list) should be determined in this phase. However, identification of these requirements is often difficult. (Seyff et al., 2009) propose a promising approach by applying use cases that incorporate possible scenarios to identify the product requirements. The initial requirements spawn further subrequirements, thus creating a hierarchy of requirements. Therefore a hierarchical structure is necessary to specify the requirements.

In the functional modelling phase, the mechatronic system's overall function, its most important sub-functions and their interactions should be determined, which leads to a functional structure. This functional structure can be organized hierarchically in order to describe the different levels of abstraction. Stone and Wood present a functional basis for design (Stone & Wood, 2000). They propose that functional modelling of a device is an important step in the design process in which the focus is on the flows of material, energy and signals. (Pahl et al., 2007) propose a method to show how a function structure can be developed by decomposing an overall function into sub-functions. In the architectural definition phase, the system's architecture is formed by grouping the functions which are already collected during the functional design phase. The system's structure can be decomposed hierarchically into subsystems, components and the interfaces among them. (van Beek, Erden, & Tomiyama, 2010)

propose a method based on the FBS (Function–Behaviour– State) modelling. In this method, the three steps of decomposition into elements, identification of the relations between the elements, and clustering of the elements into modules are realized.

The literature review of the hierarchical structures in the requirement specification phase, the functional modelling phase and the architectural definition phase shows that the hierarchical design model can be used as an effective support for the micro level collaboration. The consistency among different design phases can be also ensured by multi-disciplinary interface model.

2.2.4 Assessment of existing design methods for mechatronic engineering

A non-exhaustive list of design methods for mechatronic engineering has been presented in previous sections.

The survey shows that the design of mechatronic systems requires a high level of integration. Due to the increasing integration degree for the design of mechatronic systems, a complex mechatronic system is often broken down into simpler subsystems or components. Meanwhile, the complex design project calls for the resources management and coordination of project team members in order to be successful. Hence, the collaboration among different expertise and disciplines during the design process of mechatronic system plays a key role to ensure that the results of their efforts are successful, especially to obtain an integrated system.

Two kinds of research questions are proposed in Chapter 1. According to the research questions, two criteria can be proposed as follows:

- Can the existing design methods realise the macro level collaboration?
- Can the existing design methods realise the micro level collaboration? The

macro level collaboration emphasises the discipline homogeneous collaboration. While the micro level collaboration focuses on the collaboration of individuals, in other words, the interaction between projects team members. Two criteria will be used to evaluate existing design methods and Table 2.1 shows the details of the evaluation for each design method which have been surveyed in previous sections.

Table 2.1 Details of the evaluation for design methods

Design method	Macro level collaboration	Micro level collaboration
Sequential design process	There are explicit links between subsystems.	It does not support the collaboration of different engineers.

V-model	During the phase of Implementation, the V-model simply divides the mechatronic system into software and hardware.	It does not support the collaboration of different engineers.
VDI2206	VDI 2206 unifies the disciplinespecific design systematically.	It does not support the collaboration of different engineers.
RFLP method	Multi-domain modelling methods are used to support the macro level collaboration in the logic view.	In the physical view, the geometric definition of product, the schematic definition and the source code will be created, but the exchange of information between software design and other disciplines remains a challenge and should be further developed.
Hierarchical design model	The hierarchical design model cannot fully support the disciplinespecific design to achieve the multi-disciplinary collaboration.	The hierarchical structures can be used to support the early design phases, such as the requirement specification phase, the functional modelling phase and the architectural definition phase.

Table 2.2 shows the assessment result of the design methods according to the above proposed criteria. There exist explicit links between the disciplinespecific components in the sequential design process and VDI 2206. So these two methods can fully support the macro level collaboration. However, only the RFLP method and hierarchical design method partially support the micro level collaboration during the design process.

Table 2.2 Assessment of the design methods regarding needs of multi-disciplinary collaboration

Design method	Macro level collaboration	Micro level collaboration
Sequential design model	Yes	No
V-model	Partial	No
VDI 2206	Yes	No
RFLP	Partial	Partial
Hierarchical design model	Partial	Partial

Design methods which help and guide the engineers in the development of mechatronic system have been previously discussed. They mainly focus on the “process-based problems”. However, not all these methods can both support the macro level and micro level collaboration. As discussed at the end of Chapter 1, the design data instantiated in the product model can be used as an effect support to the existing design methods during the development process of mechatronic systems. Next sub-section will discuss the potential solutions to the design data-related problems proposed by current research works.

□ 2.3 Product models for mechatronic engineering

Another concern of this chapter is the “design data-related problems”. Traditionally, product models are used to address this kind of issues. Some of current studies on product models begin to concern the links between the design process or organisational models and the product models. Hence, the product models are also considered as effective supports for the “process-based problems”. The main objective of product model is to support PDM functions of PLM throughout the whole product lifecycle. Product model includes all the information that can be accessed, stored, served and reused by stakeholders throughout the entire product lifecycle (Eynard et al., 2004; Rachuri et al., 2005; X. Tang & Yun, 2008). STEP (ISO 10303, 1994), CPM (Fenves et al., 2006) and PPO (Noël & Roucoules, 2008), as three main product models, will be hereafter presented.

2.3.1 STEP (STandard for the Exchange of Product)

STandard for the Exchange of Product model data (STEP) is actually a series of standards, known as ISO 10303 developed by experts worldwide (ISO 10303, 1994). Its scope is much broader than that of other existing CAD data exchange formats, such as Initial Graphics Exchange Specification (IGES) which was developed primarily for the exchange of pure geometric data between CAD applications (Kemmerer, 1999). STEP is intended to handle a much wider range of product-related data covering the entire life-cycle of a product (Pratt, 2001).

As the area of STEP application is extremely broad, it is issued in numerous sections, identified as Parts. The Parts known as APs (Application Protocol) define the scope, context and information requirements of applications (Gu & Chan, 1995; Smith, 2002). STEP has developed more than forty standard APs for product data representation. They reflect the consolidated expertise of major industries for more than twenty years, covering the principal product data management areas for the main industries (Jardim-Goncalves, Figay, & Steiger-Garção, 2006). In other words, the APs are specific data models based on STEP standard covering the entire lifecycle of a product or /and a certain industrial domain. The STEP APs can be roughly grouped into the three main areas: design, manufacturing and life cycle support.

Nowadays, the STEP APs are widely used in mechanical design domain, such as AP 203, AP 209 and AP 214. Some APs related to electronic/electrical design are also proposed. However, an AP which can systematically support the whole design process of mechatronic system has not been fully developed. The STEP APs which can be used for design of mechatronic system will be introduced in more detail.

STEP AP233 (ISO10303-233, 2012) describes the key product data and information for SE that must be exchanged between dissimilar applications for requirements engineering and for systems modelling and simulation (Lefèvre et al., 2014). Industries that can benefit from using AP233 are automotive, aerospace, shipbuilding, consumer goods electronics, and others with complex products and processes. In AP 233, a multi-disciplinary “interface connector” is defined as the term for one part of a system that interacts with other parts or the environment, and the interface connection as the link between connectors, but no more details of the interface connector and interface connection are provided by AP 233 (Sellgren, Törngren, Malvius, & Biehl, 2009). AP 239 provides an integration and exchange capability for product life cycle support data (Paviot, Cheutet, & Lamouri, 2011). Besides AP 233 and AP 239, other APs related to the different expert knowledge of mechatronic system have been proposed. AP 210 (ISO10303-210, 2011) describes the requirements for the design of electrical printed circuit assemblies (PCA). AP 214 (ISO10303-214, 2010) specifies

the exchange of information between various applications which support the automotive mechanical design process, but it only focuses on the vehicle development process. According to the research questions proposed in Chapter 1, the micro level interface helps designers to collaborate or coordinate by sharing information through formal or informal interaction. STEP AP 239 not only integrates the information for defining a complex product and its support solution, but it also represents the planning, the scheduling of tasks and the management of subsequent work in order to help the designers to achieve a well-organised concurrent engineering. However, it remains very generic to support design of mechatronic system and some characteristics and parameters of mechatronic system have not been integrated in this data model. AP 214 specifies the interfaces between various CAX applications which support the automotive mechanical design process. STEP AP 210 describes the information needed for the design of electrical printed circuit assemblies. As to the micro level interface, STEP is a powerful standard which supports the exchange of geometric data between CAD applications. It focuses on the electronic/electrical discipline and mechanical discipline but not in an integrated perspective of both disciplines. It does not provide an effective interface to fully support the data exchange in software discipline.

In this section, the STEP data model and its Application Protocols have been discussed. In the next section, the Core Product Model (CPM), another standard product model, will be presented.

2.3.2 CPM (Core Product Model)

CPM, an abstract model with generic semantics, initially developed at NIST (National Institute of Standards and Technology), can support the full range of PLM information (Fenves et al., 2006).

CPM is based on two principles. First, the key object in the CPM is the Artifact. Artifact represents a distinct entity in a product, whether that entity is a component, a part, a subassembly or an assembly. Second, the artefact aggregates three objects representing the artifact's principal aspects: function, form and behaviour. CPM consists of two sets of classes, called object and relationship classes (Fenves et al., 2006). The two sets of classes are equivalent to the Unified Modelling Language (UML) terms of class and association class, respectively (Booch, Rumbaugh, & Jacobson, 1998). A UML class diagram of the CPM data model is shown on Figure 2.10.

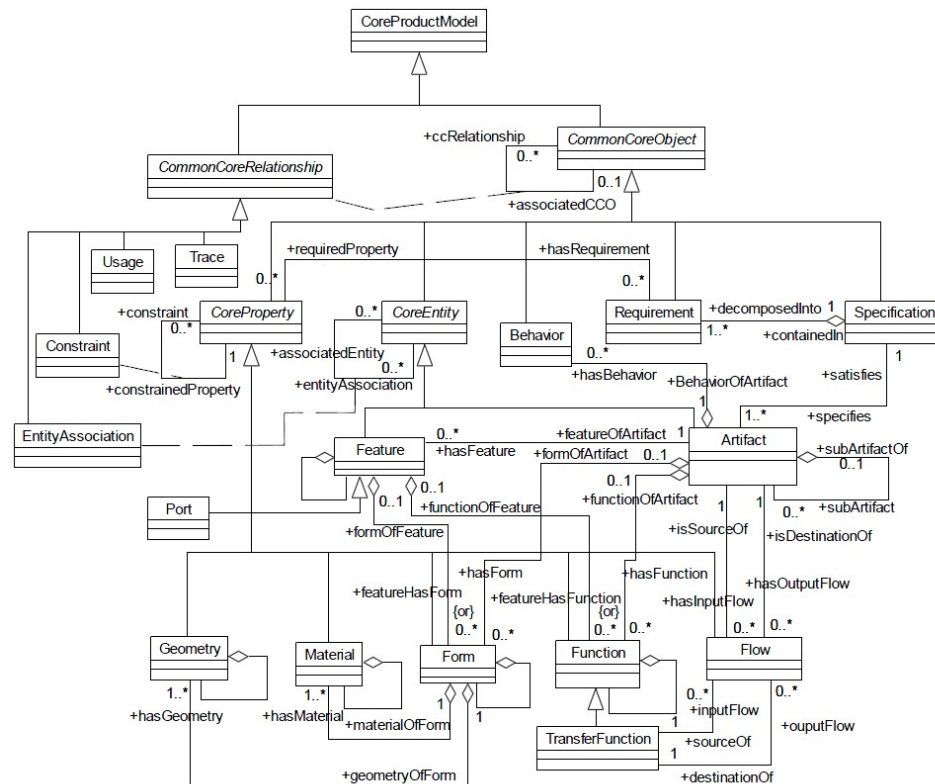


Figure 2.10 UML class diagram of the Core Product Model (Fenves et al., 2006)

As to the multi-disciplinary design of mechatronic system, the CPM model does not provide the interfaces between different disciplines. In order to meet the requirements of multi-disciplinary design, some extensions have been proposed.

(Zha, Fenves, & Sriram, 2005) propose the extension of CPM Embedded System Model (ESM) which is a feature-based approach to the co-design of hardware (HW) and software (SW) in embedded system. The extended model provides a framework for co-design of feature-based HW/SW components allowing the designer to develop a virtual prototype of embedded system through assembly of virtual components. The interfaces between HW/SW,

HW/HW and SW/SW are proposed in this model (Figure 2.11). Like the term “interface connector” of STEP AP233, “Port” is defined to describe the connection point of interfaces in ESM. To a certain extent the embedded systems can be fairly assimilated to mechatronic systems and ESM partially performed the collaboration between electronic and software disciplines, but the collaboration with mechanical discipline has not been deeply discussed. As for the micro level interface which helps the designers to fulfil the collaboration between designers, an IT platform based on ESM is developed.

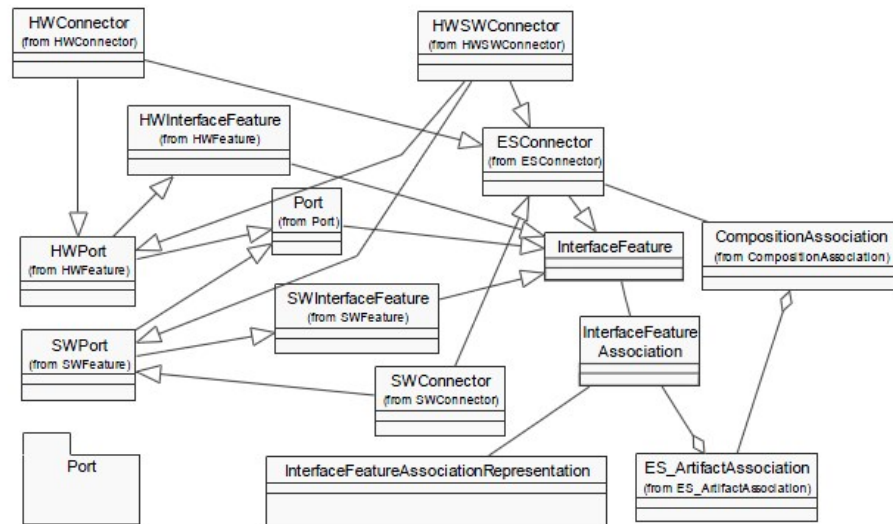


Figure 2.11 Interface feature of Embedded System Model (Zha et al., 2005)

2.3.3 PPO (Product-Process-Organisation) Model

Design process of mechatronic system requires collaboration among different disciplines and designers. The collaboration during design process can be considered as a problem that has to be solved. Therefore, the process and the organisational models have to be linked with the product model.

In order to fulfil these aims, the IPPOP (Integration of Product, Process and Organisation for improvement of engineering Performance) project has developed the PPO model which describes information of product, process and organisation (Robin & Girard, 2006; Robin, Rose, & Girard, 2007). It enhances interoperability of heterogeneous expert tools during the product development process (Noël & Roucoules, 2008). The product model developed in the IPPOP project is shown on Figure 2.12. It consists of 4 main concepts: Component, Interface, Function and Behaviour.

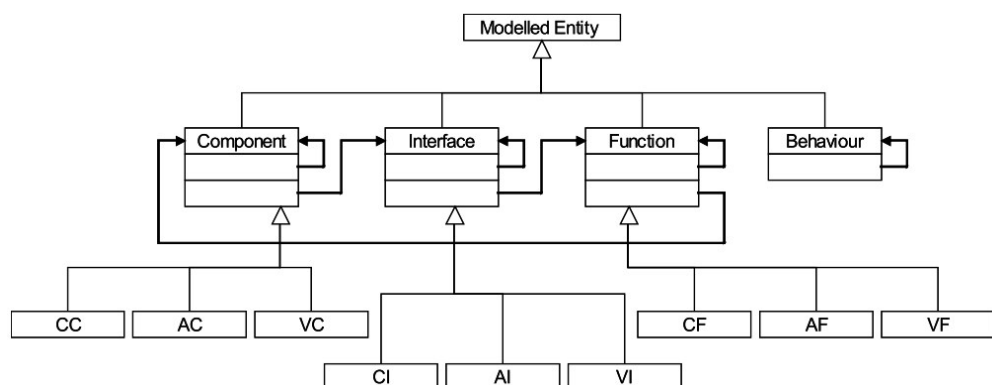


Figure 2.12 Product model class diagram (Noël, Roucoules, & Teissandier, 2005)

An interface class is described in the product model by the way a component (mechanical, electrical, etc.) may be linked to another. Although the interface class is derived into Common Interfaces (CI), Alternative Interfaces (AI) and View Interfaces (VI), it needs to be further specified for mechatronic system and the collaboration among different disciplines has to be fully implemented in the product model.

As for the micro level interface, on the one hand, the product model is extended according to the process and organisation models, based on which a decision framework has been developed (Figure 2.13). This framework can help the designers to manage the design process and to solve the design conflicts. On the other hand, a prototype of software supporting the PPO model has been developed during the IPPOP project. The designers can find all information necessary to achieve their tasks by using a specific Graphical User Interface (GUI).

The PPO model is considered as an effective support for the development process of a complex system because the data of product, process and organisation during the design process have been taken into account by the

PPO model, but it should be further specialised for mechatronic engineering. As shown with recent PPO model developments, PPO is generally considered as an extensible data model (Le Duigou, Bernard, & Perry, 2011). Hence, a special extension for design of mechatronic system can be developed based on PPO model.

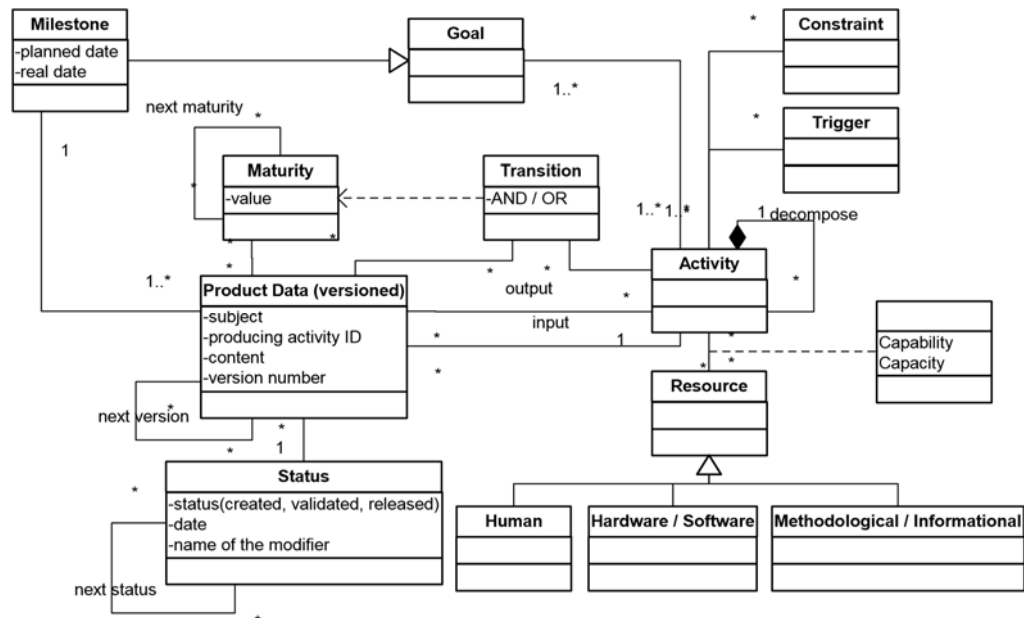


Figure 2.13 Process model developed during the IPPOP project (Nowak et al., 2004)

2.3.4 Assessment of existing product models for mechatronic engineering

Previous sections surveyed three product models which have been widely used nowadays to support the PDM functions of PLM throughout the whole product lifecycle. According to the research questions presented in Chapter 1, two criteria are proposed to evaluate the product models:

- Can the existing product models realise the macro level interface?
- Can the existing product models realise the micro level interface? As

shown in Table 2.3, the assessment of the studied product models according to the proposed criteria. All the product models reviewed before have partially developed the interfaces (macro level interface and micro level interface) to meet the requirements of collaboration between various experts and disciplines.

Table 2.3 Details of the evaluation for product models

Product model	Macro level interface	Micro level interface
STEP	<p>STEP standard only fulfils macro level interface in some specific disciplines (AP 214 specifies the automotive mechanical design process; AP 210 focuses on PCA design...). Interface is composed of interface connector and interface connection (AP 233), but the details are not given. It does not provide an effective interface to fully support the data exchange in software discipline.</p>	<p>STEP AP239 represents the planning and scheduling of complex design tasks, but it is still very generic regarding the specificities in design of mechatronic system.</p>

CPM	The Extension of CPM Embedded System Model defines the interface between hardware and software in embedded systems.	An IT platform based on CPM Embedded System Model has been established. It allows the designers to develop a virtual embedded system prototype through the collaboration between designers.
PPO	An interface class is proposed by which a subsystem (mechanical, electrical and etc.) may be linked to another, but this model is very generic and should be further specialised for mechatronic system modelling.	One the one hand, a decision frame work has been developed based on the process and organisation models; on the other hand, a prototype of software supporting the PPO model has been developed during the IPPOP project. An engineer can find all information necessary to achieve his task by using a specific Graphical User Interface (GUI).

According to the details of evaluation for product models shown in Table 2.3, Table 2.4 presents the assessment result of the product models according to the above proposed criteria. The product models, such as STEP, CPM and PPO, have partially developed the interfaces (macro level interface and micro level interface) to meet the requirements of collaboration between various experts and disciplines, but none of them can realise both the macro level and micro level interface simultaneously.

Table 2.4 Assessment of the product models regarding needs of multi-disciplinary integration

Product model	Macro level interface	Micro level interface
STEP	Partial	Partial
CPM	Partial	Partial
PPO	Partial	Yes

The above review on product models indicates that the increasing number of product models entails effective collaboration among different disciplines. Like product models, ontology has been recently applied to product modelling from the perspective of the semantic data representation (Abdul-ghafour et al., 2012). In the following sub-

section, some ontology-based approaches dedicated to the product modelling will be surveyed.

□ 2.4 Ontology-based product modelling approaches for mechatronic engineering

Although this thesis does not choose the ontology as the main research direction to solve the design data-related problems, the ontology-based product modelling approaches have been widely used to structure the product data in order to achieve the collaboration. For this reason the ontology based product modelling approaches are reviewed in this sub-section. Some of them inspire us to propose the new way for product modelling dedicated to mechatronic engineering.

2.4.1 Basic concept of ontology

The word “ontology” comes from the Greek *ontos* (being) and *logos* (words). It has been introduced in philosophy in the 19th century, by German philosophers, to distinguish the study of being as such from the study of various kinds of beings in the natural sciences. As a philosophical discipline, ontology building is concerned with providing category systems that account for a certain vision of world (Guarino, 1998). In computer engineering and information science, an ontology is a formal naming and definition of the types, properties, and interrelationships of the entities that really or fundamentally exist for a particular domain, which can be used to facilitate knowledge sharing and reuse.

Nowadays, ontology is becoming widespread in the research fields such as cooperative information systems, artificial intelligence, agent based software engineering, etc., because it proposes a means of overcoming difficulties caused by disparate terminologies, approaches and tools in knowledge representation. Thus the conception of ontology has been largely expanded and various definitions have been proposed. One of the most cited ontology definition is provided by Gruber (Gruber, 1993): “An ontology is a formal, explicit specification of a shared conceptualisation”.

2.4.2 Current research works on ontology-based product modelling approaches

Ontology-based product modelling approaches have been widely used for complex product development in a collaborative environment because they can provide a successful semantic interoperability to help the designers achieve the basis of

seamless communication and thereby enable better integration of different disciplines (Patil, Member, Dutta, Sriram, & Member, 2005).

(Yoo & Kim, 2002) use ontology as the meaning and relationship of vocabulary to improve the search capability in product databases where the product data can be provided by different standards such as STEP, XML (Extensible Markup Language) and etc. (Vegetti, Leone, & Henning, 2011) propose an ontology-based approach which can easily manages crucial features that should be taken into account in a product representation, such as the efficient handling of product families and variants concepts, composition and decomposition structures and the possibility of specifying constraints. (Patil et al., 2005) develop a Product Semantic Representation Language (PSRL) which enables the automation of the exchange of meaning associated with the data among information resources throughout the product development in order to achieve the semantic interoperability of product information. (Matsokis & Kiritsis, 2010) develop an ontology model of a Product Data and Knowledge Management Semantic Object Model for PLM, with the aim of implementing ontology advantages and features into the model. (Oestersötebier, Just, Trächtler, Bauer, & Dziwok, 2012) develop an initial multi-domain model based on the active structure and of idealized simulation models which are part of a free library and associated with the chosen solution patterns via the ontology. (Abdul-ghafour et al., 2012) propose an ontological approach based on the construction of common design features ontology, used to exchange not only the product shape, but also other design engineering data such as parameters, constraints, feature and etc. All the studies reviewed above can help the designers with various backgrounds, different terminologies to use the same concept or the same terminology during the collaborative design process. Therefore the micro level interface among the designers has been partially established, but the macro level interface dedicated to the different design disciplines for the design of mechatronic systems has not been mentioned. In order to overcome the barriers resulting from the multi-discipline during the design process of mechatronic systems, some ontology-based approaches have been developed in recent years. (I. Horvath & van Der Vegte, 2003) define a general ontology for design concepts and propose a nucleus-based conceptualization to describe the interactions between two objects or devices as a set of connected surface regions, but the nucleus concept applies primarily to the mechanical domain. (Kitamura & Mizoguchi, 2003) propose an extended device ontology to describe the component interactions in terms of conduits and input/output ports. In their proposition, they define ports in terms of form, function, and behaviour attributes. (Liang & Paredis, 2004) further develop the proposition of Kitamura and

Mizoguchi, their proposed port ontology includes axioms that can be used to support a variety of engineering design tasks, such as port refinement, port compatibility checking, and the instantiation of interaction models. The authors define a formal representation for the ports and the interfaces. However, in this proposition, it lacks a connectivity model that allows full communication among the different design teams. (Rahmani & Thomson, 2012) propose an ontology based interface design and control methodology, in which interfaces are considered as interconnections between subsystem ports and ports are specified by using an ontology that ensures consistency of interface definitions among different design teams. However, this research work does not link with CAD and PDM systems, which hinders the interoperability and collaboration among the designers because the interface information cannot be imported to and exported from a CAD/PDM system. As a result, the micro level interface among the designers is not fully realised.

2.4.3 Assessment of existing ontology based product modelling approaches

As discussed in previous sections, ontology is initially proposed for representing real world things. With the support of computer engineering and information science, nowadays it has been adopted for the product modelling in which several disciplines are involved. Current research works on ontology based product modelling approaches have been reviewed. Table 2.5 shows the evaluation results according to the proposed criteria of macro level and micro level interface.

Table 2.5 Assessment of the ontology based product modelling approaches regarding needs of multi-disciplinary collaboration

Ontology based product modelling approach	Macro level interface	Micro level interface
(Yoo & Kim, 2002)	No	Partial
(Vegetti et al., 2011)	No	Partial
(Patil et al., 2005)	No	Partial
(Matsokis & Kiritsis, 2010)	No	Partial
(Oestersötebier et al., 2012)	No	Partial

(Abdul-ghafour et al., 2012)	No	Partial
(I. Horvath & van Der Vegte, 2003)	Partial	Partial
(Kitamura & Mizoguchi, 2003)	Partial	Partial
(Liang & Paredis, 2004)	Partial	Partial
(Rahmani & Thomson, 2012)	Yes	Partial

By now numerous design methods and product models (including ontology based product modelling approaches) have been reviewed and discussed in the above section. The following section will summarise the assessment of different design approaches.

□ 2.5 Summary and assessment of studied research works

Multi-discipline based integrated design for mechatronic systems plays an increasingly key role for mechatronic systems. According to the principle of multi-discipline based integrated design, engineers intend to develop a mechatronic system with a high level integration (integrated mechatronic system) through a well-organised design method (integrated design method). As a result, two main categories of scientific problems have been pointed out: the “Process-based problems” and the “Design data-related problems”. Several approaches to overcome these problems have been put forward. The design method is considered as a potential solution to this research question because it can help the engineers from different disciplines to enable their collaboration for the increasingly complex tasks (Hazelrigg, 1996). The second main research question is relevant to the design data during the design process of mechatronic systems. Product model is dedicated to such research question and enables mechatronic design and disciplines integration because it includes all the information that can be accessed, stored, served and reused by stakeholders throughout the entire product lifecycle (Rachuri et al., 2005; X. Tang & Yun, 2008). Based on the assessment outcomes shown in Table 2.2, the existing design methods partially support the design of mechatronic system, but none of them can help the designers to achieve the macro level and micro level collaboration simultaneously. As a result, the integrated design cannot be fully achieved by existing design methods.

In order to overcome the barriers resulting from existing design methods, the thesis tries to seek solutions by making use of the product model. As discussed in Chapter 1, the macro level and micro level interfaces represented by product data can be used as an effective support to the macro level and micro level collaboration respectively. By considering the criteria on macro level and micro level interface, a survey on existing product models have been carried on.

However, according to the assessment result shown in Table 2.4, none of the product model surveyed encompasses simultaneously both the macro level and micro level interface.

The previous discussion indicates that although there have been efforts that address such two kinds of problems, the research questions which are proposed in Chapter 1 cannot be fully answered by the two approaches. Therefore, both design method and product model should be further improved to meet the requirement of multi-discipline based integrated design.

On the other hand, the evaluation results of current studies on design methods and product models greatly help us to propose the research approaches. Two main hypotheses can be drawn by evaluating and analysing existing design methods and product models:

- Interface data model which can realise both the macro level and the micro level interface should be developed as an extension of the existing product models.
- A new design method based on the proposed interface data model should be developed to achieve both the macro level and the micro level collaboration.

Focusing on the two main hypotheses, the research approaches can be proposed, which will be presented in next sub-section.

□ 2.5 Research approaches

The main research objective of this thesis is to help the designers to achieve the design and integration of multi-disciplinary interface in mechatronic engineering. The previous sections review the current studies on design methods. The evaluation result shows that none of them has provided an effective solution to achieve both the macro level and micro level collaboration during the design process of mechatronic systems to achieve the multidiscipline based integrated design. In order to overcome this problem resulting from existing design methods, the thesis tries to seek solutions by making use of the product model. As discussed in Chapter 1, the macro level and

micro level interfaces represented by product data can be used as an effective support to the macro level and micro level collaboration respectively. Therefore, like the design methods, the thesis reviews the existing product models and ontology based product modelling approaches. The evaluation result shows that the existing product model should be further developed since none of them has fully defined the macro level and micro level interface simultaneously. By considering the evaluation results from the previous survey, the research approaches can be developed.

The first research proposition is to develop a multi-disciplinary interface model which can be used as an extension of the existing product models. By considering the limitation of the existing product models, the details of an interface will be described in this data model. Moreover, how to use the proposed multi-disciplinary interface to guarantee the different sub-systems designed by different project teams integrate correctly (macro level interface) and to manage the collaboration of designers (micro level interface) should be also taken into consideration.

The second research proposition is a new design methodology with which the multi-disciplinary interface model can be implemented. The evaluation results of existing design methods show that VDI 2206 based on the V-model has fully supported the macro level collaboration but the micro level collaboration has not been achieved. With the support of multi-disciplinary interface model in which the micro level interface has been well established, a new variant of Vmodel based on the multi-disciplinary interface model can help the designers to both achieve the macro level and micro level collaboration. The research propositions will be presented in detail in Chapter 3.

Propositions

This chapter will present the propositions in detail in line with the research approaches discussed in Chapter 2. The first proposition of the thesis is the multi-disciplinary interface model. This multi-disciplinary interface model is developed as an extension of existing product models and can help the designers to overcome the limitations of existing product models and to realise the macro level and micro level interfaces. The second proposition is a new methodology for the multi-disciplinary integrated design of mechatronic systems. In this design methodology, an extended V-model is developed based on the multi-disciplinary interface model. It will help the designers to achieve both the macro level and micro level collaboration.

□ 3.1 Multi-disciplinary interface model

The first proposition of the thesis is the multi-disciplinary interface model. This multi-disciplinary interface model can be used as an extension of existing product models to help the designers to overcome the design data-related problems and to achieve the multi-discipline based integrated design of mechatronic systems.

Before detailing the multi-disciplinary interface model, let's go over several concepts which have been introduced in previous chapter. The first concept is the macro level interface. With the purpose of two sub-systems (or components) defined by the design teams of different disciplines to be interconnected, there must be compatible interfaces in mechanical, electronic/electrical and software disciplines. The second concept is the micro level interface. The micro level interface allows the designers to exchange and share information or data during the process of mechatronic design. It intends to help designers to collaborate or coordinate by sharing information through formal or informal interaction. The proposed multi-disciplinary interface model allows supporting the macro level and micro level interface simultaneously. On the one hand, it should propose a standard representation of the interfaces which may be defined by the teams of different disciplines. On the other hand, it should be also used to indicate the collaboration of designers of different disciplines and to provide a high-level guidance for organising the design process.

PROPOSITIONS

In this section, the interface classification will be first introduced. Classifying the interfaces can provide a standard representation for the interfaces defined by design teams from different disciplines. Based on the proposed interface classification, the multi-disciplinary interface model is developed and will be then introduced. The last aspect of the proposition is how to use the proposed multi-disciplinary interface model to check and guarantee the interface compatibility. The interface compatibility rules will be therefore introduced in the end of this section. Interface compatibility plays an important role during the whole design process of mechatronic systems. It can be used to help the designers to ensure the right integration of the different sub-systems designed by the project teams from different disciplines during the discipline-specific design sub-process. In addition, the multi-disciplinary interface model can support the improvement of mechatronic systems and help the designers to achieve a complete and appropriate architecture. LEGO Mindstorms¹² is finally used to illustrate the first proposition on multi-disciplinary interface in this section.

3.1.1 Introduction of LEGO Mindstorms used for training the basics of mechatronic engineering

The LEGO Mindstorms series of kits contain software and hardware to create customizable, programmable mechatronic systems, such as powered vehicles, working robots and etc. They include an intelligent brick computer that controls the system, a set of modular sensors and motors, and LEGO parts to create the mechanical systems (Bagnall, 2002).

The triangle of the Figure 3.1 shows the aspects of mechatronics proposed by (Schöner, 2004). The analysis of the components provided by LEGO Mindstorm indicates that the aspects of mechatronics can be well represented by the components of LEGO Mindstorm. The details will be presented as follows.

¹² <http://mindstorms.lego.com>

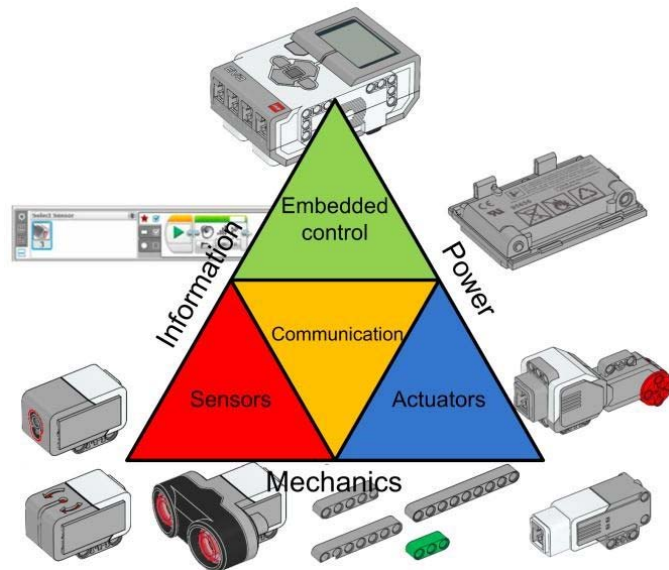


Figure 3.1 Mechatronics and LEGO

3.1.1.1 Mechanical parts

The mechanical parts provided by LEGO Mindstorm are called LEGO technic pieces. These technic pieces are the most basic building blocks of LEGO Mindstorm (Figure 3.2). They can be integrated with standard LEGO but are so diverse that they're a whole range of building blocks in their own right. They can easily be assembled with LEGO motors to create the powered vehicles or working robots (Orionrobots, 2011).

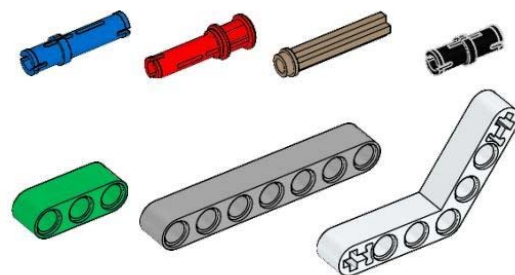


Figure 3.2 Examples of LEGO technic pieces (Dimensionsguide, 2011)

3.1.1.2 Information code

The LEGO Mindstorms NXT software enables the designers to program the NXT robotic invention and upload the programs to the NXT via USB or Bluetooth connectivity. LEGO Mindstorms is command block programming, rather than code programming. All the programming blocks that are used for controlling the

PROPOSITIONS

mechatronic systems have already been pre-programmed. Figure 3.3 shows the (a) action blocks, (b) flow blocks and (c) sensor blocks respectively.

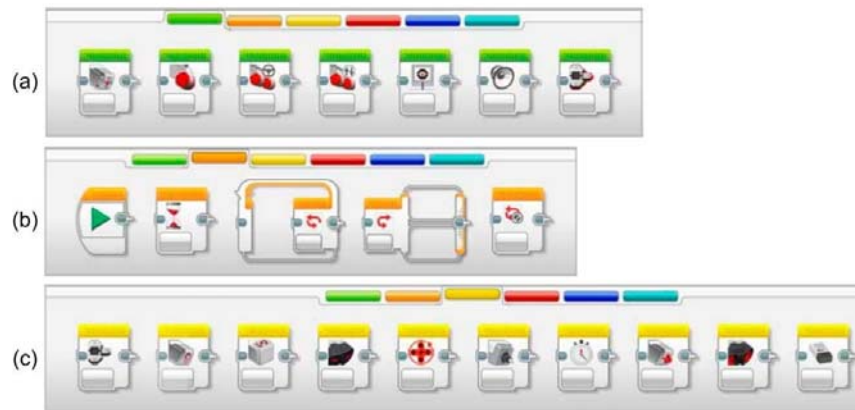


Figure 3.3 Programming blocks of LEGO Mindstorm

3.1.1.3 Power supply part

Two mode of power supply is provided by LEGO Mindstorm. The designers have the choice of using (a) the rechargeable battery pack or (b) normal AA batteries (Figure 3.4). The designers should consider carefully his choice when constructing the mechatronic system according to the different characteristics of rechargeable battery and the batteries. For instance, six AA batteries weigh more than the rechargeable battery, and the EV3 Brick with the rechargeable battery installed is slightly larger than the EV3 Brick with six AA batteries.

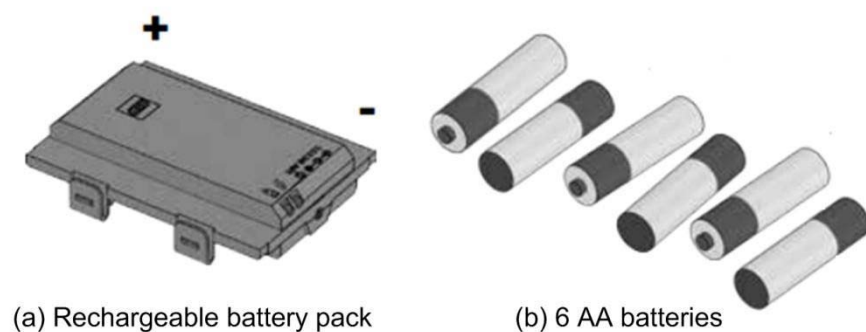


Figure 3.4 Power supply of LEGO Mindstorm

3.1.1.4 Sensors

PROPOSITIONS

According to (Schöner, 2004), sensors are developed by integrating the information code and mechanical parts. They provide details information about the status of mechanical parts, allowing corrective response to different operating conditions or unpredictable changes in the process. LEGO Mindstorm provides four types of sensors, (a) colour sensor, (b) gyro sensor, (c) touch sensor and (d) ultrasonic sensor (Figure 3.5).

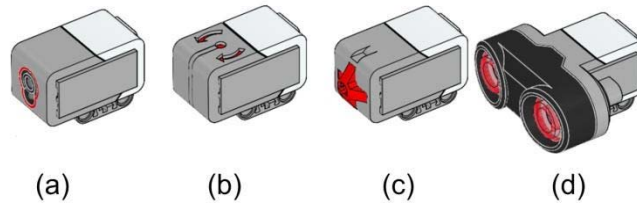


Figure 3.5 Sensors of LEGO Mindstorm

3.1.1.5 Actuators

Actuators are used to increase actuation forces or actuation speed based on the integration of power parts and mechanical parts (Schöner, 2004). Two kinds of actuators are introduced by LEGO Mindstorms, (a) large motor and (b) medium motor (Figure 3.6).

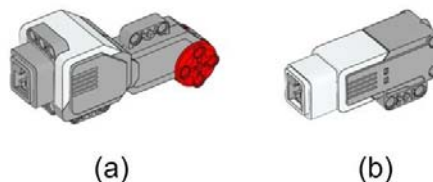


Figure 3.6 Actuators of LEGO Mindstorm

3.1.1.6 Embedded control

With the goal of an automatic or more reproducible process, an embedded control system based on electronics and software should be added in the mechatronic systems. EV3 brick of LEGO Mindstorm, as an embedded control system, is the “brain” of a Mindstorms machine (Figure 3.7). It lets the robot autonomously perform different operations.

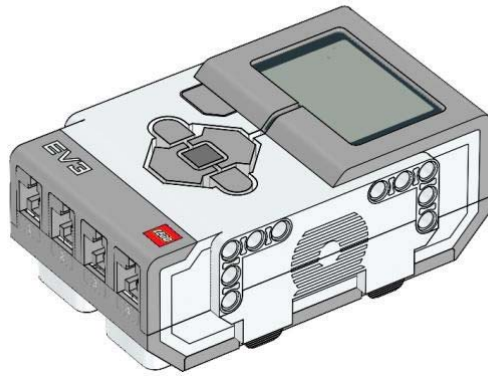


Figure 3.7 EV3 brick of LEGO Mindstorm

This sub-section shows that all the aspects of mechatronics can be well represented by the components of LEGO Mindstorm. Therefore the examples existing in LEGO Mindstorm will be used to illustrate our propositions on interface model hereafter. In next sub-section, all the characteristics of the interfaces will be presented through the way they are classified.

3.1.2 Interface classification

From the mid-1980s, the interface between systems or sub-systems has been widely used in software engineering (Dorfman, 1990; Hoffman, 1990). During design process of software, separated module of a program executes one aspect of the desire functionality. Such modules interact with each other through interfaces. As system became increasingly complex, it is divided into subsystems. The topic of interface is at the heart of the multi-disciplinary nature of Systems Engineering (Fosse & Delp, 2013). Interface management is considered as one of the most powerful tools of systems management (Blyler, 2004). The interface in mechatronic systems which refers to the logical or physical relationship integrating the elements of one mechatronic system or the elements with their environment can be used to describe the interactions of sub-systems designed by different disciplines. Therefore it is significant to propose a proper interface classification in order to represent much more details of an interface and help designers to avoid the confusion by the misuse of interfaces. Next sub-section will present the related work of interface classification.

3.1.2.1 Related work

To solve the collaboration problems during the design process of mechatronic systems, (Steward, 1965) describes the interactions of sub-system as “information flows”, but such information flows are not described in detail. (Counsell, Ian, David, & Duffy, 1999) describe the connections between different components as material, information and power. (Sellgren, 2006) proposes that interfaces can be classified as attachment, constraint and contact. His proposition mainly focuses on the physical interface. An international standard ISO/IEC 81346 also specifies how to define a physical interface (ISO/IEC 81346-1, 2012). However, modelling and controlling relationship for the formalized specification of interfaces have not been fully realized (TorrySmith, Mortensen, & Achiche, 2013). (Chen, Bankston, Panchal, & Schaefer, 2009) classify the interfaces as the “constraints” between electrical/electronic discipline and mechanical discipline, but the interfaces between software discipline and electrical/electronic discipline or mechanical discipline have not been mentioned. (Pahl et al., 2007) propose a method named Modular Product Development (MPD) for complex system. This method starts by decomposing the product into modules. The exchanges of energy, materials, and signal between the modules were mentioned in this method. (Liang & Paredis, 2004) develop a more detailed classification based on the proposition of (Pahl et al., 2007) by refining the energy as electrical, mechanical and hydraulic, etc. However, these two methods do not consider the interface between software and other disciplines. (Komoto & Tomiyama, 2012) believe that some physical implementations have nothing to do with transformation of energy, material, and signal (e.g., a function to fix connection between two mechanical components or a function that holds a position), but they can be used to connect two components as the interfaces. Thus geometry plays a crucial role during the design process. They point out that attention should be also paid to such geometric information. (Sosa, Eppinger, & Rowles, 2000) distinguish the interfaces in terms of spatial dependency, structural dependency, energy dependency, material dependency and information dependency. Such classification method may lead to the misuse of overlapping interfaces. For instance, the material dependency is described as “a requirement related to transferring airflow, oil, fuel, or water” (Sosa et al., 2000). However, such process of material transfer often occurs with the energy transfer which was defined as “energy dependency”. (Bettig & Gershenson, 2010) point out the interface representation problem and try to identify an overall representational schema. Seven classes of interfaces are firstly suggested: Attachment interface, transfer interface, control and communication interface, power (electrical) interface, spatial interface, field interface and environmental interface. The seven interface classes are then

PROPOSITIONS

reduced to four general classes of interface: attachment interfaces, control and power interfaces, transfer interfaces and field interfaces. The reduced classification defines the field interface as “an interface that transmits energy, material or signal as an unintended side-effect of the intended function of a module”. This classification begins to consider the negative effects of interfaces. However, the field interface is so generic and need to be detailed.

By analysing the interface classifications for mechatronic systems, limitations of each previous classification are shown the in Table 3.1:

Table 3.1 Main limitations of previous classifications

Previous classification	Limitations
(Steward, 1965) (only information) (Counsell et al., 1999) (material, information and power) (Liang & Paredis, 2004) (energy, material, and signal) (ISO/IEC 81346-1, 2012; Komoto & Tomiyama, 2012; Sellgren, 2006) (only geometry or physical interface) (Chen et al., 2009) (only constraint: electric/electronic and mechanic)	Does not cover all types of information transferred in the interface
(Counsell et al., 1999) (material, information and power) (Liang & Paredis, 2004; Pahl et al., 2007) (energy, material, and signal) (Sosa et al., 2000) (space, structure, energy, material and information) (Bettig & Gershenson, 2010) (attachment, transfer, control and communication, electric, space, field and environment)	Leads to the misuse of overlapping interfaces (such as the process of material transfer often occurs with the energy transfer)
All reviewed classifications except (Bettig & Gershenson, 2010) (Bettig and Gershenson focus on the negative effects by describing such negative interface as field interface.)	Neglect the negative effects of interfaces

3.1.2.2 Interface classification for design of mechatronic systems

As previously discussed, the existing interface classifications show several drawbacks. Moreover, some properties of interface have not been revealed by existing classifications. The new interface classification proposed in the thesis concerns an interface through three attributes based on literature review: Type, Configuration and Desired/undesired. The details will be presented and the LEGO Mindstorm will be used to illustrate the proposition on interface classification.

PROPOSITIONS

3.1.2.2.1 Attribute "types" of interface classification

The first attribute focuses on which types of transfers occur through one interface. Four general types of interfaces will be suggested as follows:

- (1) Geometric interface indicates how one element is physically connected to another.

Figure 3.8 shows an example of geometric interface existing in LEGO Mindstorm. In this example, two technic pieces (one connector peg with friction and one frame) are connected through one geometric interface.

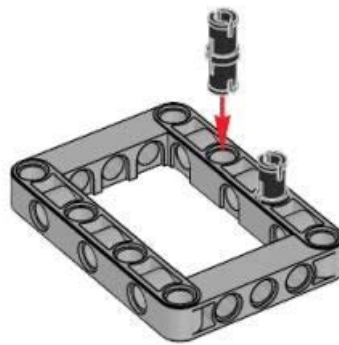


Figure 3.8 Example of geometric interface

- (2) Energy interface indicates how energy (electrical energy, mechanical energy...) is transferred between elements.

The interface between the EV3 brick and the rechargeable battery is considered as an energy interface through which the electrical energy is transferred (Figure 3.9).

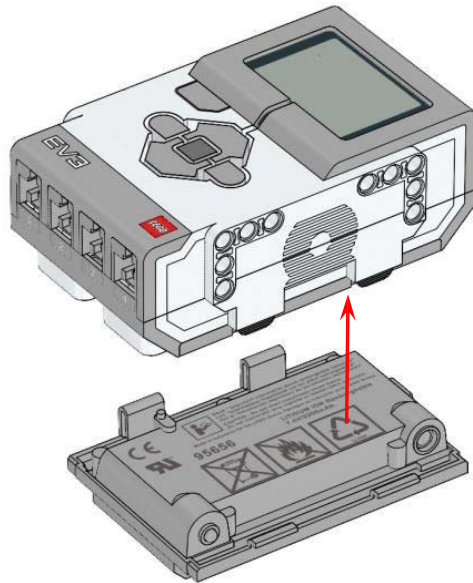


Figure 3. 9 Example of energy interface

- (3) Control interface indicates how one element will be controlled by another, or in other words, one element receives and executes the order sent by another, which is mainly related to the electronic discipline of mechatronic systems.

Figure 3.10 shows the control interface between the EV3 brick and the large motor of LEGO Mindstorm.

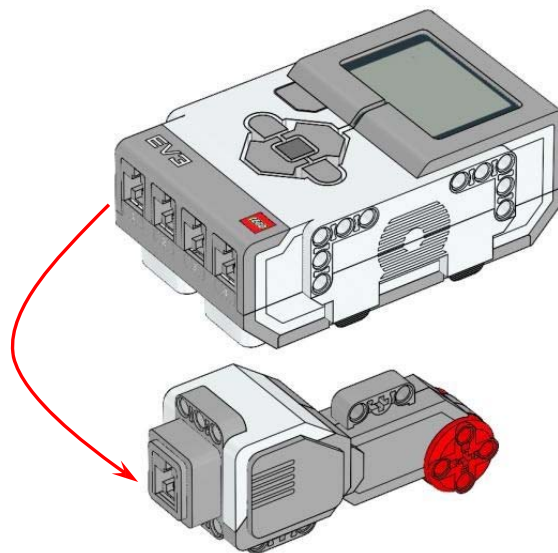


Figure 3.10 Example of control interface

- (4) Data interface indicates how communication information is transferred between two components, which is mainly related to the computer program in software discipline of mechatronic systems.

PROPOSITIONS

The interface between the PC and the EV3 brick is considered as a data interface (Figure 3.11). The program created in PC can be sent to the EV3 brick through the USB cable or by making use of Bluetooth.

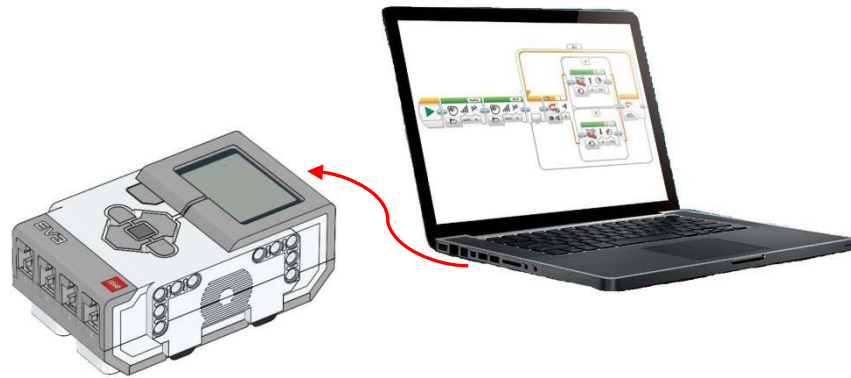


Figure 3. 11 Example of data interface

Nowadays, the increasing integration of mechatronic systems shows the trend that different types of transfers exist in one interface. Such interface can be generally divided into two cases. The first case concerns the interface through which a primary type of information is transferred while other types are transferred as subsidiary information. For instance, as for the interface between two electrical components, the electrical energy (voltage) can be transferred through this interface, which is considered as the primary type of transfer. Meanwhile, the geometric connections (pin numbers) considered as this interface's subsidiary types of transfers exist between the two components in order to have a better physical integration. The second case concerns the interface in which several types of transfers exist simultaneously and the priority of such transfers cannot be clearly defined. For instance, the technology of power-line communication is used to carry data among conductors that are also used simultaneously for AC electric power transmission or electric power distribution (Ferreira, Lampe, Newbury, & Swart, 2010). In other words, data and energy can be transferred simultaneously through one interface and they are of equal importance for this interface. Such interfaces described in the two cases should be further decomposed and refined into sub-interfaces according to the different transfers through them.

3.1.2.2.2 Attribute "configuration" of interface classification

The second attribute neglected in previous research is the configuration of interfaces. The three main elements in SE are component, environment and interface. Therefore

PROPOSITIONS

the interfaces related to the design of mechatronic systems should take into account not only (1) the interface between components (C-I-C), but also (2) the interface between a component and the environment (C-I-E), (3) the interface between a component and an interface (C-I-I), (4) the interface between two interfaces (I-I-I) and (5) the interface between an interface and the environment (I-I-E) (Table 3.2).

Table 3.2 Five types of interfaces

	Component	Environment	Interface
Component	(1) C-I-C	(2) C-I-E	(3) C-I-I
Environment	(2) C-I-E	No	(4) I-I-E
Interface	(3) C-I-I	(4) I-I-E	(5) I-I-I

- (1) Interface between two components (C-I-C) indicates how one component connects, interacts and collaborates with another.

The red arrows of Figure 3.12 show the interface between two components. In this example, the interface between the two components indicates how the two components are connected with each other.

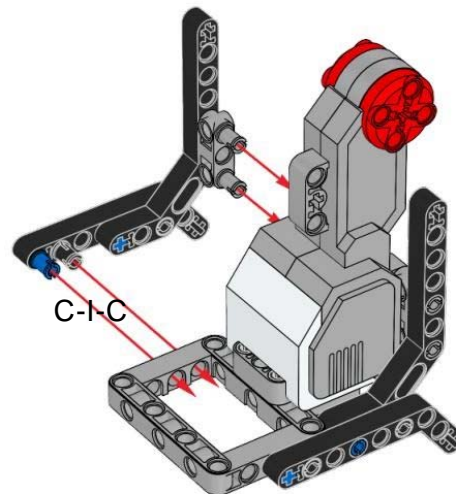


Figure 3.12 Example of C-I-C

- (2) Interface between component and environment (C-I-E) indicates how the component operates in certain environment.

Figure 3.13 shows an example of interface between component and environment. In some design cases, the designers should propose the architecture of the mechatronic system according to the different

environmental characteristics. In this example, the designers choose a continuous track rather than tyres by considering the interface between the system and the environment, because the large surface area of the tracks distributes the weight of the vehicle better than tyres on an equivalent vehicle. This enables a continuous tracked vehicle to traverse soft ground with less likelihood of becoming stuck due to sinking.

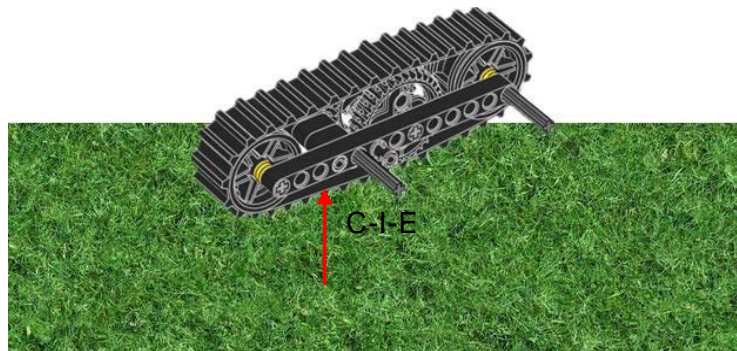


Figure 3.13 Example of C-I-E

- (3) Interface between component and interface (C-I-I) indicates that one interface must be accommodated by the effects generated by other components, such as heat, magnetic fields, vibration and other effects, or one component must be accommodated by the effects generated by an interface.

The example of C-I-I will be shown together with the interface I-I-I in Figure 3.14.

- (4) Interface between two interfaces (I-I-I) indicates that two interfaces are affected and interacted by each other.

The examples of C-I-I and I-I-I can be found in Figure 3.14. Three components of a robot are shown in this figure. The colour sensor (a) is used to detect the path (two black coloured strips separated by a white strip) along which the robot moves forward. The arm of the robot is actuated by one motor (b) while the wheels are driven by another motor (c). The detecting distance between the colour sensor and the path is considered as an important parameter of the interface between colour sensor and the path (an interface C-I-E) because the performance of the colour sensor is extremely sensitive to such distance. However, the vibration generated by the motor (b) (considered as a component) or the connection between the motor (c) and the wheel (considered as an interface C-I-C) greatly affects the detecting distance between the colour sensor and the path. Therefore the interface between the

motor (b) and the C-I-E (interface between the colour sensor and the path) is an interface C-I-I while the interface between the C-I-C (interface between the motor (c) and the wheel) and the C-I-E is an interface I-I-I.

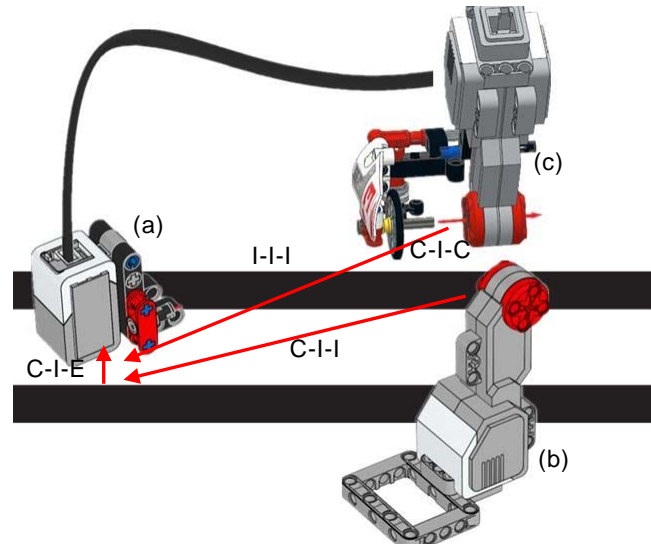


Figure 3.14 Examples of C-I-I and I-I-I

- (5) Interface between environment and interface (I-I-E) indicates how an interface is affected by the environmental effects.

Figure 3.14 shows an example existing in LEGO Mindstorm. The robot constructed with LEGO Mindstorm can be controlled by an official command app via Bluetooth or Wi-Fi; therefore an interface C-I-C between the command app and the robot. However, such wireless communication through the interface may be disturbed by the electromagnetic signals existing in the environment. In other words, the environment factors affect the interface and the interface between the environment factors and the C-I-C is an interface I-I-E.

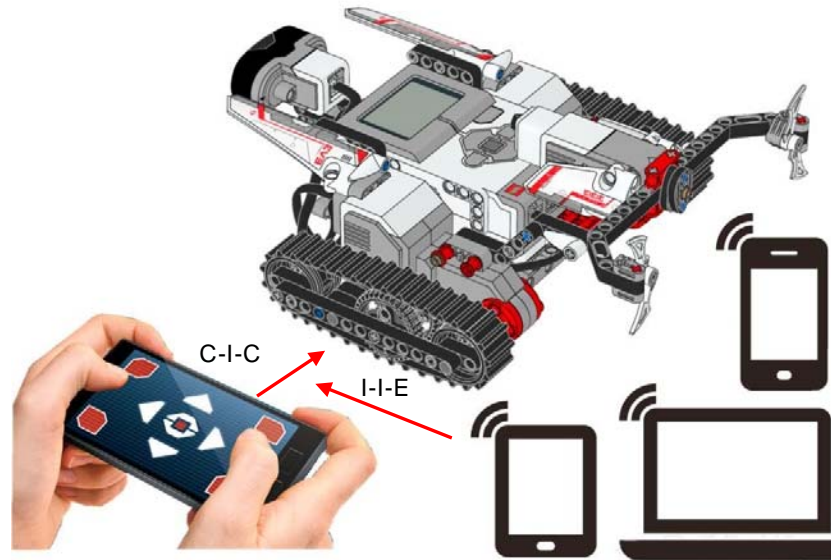


Figure 3.15 Examples of I-I-E

In current design cases, a simple example of these configurations C-I-I, I-I-I or I-I-E is the EMC (ElectroMagnetic Compatibility). A typical electronic system usually consists of several components which communicate with each other. A means to provide the energy to these components is usually the commercial AC power, which may have the potential for emitting and picking up electromagnetic energy (Paul, 2006). Such electromagnetic energy may be generated by a certain component, or during the process of AC power transfer through one interface. It will greatly affect the performance of other elements of the whole system. In order to improve the performance of the mechatronic system, such interfaces must be considered during the design process.

3.1.2.2.3 Attribute "desired/undesired" of interface classification

The last attribute proposed by the interface classification is the desired/undesired interface. The desired interface is used to describe the interface which creates positive effects (e.g., data or energy transmission), while the undesired interface is used to describe the interfaces which create the unintended sideeffects (e.g. heat, magnetic fields, vibration and other side effects).

The interface shown in Figure 3.11 is an example of desired interface and that shown in Figure 3.15 is an undesired interface.

PROPOSITIONS

However, attention should be paid to a special case in which one interface can be used to transfer useful data or energy transmission, and meanwhile, it creates the unintended side-effects. Such interfaces described should be further decomposed and refined into desired sub-interface and undesired sub-interface according to the different transfers through it.

Figure 3.16 shows an example of such special case. On the one hand, the interface between the motor and the wheel transfers mechanical energy, which is considered as a positive effect; on the other hand, such interface also creates vibration, which greatly affects the performance of the colour sensor (shown in Figure 3.14).

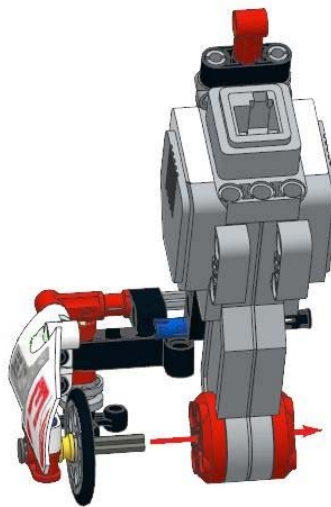


Figure 3.16 Example of an interface creating positive effects and unintended side-effects simultaneously

A representation with UML class diagram of such classification is presented in Figure 3.17. The class Interface has four attributes: name, type, configuration and desired.

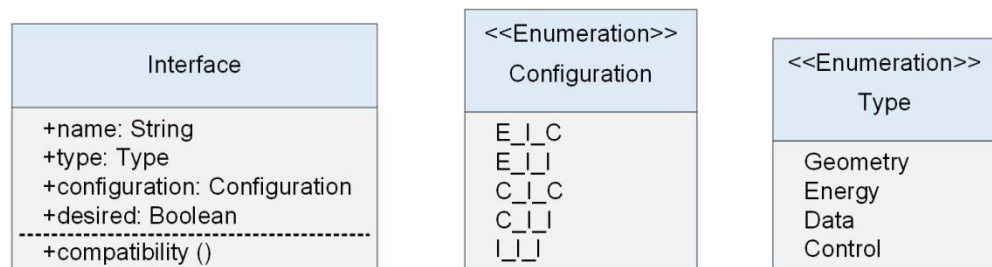


Figure 3.17 UML class diagram of interface and its classification

The attribute **name** is used to store the name of the interface to distinguish it from the others and to be able to track changes performed during the design process. The attributes **type**, **configuration** and **desired**, represent the interface classification. One

PROPOSITIONS

method **compatibility()** is contained in the interface class, which will be discussed in Section 3.1.4. Two Enumeration types, **Type** and **Configuration** are created for the attributes type and configuration.

The Object Constraint Language (OCL) (Warmer & Kleppe, 1998) is used to define the details of the attribute **configuration** as follows:

```
inv ConfigurationDetail:
    self.component_link->size()=2                                implies
Interface.configuration=Configuration::C_I_C and
    self.interface_link->size()=2                                implies
Interface.configuration=Configuration::I_I_I and
self.interface_link->size()=1    and    self.component_link->size()=1
implies Interface.configuration=Configuration::C_I_I and
self.environment_link->size()=1    and    self.component_link->size()=1
implies Interface.configuration=Configuration::E_I_C and
self.environment_link->size()=1    and    self.interface_link->size()=1
implies Interface.configuration=Configuration::E_I_I
```

In these formulas, **self.component_i_link**, **self.interface_i_link** and **self.environment_i_link** respectively represent the associations between the component and the interface, two interfaces and the environment and the interface. The **size()** operation is used to calculate the multiplicity of an association. For example, if the multiplicity of the association between the component and the interface is “2”, which means two components are connected by an interface, it indicates that the configuration of this interface is C_I_C.

The attribute **desired** describes whether an interface is a desired interface (**desired=true**), i.e. it has a positive effect on the system, or an undesired interface (**desired=false**), i.e. it has a negative effect on the system.

This section has introduced the interface classification, which is considered as the foundation of the multi-disciplinary interface model. In the following section, by considering the drawbacks of the interface model represented by existing product models in Chapter 2, a new interface model which can be used as the extension of existing product models will be introduced based on the interface classification proposed in this sub-section.

3.1.3 Interface model

PROPOSITIONS

The above evaluation of the existing product models in Chapter 2 shows the concept of interface model has not been fully developed in existing product models. On the one hand, the proposed interface classification should be included in the interface model. On the other hand, the relationship between the interface and other entities of the product model should be represented.

3.1.3.1 Port representation

In the context of multi-disciplinary interface modelling of mechatronic systems, the “port” refers to the primary location through which one part of the system interacts with other parts of the environment. In previous section, five different configurations of interfaces have been discussed (shown in Table 3.2). According to the configurations of interfaces, three types of ports exist in a mechatronic system: component port, interface port and environment port. Figure 3.18 shows the three types of ports existing in the different interface configurations and two elements are connected by an interface through these ports.

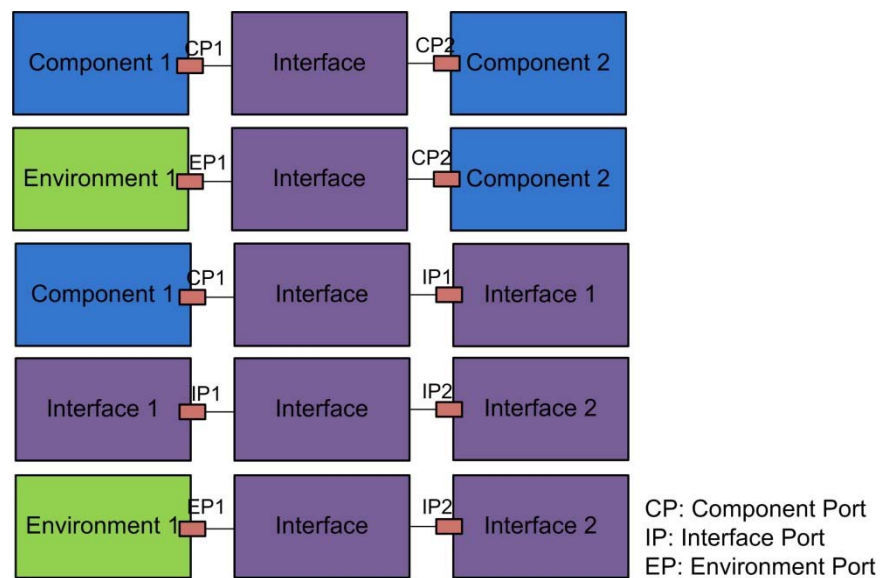


Figure 3.18 Three types of ports existing in the different interface configurations

- (5) Component port: the component port is the connection point of a component which interacts with other elements of a system. Every component can interact with several elements in one system, so one component can have more than one port.
- (6) Environment port: the environment port is the point where the external factor of the environment affects other elements of the system. The environment can affect several elements in one system, so it can possess more than one port.

- (7) Interface port: the interface port is the location which affects other elements of the system. Like the component and the environment, the interface can also possess more than one port. However, the main function of one interface is to connect two separated elements and to transfer the exchange between them. Therefore an interface can possess more than one port, but it can only connect two ports.

Figure 3.19 shows the UML class diagram of the port, and three attributes have been taken into consideration to describe one port: **name**, **direction** and **visibility**. This UML class diagram will be discussed in more detail as follows:

- Like the attribute **name** in the class interface, the attribute name in the class Port is used to store the name of one port to distinguish it from the others and to track changes.
- The attribute **direction** represents the direction of the transfers through this port. The direction of a port has been used to define which one is the master and which one is the slave of the two elements linked by the interface (Bruun, Mortensen, Harlou, Wörösch, & Proschowsky, 2015). A compartment listing the attributes (in, out and in/out) for the enumeration is placed to indicate that the transfer flows in (out of or in & out of) the element through the port. This attribute comes from the electrical engineering but is extended to the mechatronics engineering in this thesis.
- The last attribute of the class port is the **visibility**. This attribute describes how the port can be accessed. The **visibility** has been considered as a very important attribute which specifies whether it can be used by other stakeholders (Booch et al., 1998). The authorized values are “public”, “protected” and “private”. The parameter and document linked with one public port is accessible directly by any engineers from any disciplines during the design process. A protected port can only be accessed by the creators and the authorised engineers from other disciplines, but parameters and documents related to it become un-changeable. The port carrying the private property is accessible only by those who design it. In summary, the **visibility** allows the interface creator or the system architect to manage access rights. Through that functionality, the impacts of the changes can be more precisely anticipated and managed. This attribute comes from the computer engineering but is extended to the mechatronics engineering in this thesis. The attributes **direction** and **visibility** can greatly help the designers to collaborate or

PROPOSITIONS

coordinate during the design process of mechatronic systems, so they are both considered as an effective support for the realisation of micro level interface.

- The ports can be decomposed during the design process of mechatronic systems, so the class **Port** can be an aggregation of itself.
- The class **Port** is also the aggregation of class **Parameter**. The class **Parameter** specifies the parameter related to the port which can be quantified, such as the wheel size, the input impedance or the image resolution. The attribute **value** in the class **Parameter** is used to express an exact value of a parameter. In some cases, not all the parameters can be specified precisely; such parameters can be defined by an interval ([minValue, maxValue]). However, the class **Document** is used to store the documents which cannot be quantified to describe the port. For example, during the conceptual design phase, the exact ports' parameters of components cannot be accurately decided, and sometimes the designers just use a rough description to describe the port. Such description can be stored in the class Document. Another example of the document is a CAD file (representing for instance a bounding box or a frontier between two components) or the data sheet. During the detailed design phase, the CAD file or the data sheet can be stored in the class Document to provide more details about the port to the designers of other project teams.

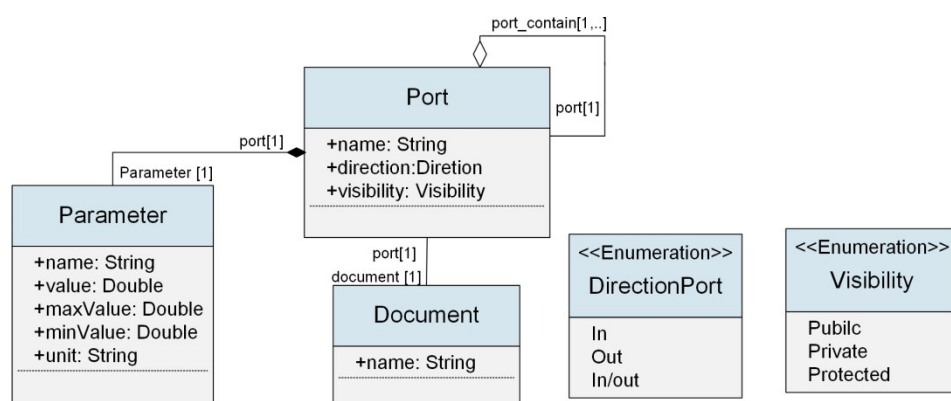


Figure 3.19 UML class diagram of Port

Considering the modelling of port presented previously, the multi-disciplinary interface model will be discussed in next subsection.

3.1.3.2 Multi-disciplinary interface model

PROPOSITIONS

In this thesis, the interface is considered as a special element of a mechatronic system. On the one hand, like the component and the environment, the interface can affect other elements of the system, so an interface can have more than one port. On the other hand, the interface is also considered as the logical or physical relationship through which two elements interact with each other, so it can only connect two ports. The proposed interface model can provide the designers with a standard representation for the interfaces defined by project teams from different disciplines.

Figure 3.20 shows the interface model represented as a UML class diagram.

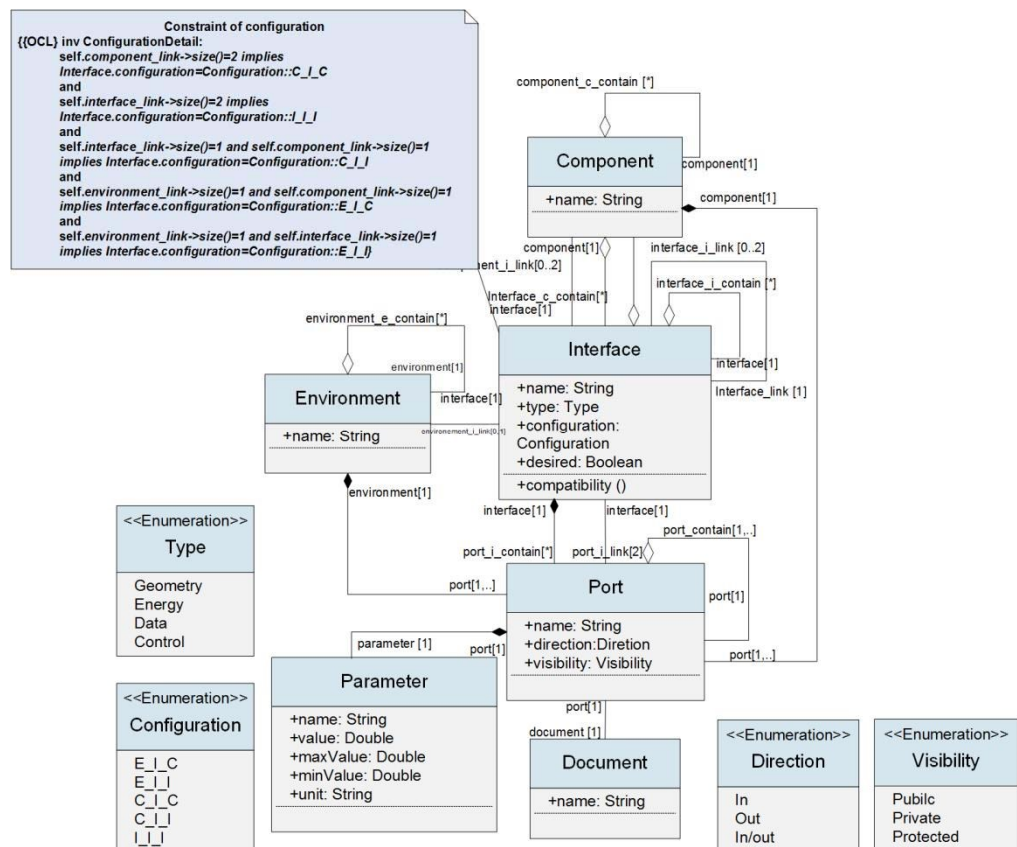


Figure 3.20 UML class diagram of interface

This interface model will be discussed in more detail as follows:

- A component can be decomposed into several sub-components connected by sub-interfaces, while an interface can be also decomposed into several sub-components and sub-interface. As a result, the class **Component** (or **Interface**) can be an aggregation of **Interface** (or **Component**) and itself. The component and interface decomposition will be presented in detail in Section 3.2.
- According to the five different interface configurations, a constraint type **Constraint of configuration** uses the Object Constraint Language (OCL) to

define the details of the attribute **configuration** in the class **Interface**. The multiplicity of the association among the class **Component**, the class **Environment** and the class **Interface** presents this fact.

- The environment, the component and the interface can have more than one port. However, the main role played by the interface is to transfer different kinds of information, energy, etc. through two ports, so one interface can only link to two ports. Therefore there exist two different links between the class **Interface** and the class **Port**. In Figure 3.19 the association ends, **port_i_contain** and **port_i_link** represent, respectively, the ports possessed and connected by one interface.

The multi-disciplinary interface model is proposed in the thesis as an extension of existing product data models. Therefore the links between the proposed interface model and the product models reviewed in Chapter 2 can be established.

- Link with STEP AP 233: In AP 233, a multi-disciplinary “interface connector” is defined as the term for one part of a system that interacts with other parts or the environment, and the interface connection as the link between connectors. By analysing the definition of “interface connection” and “interface connector”, their corresponding classes – class **Interface** and class **Port** in the data model can be found.
- Link with CPM ESM: ESM is developed as an extension of the product model CPM for the embedded system. The extended model provides the interface features between hardware/software, hardware/hardware and software/software. In ESM, “Interface feature” is defined to express the overall form and structure of an embedded system or its hardware/software components and their relationships, and “Port” is defined to describe the connection point of interfaces. Therefore the interface model can be well integrated to the CPM and extend it to allow the designer to develop a mechatronic system.
- Link with PPO: an interface class is described in the product model of PPO by the way a component (mechanical, electrical, etc.) may be linked to another, but its details have not been given. The interface model gives much more details about the interfaces existing in mechatronic systems and is considered as a useful complement of PPO. The previous discussion shows that the main entities of existing product models can be found in the proposed interface model so that mapping can be specified between existing product models and the interface model. Table 3.3 shows the main entities of the

PROPOSITIONS

proposed multi-disciplinary interface model and their equivalent entities of other product models. Therefore the interface model can be used as an extension of existing product models.

Table 3.3 Mapping between main entities of multi-disciplinary interface model and their equivalent entities in other product models

Multi-disciplinary interface model	STEP AP 233	CPM-ESM	PPO
Component	Sub-system	CoreEntity	Component
Interface	Interface connection	InterfaceFeature	Interface
Port	Interface connector	Port	(No equivalent)

3.1.4 Interface compatibility rules for design of mechatronic systems

The thesis proposes the interface compatibility rules for design and integration of mechatronic systems based on the multi-disciplinary interface model. The design of mechatronic systems requires the multi-disciplinary collaboration of different project teams. Therefore most of nowadays design methodologies for mechatronic engineering, such as V-mode (VDI 2206, 2003) or hierarchical design (Hehenberger, Poltschak, Zeman, & Amrhein, 2010), propose a concurrent engineering approaches where the design tasks for different sub-systems can be executed by different design teams in parallel. According to such concurrent design methodologies, all the subsystems defined by different teams should be integrated during the integration design phase. Therefore the interface compatibility rules can be integrated with existing design methodologies to guarantee the different sub-systems assemble correctly and ensure the multi-disciplinary integration among design. The proposed interface compatibility rules will be first introduced in this sub-section. The solutions to solve the incompatibility problems will be then proposed.

3.1.4.1 Interface compatibility rules

As discussed in Section 3.1.3, one method **compatibility()** is contained by the interface class. Once the data model of an interface is instantiated, the interface

PROPOSITIONS

compatibility should be checked by this method. One example is cited here to illustrate the compatibility test method. Two components (Component 1 and Component 2) are connected by an interface (Interface) through the ports (CP1 and CP2). Two compatibility rules are presented as follows:

Rule 1:

```
CP1. Parameters1. value = CP2. Parameters2. value  
CP1. Parameters1. unit = CP2. Parameters2. unit
```

Rule 2:

```
CP1. Parameter1. value< CP2. Parameters2. max  
CP1. Parameter1. value> CP2. Parameters2. min  
CP1. Parameter1. unit= CP2. Parameters2. unit
```

In the compatibility **Rule 1**, **CP1.Parameter1** represents the parameter stored in the class **Parameter** of the port **CP1**, and **CP2.Parameter2** is the parameter of port **CP2**. In order to ensure the two components integrate with each other correctly, both the value and the unit of the parameters of **CP1** and **CP2** should be equal.

One example can be found in LEGO Mindstorm. By analysing the interface between the power supply and the rechargeable battery, only the power with the frequency of 50 Hz (or 60 Hz) can recharge the battery.

The **Rule 2** is applied to two cases. The first case is that sometimes the design parameter of one port is not specified by an exact value accurately but described as a constraint, such as the minimum input current, maximum diameter of the hole and etc.

Concerning the previous example of the interface between the power supply and the rechargeable battery again, the power delivered by an AC voltage from 120 V (minValue) to 240 V (maxValue) can recharge the battery.

The second case concerns the component tolerance. Component can hardly hold dimensions precisely to the nominal value, so there must be an acceptable degree of variation.

For example, LEGO bricks have tolerances as small as 10 micrometres (Orionrobots, 2011).

PROPOSITIONS

If the port **CP2** specifies the parameter by using an interval (**minValue**, **maxValue**), in the first case, this interval is used to describe the constraint under which the CP1 should obey. While in the second case, this interval describes the minimum possible component size and the maximum possible component size. Generally speaking, the **Rule 2** means that the parameter of

CP1 should satisfy that **CP1.Parameter1.value** \in (**CP2.Parameters2.minValue**, **CP2.Parameters2.maxValue**).

A compatible interface indicates that the two elements linked by this interface integrated correctly and the design process can move on to the next step. However, if an interface proves to be incompatible, some measures should be taken to solve the incompatibility problem. The solutions to solve such incompatibility problems will be presented in next sub-section.

3.1.4.2 Solutions to incompatibility problems

Two solutions can be adopted to deal with the interface which reveals to be incompatible. These solutions will be presented as follows:

- Solution 1: Change one of the two elements linked by the interface and the compatibility should be checked again. This solution is called “component change solution”.
- Solution 2: Decompose the interface into an Interface-ComponentInterface structure. The compatibility of the two new interfaces should be checked. This solution is called “interface decomposition solution”. Fig. 3.21 illustrates these two kinds of solutions. In this example, a simple mechatronic system (System) is decomposed into two components (Component 1 and Component 2) and an interface (Interface 1). However, when the designs of the two components have been finished by the designers, the compatibility test result of the interface (Interface1) indicates that both components are incompatible with each other (Fig. 3.21 (a)). Fig. 3.21 (b) shows the component change solution. The Component 2 can be redesigned and replaced by the Component 3. The compatibility of the interface (Interface 1) should be then checked again. Fig. 3.21 (c) shows the second solution, interface decomposition solution. A new component (Component 3) can be added between the two components and two new

PROPOSITIONS

interfaces (Interface 1.1 and Interface 1.2) will be created accordingly. The compatibilities of the two new interfaces should be checked.

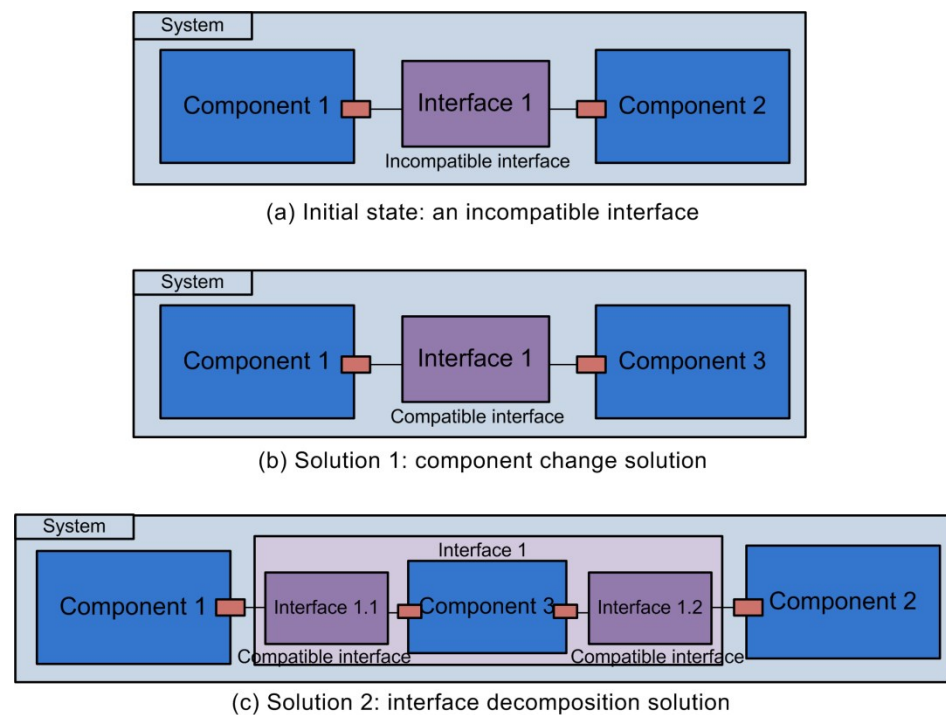
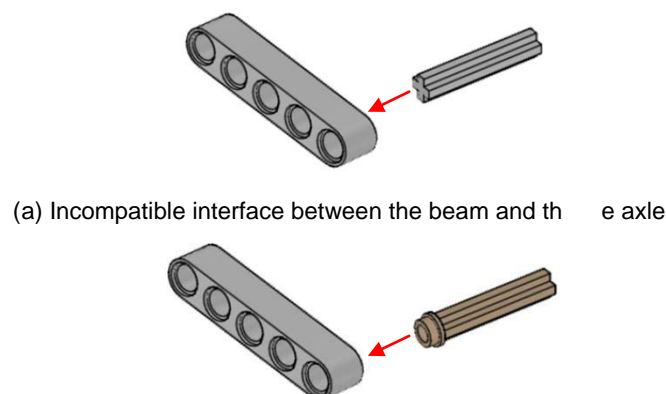


Figure 3.21 Solutions to solve incompatible interfaces

Figure 3.22 shows an example of the solutions to incompatible interface by using LEGO technic pieces. As shown in Figure 3.22 (a), a beam and an axle cannot be connected directly due to their different geometric shapes of the connection points (i.e., ports). Figure 3.22 (b) shows the component change solution to solve the incompatibility problem of the interface between the beam and the axle. The axle can be replaced by an axle with stud so that the two technic pieces can be perfectly connected with each other. The interface decomposition solution is shown in Figure 3.22 (c). The incompatible interface can be further decomposed into an interface-component-interface structure. A new component - a connector peg with bushing can be added and the beam and the axle can be connected by using the connector.



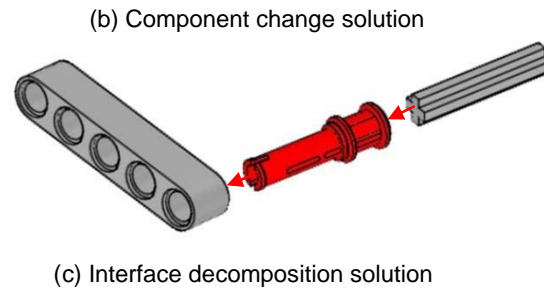


Figure 3.22 Example of solutions to solve incompatible interfaces

3.1.5 Synthesis of multi-disciplinary interface model

The multi-disciplinary interface model is introduced in this section to answer the research questions on design data. Interface classification is represented by the attributes of class **Interface** in the multi-disciplinary interface model. It provides a standard representation for the interfaces defined by project teams from different disciplines. During the design process, the compatibility of the interfaces should be checked by the method **compatibility()** so that the different sub-systems can integrate correctly and the designers can eventually ensure the multi-disciplinary integration among design teams. Macro level interface can be therefore achieved by multi-disciplinary interface model. In addition, the micro level interface can be also achieved by implementing the multi-disciplinary interface model. The attributes **direction** and **visibility** are used to describe one port. The attribute **direction** holds a definition on which one is the master and which one is the slave of the two elements linked by the interface. The attribute **visibility** defines how the parameter and document linked with one port can be accessed. With the support of the two attributes, the multi-disciplinary interface model can greatly help the project teams to collaborate or coordinate with each other during the design process of mechatronic systems. Thus they are both considered as an effective support for the achievement of micro level interface.

The first proposition on multi-disciplinary interface model proves to be an effective support for the design of mechatronic systems and can well answer the research questions on macro level interface and micro level interface. However, it cannot be used directly to manage the design process. In order to answer the research questions on the macro level and micro level collaboration during the design process of mechatronic systems, a new design methodology based on the multi-disciplinary interface model is proposed.

□ 3.2 Design methodology based on multi-disciplinary interface model

In this sub-section, the design methodology based on a multi-disciplinary interface model will be introduced. In line with the practices of SE, an extended V-model is used in the proposed design methodology. It starts with clarification of requirements on the entire system and ends with a uservalidated system. The design phases are detailed in this extended V-model, where individuals can structure design sub-tasks and proceed and react in unforeseen situations. The multi-disciplinary interface model is used during the whole design process to support the macro level and micro level collaboration and to help the project teams to achieve the multi-discipline based integrated design. In order to illustrate the second proposition more clearly, the thesis use the design process of the robot based on the assembly of LEGO Mindstorm to illustrate every design phases of the proposed design methodology. This LEGO robot is designed by the team of UTC for the RobAFIS 2014 competition which has been organised every year since 2006 by AFIS (Association Française d'Ingénierie Système), the French chapter of INCOSE (Tucoulou & Gouyon, 2013). The details of the proposed design methodology will be presented hereafter.

3.2.1 General design process

From the SE perspective, the proposed design methodology adopts the Vmodel to develop a practice-oriented guideline for the systematic development of mechatronic systems (Figure 3.23).

The left branch of the extended V-model represents the system design subprocess and is described with qualitative models. After specifying all requirements for the whole system, the sub-functions and sub-systems are defined. Three design phases are identified during the system design subprocess: Requirements specification phase, Functional modelling phase and architectural definition phase.

The discipline-specific design sub-process is presented at the bottom of the extended V-model. The objective of the discipline-specific sub-process is to obtain the physical elements of the system such as hardware components or software code. The sub-systems have a very discipline-specific character and are developed simultaneously by the different project teams. The models defined by the teams from different disciplines are quantitative for the most part.

PROPOSITIONS

The right branch of the V-model represents the system integration subprocess. The objective of the system integration sub-process is to test the performance of the integrated system and check whether the system realises the proposed function and fulfil all the requirements proposed before. The proposed multi-disciplinary interface model will be used to support the whole design process. The compatibility method of the proposed multidisciplinary interface model can help the designers to guarantee the right integration of components designed by the teams of different disciplines during the discipline-specific design sub-process. Moreover, the multidisciplinary interface model can support the architectural definition phase and help the designers to improve the architecture of mechatronic systems. During the architectural definition phase of the system design sub-process, the multidisciplinary interface model can help the designers to achieve a complete architecture of a mechatronic system; multi-disciplinary. The details of each design phase will be presented in the following sub-sections.

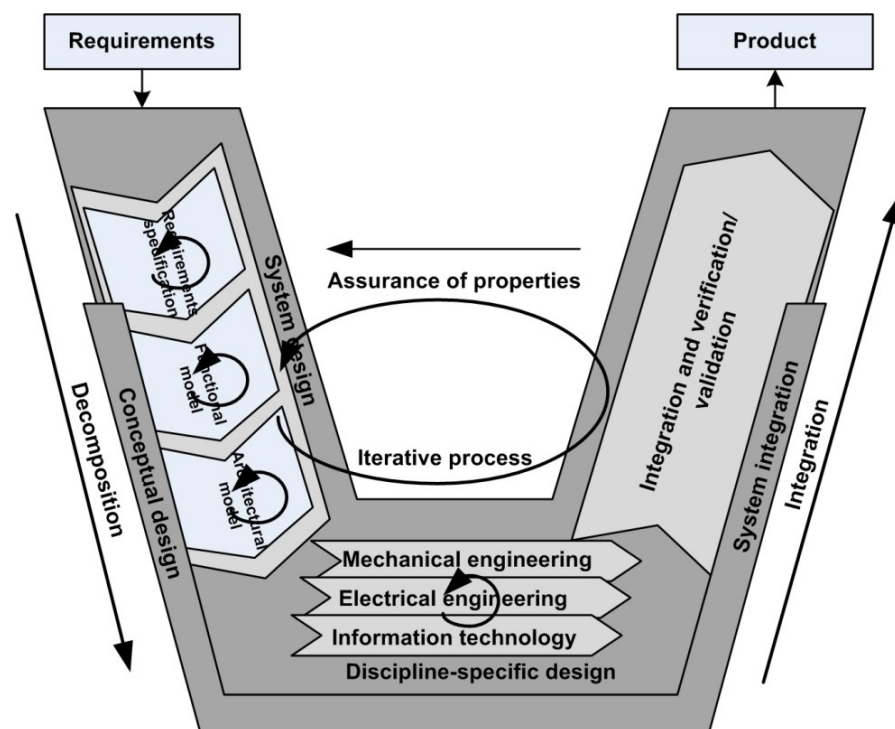


Figure 3.23 General design process: an extended V-model

3.2.2 Design phases

PROPOSITIONS

This sub-section presents the details of each design phases of the proposed methodology. The multi-disciplinary interface model is applied in the entire design process to help the designers to achieve a multi-disciplinary integration.

3.2.2.1 Requirement specification phases

The requirements specification is derived from all requirements on the mechatronic system, and these requirements can provide initial information about what is required by the customers. Requirements can be applied to the overall system, every single sub-system (or component) and the interconnection between two sub-systems. Therefore requirements can be detailed by decomposing into further sub-requirements, thus creating a hierarchy of requirements.

A hierarchical structure is necessary in order to distinguish between different system levels and is therefore closely connected to the management of requirements at different levels. Many mainstream modelling languages (e.g. SysML) can be used to support the requirement specification.

The RobAFIS 2014 Competition proposes the list of requirements to be fulfilled by the multi-usage robot that is configurable in order to achieve three different missions. Each configuration, named with the generic term

Polyval'IS system, uses a mobile, autonomous and common platform

Universal'IS plus a specific device to achieve a specific mission.

One of the three configurations, Transpal'IS, will be taken as an example to represent the functional requirements (the other two configurations are Transcont'IS and Tract'IS). Transpal'IS moves automatically from its parking stand, following an imposed path, to an area that is marked with a yellow road marker, and then the operator manually operates it to shifts a pallet from a storage area A towards a destination area B.

According to the requirements specification of the proposition, the hierarchical structure of the requirements for the LEGO robot can be obtained. Considering the design of the robot, the general functional requirement proposed by the customers is that “the robot should carry out three different missions”. In order to fulfil this requirement, hierarchical structure of the requirement should be specified to detail this general requirement.

Once the requirements are placed in groups, they can be considered accordingly in specific sub-functions.

3.2.2.2 Functional modelling phase

A functional modelling phase refers to the phase for modelling and specification of the functional solutions. Functional modelling plays a significant role during the system design process because it is considered as a bridge between the customers' needs and the mechatronic system. On the one hand, the functions and sub-functions proposed in the functional modelling phase are used to satisfy the customer's requirements. On the other hand, the system architecture is decided based on this functional model.

In this design phase, the functional model of mechatronic systems should be defined. It must answer the specified requirements and thus provide the basis for deriving the functional structure. Hierarchical structure can be also used for functional modelling. If it is assumed that a complex mechatronic system comprises a certain number of elementary functions, the functional structure as cooperation among these elementary functions should be taken into consideration. A single elementary function is characterized by using primarily one clearly defined effect (e.g. physical, chemical or biological), which may be considered as indivisible within the set of functions. An architectural model is then formed by grouping sub-functions from the functional model in order to realise the proposed functions.

Taking the functional model of the robot as an example, when the requirements specification has been finished, the design process enters into the functional modelling phase. The hierarchical functional model should be proposed according to the requirements specification. Attention should be paid to the consistency between the requirements specification and the functional model, which means that every requirement should be fulfilled by one or more sub-functions. For example, the requirement "Power should be supplied to the robot" can be fulfilled by the sub-function "Supply the power".

3.2.2.3 Architectural definition phase

After the functional modelling phase, the designers should find the subsystems which can embody the proposed sub-functions. In other words, these sub-systems should exhibit decomposed sub-functions.

The decomposition process is based on the proposed multi-disciplinary interface model and should be applied recursively. In the design methodology, hybrid-hierarchical architecture is adopted by the decomposition method presented, which

means that both the component decomposition and the interface decomposition have been described (Bloebaum, Hajela, & SobieskiSobieszczanski, 1993).

As introduced before, every sub-system should be recursively decomposed during the decomposition process. However, there is always a research question, that is, what are appropriate hierarchies and granularities for architectural models of mechatronic systems? In order to answer this question, the term “mechatronic module” will be firstly introduced. A mechatronic module is defined as a mechatronic sub-system at the lowest hierarchical level of mechatronic system and is indivisible within the set of mechatronic sub-systems (Hehenberger et al., 2010). Considering the concept of mechatronic module, two steps to realise the component decomposition are proposed. The first step is called in this thesis the step of “system-module”, in which the mechatronic system is decomposed into mechatronic modules. In this step, designers should firstly consider their design experiences to propose the architectural model. For example, can the mechatronic module be implemented based on past design or standard component in handbooks? If certain mechatronic modules can be perfectly realised by the past designs or the standard components, further decomposition is not necessary. It implies that it is not necessary to decompose the system until the discipline-specific components. But such past designs or standard components are not always available. In order to solve this problem, the second step of the component decomposition method is proposed, which is called the step of “modulecomponent”. In this step, the mechatronic module should be further decomposed until the discipline-specific components which can be obtained with the standard components, the past designs or the new design which can be carried out by discipline-specific team. During the whole decomposition process, the interfaces among the components should be also clarified and decomposed. During the decomposition process, the interfaces among the discipline-specific sub-systems should be also clarified and decomposed in order to verify whether the sub-systems can be correctly integrated with each other.

By applying the method proposed in this section, the architectural model of the robot can be defined. The designers should firstly decompose the robot into mechatronic modules and try to use the past designs or standard components to realise the functions of these mechatronic modules. For example, a motor can realise the function “supply the driving force to the platform” of the platform driven sub-system. If such standard components or the past designs do not exist, the designers should further decompose the mechatronic module into components.

Before entering the discipline-specific design process, the consistency between each two design phases should be checked because functional models and architectural models are always created and maintained by the designers from different disciplines. As presented before, during the system design process, the functions and sub-functions proposed in the functional models are used to satisfy the costumer's requirements, while the architectural model for the sub-systems is constructed to embody the proposed sub-functions (Figure 3.24). Therefore it is necessary to ensure correctness of such models. Every requirement should be met by one or more functions (or sub-functions), and every sub-function must be realised by one or more sub-systems (in certain case several sub-functions can be realised by one sub-system).

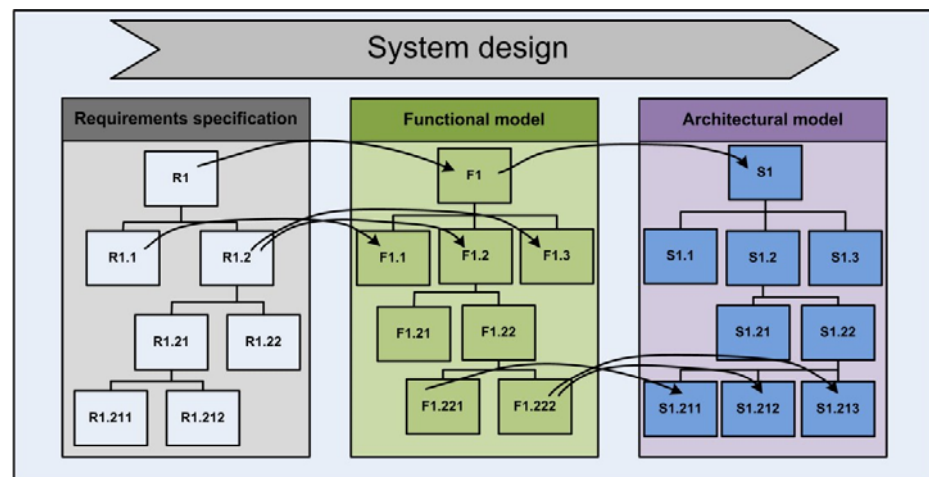


Figure 3.24 Consistency between different design phases during system design process.

As introduced before, the consistency between different design phases should be verified after finishing the system design sub-process. Figure 3.25 shows the consistency in different design phases for the LEGO robot.

PROPOSITIONS

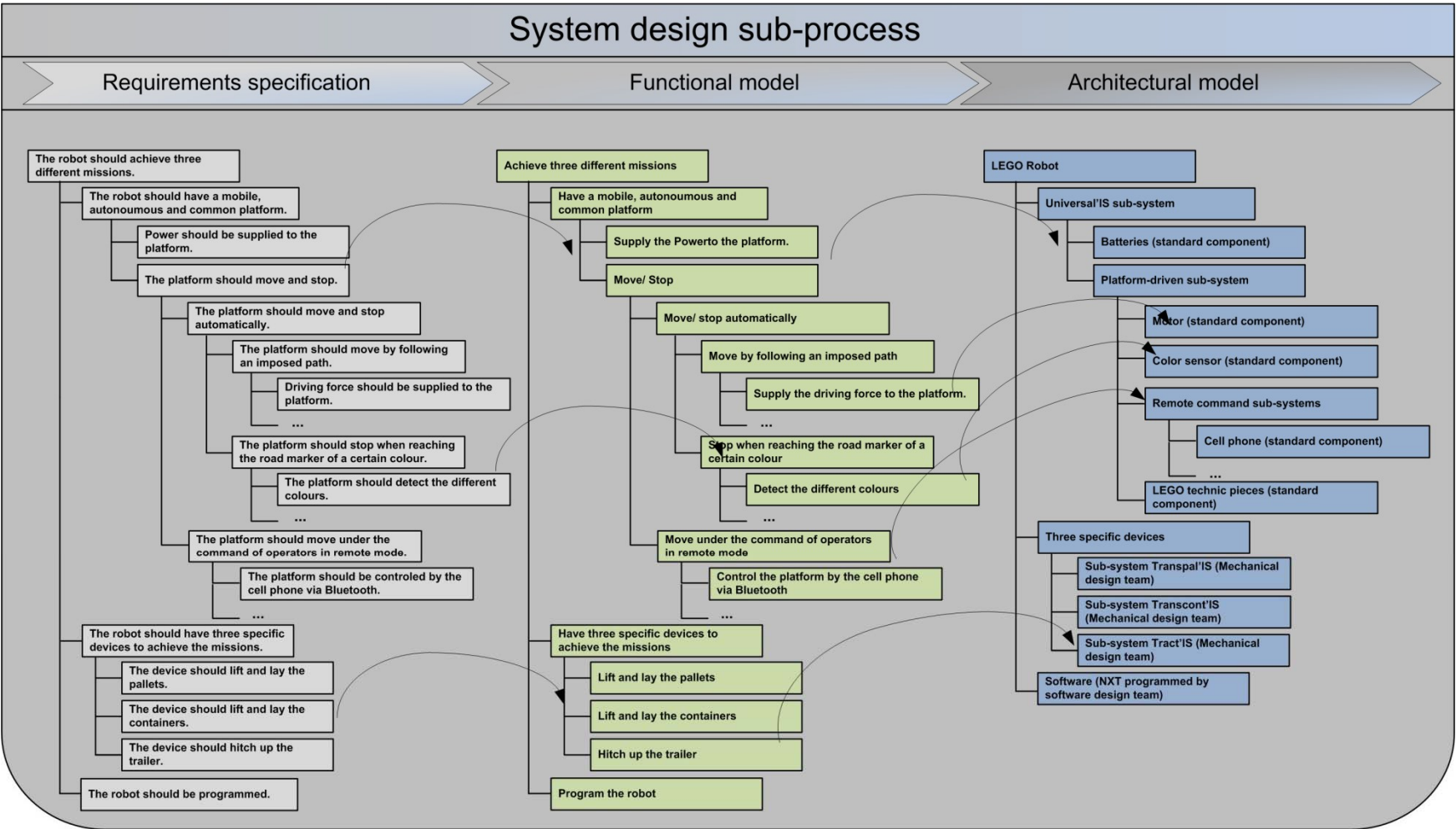


Figure 3.25 Consistency in the three design phases for the LEGO robot

PROPOSITIONS

Chen ZHENG

Université de Technologie de Compiègne

104

After the mechatronic modules are further decomposed into disciplinespecific sub-systems and the interfaces among them, these disciplinespecific sub-systems (with their interface) will be developed by the engineers in different disciplines during the discipline-specific design subprocess.

3.2.2.4 Discipline-specific design sub-process

During the discipline-specific design sub-process, the project teams of each design discipline should develop sub-systems which have been decided in the architectural model phase. Certain components of the discipline-specific sub-systems can be perfectly realised by the past designs or the standard components. If they cannot be realised by any past designs or standard components, a new design embodying the function of this component should be carried on by the project team of one certain discipline. However, current research works pay much attention to the design of components, and the significance of the interface in the discipline-specific design is always neglected. (Zheng, Bricogne, Le Duigou, & Eynard, 2014b).

The design of mechatronic systems requires the multi-disciplinary collaboration of different project teams during the discipline-specific design sub-process. The concurrent design process where the design tasks for different sub-systems (or components) can be carried out by different project teams in parallel. All the sub-systems (or components) designed in the discipline-specific design subprocess should be integrated correctly with each other. Such multi-disciplinary collaboration often leads to iterations during the concurrent design process, because the designers often have to jump back one or more steps to redesign or to tune what they created before to satisfy the integration with the parts designed by other project teams. Therefore the interface compatibility is very important for the design of mechatronic systems, as it can detect the design errors at an early stage, which can greatly reduce the unnecessary iterations and decrease development costs and the time-to-market.

The multi-disciplinary interface model can be used to support the compatibility test. Once the data model of an interface has been instantiated, the designers should check the compatibility of the interface by making use of the compatibility rules stored in the method

compatibility().

The discipline-specific design sub-process of V-model is detailed and used to represent how the multi-disciplinary interface model can support the design process in Figure 3.26. For the sake of clarity, the process in Figure 3.26 is not described in the typical V shape, but the three main subprocess of V-model, system design sub-process, discipline-specific design sub-process and system integration sub-process are represented. The initial state of the activity diagram indicates the beginning of the system design process. By analysing the customer's requirements and exhibiting the required functions, the system engineers propose the principle architecture of the mechatronic system. Such architecture should be decomposed recursively into the sub-systems and interfaces until the standard components or the components which can be designed by the discipline-specific project teams. The project teams of different disciplines can define the interfaces by instantiating the proposed interface model. Due to the design maturity during the discipline-specific design sub-process, the information stored in the component and interface model should be refined. The loop in Figure 3.26 shows such refinement process. Once the model of an interface has been instantiated, the interface compatibility should be checked by using the method **compatibility()** in order to guarantee the different components integrate correctly. If the components prove to be incompatible with each other, the iterative processes should be carried on.

Two activities can be adopted to solve the incompatibility problem of interfaces. The first activity means that the component change solution is chosen. Because one component should be redesigned, the design process will go back to the beginning of discipline-specific design sub-process.

Interface decomposition solution is used in the second solution. Due to the decomposition of the incompatible interface, the design process will return to the architectural definition phase, and the system architecture can be improved.

Compared with the interface decomposition solution, the component change solution is simpler to operate by the designers. However, in a complex mechatronic system, one component may link to others through several interfaces. After the change of such component in order to solve the incompatibility problems of one interface, other interfaces linked to this component which proved to be compatible before may become incompatible. Such conflict may always exist during the design process.

The interface decomposition solution demands the designers to further broken down the incompatible interface into an interface-componentinterface structure. It can help the designers to improve the architectural model of the mechatronic system. In most cases, the architectural model of a mechatronic system cannot be completely decided by the designers during the system design sub-process due to some reasons, such as the lack of design experiences, the unforeseen incompatibility between two sub-systems and etc. When the interface decomposition solution is adopted, the design process should go back to the architectural definition phase. Therefore the interface decomposition solution can be used as an effective support to help the designers to refine the architecture of the mechatronic system. Like the component change solution, incompatible interfaces between the newly added component and other elements of the system may be created when adopting the interface decomposition solution. Therefore, the designers should carefully select the solutions to solve the incompatibility problems, and the compatibility of the interfaces which affected by the changed (or newly added) component should be rechecked after the solutions to incompatibility problem have been adopted.

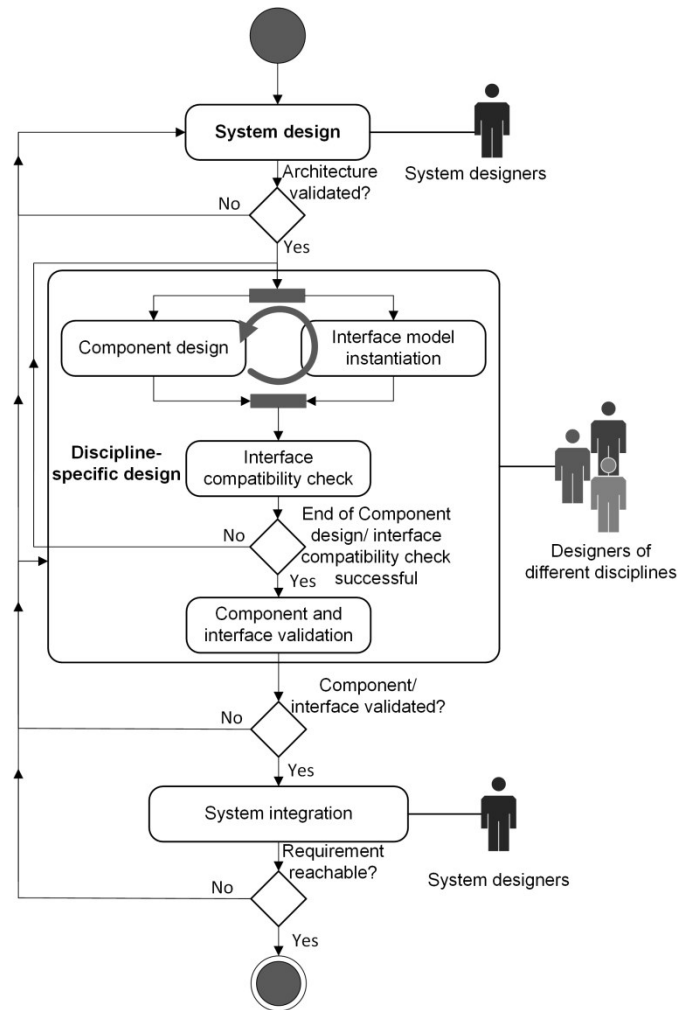


Figure 3.26 UML activity diagram for the discipline-specific design sub-process based on multi-disciplinary interface model

An instance of the multi-disciplinary interface model is created here by making use of the interface between the rechargeable battery and the EV3 brick of LEGO Mindstorm. Concerning the transfers through this interface, the electrical energy (voltage) can be transferred through this interface, which is considered as the primary type of transfer the interface through which a primary type of transfer is transferred while other types are transferred as subsidiary transfer. Meanwhile, the geometrical connections considered as this interface's subsidiary types of transfers exist between the two components in order to have a better physical integration. Such interface should be further decomposed and refined into energy sub-interface and geometric sub-interface (Figure 3.27).

Figure 3.27 is only an example to illustrate how the designers can instantiate the proposed multi-disciplinary interface model for an interface existing in LEGO Mindstorm, and the documents linked to the ports of the energy interface (a data sheet in PDF). The parameters possessed by the ports of the geometric interface are not shown in this figure for the sake of clarity.

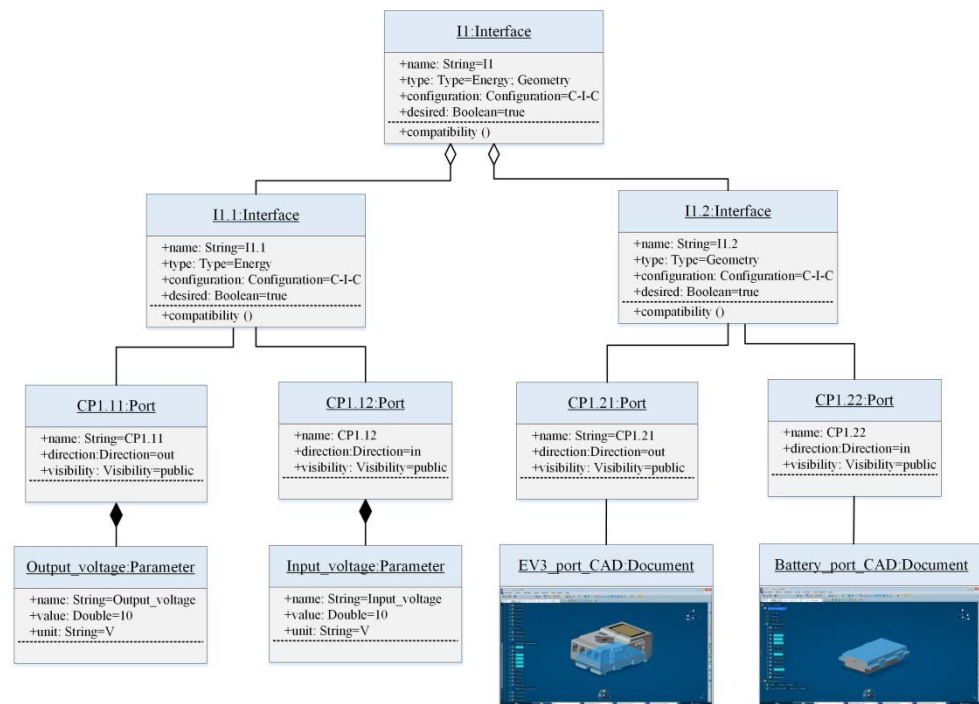


Figure 3.27 Instance of the interface between the rechargeable battery and the EV3 brick

3.2.2.4 Integration sub-process

After the design of components and interfaces have been validated at the end of discipline-specific design sub-process, the system engineers should test the performance of the mechatronic system and check whether such integrated system realises the proposed function and satisfies all the requirements specified before. If the mechatronic system has to be improved, the initial phase will be repeated.

3.2.3 Achievement of macro level and micro level collaboration

Previous sub-sections have introduced the proposed methodology for the design of mechatronic systems. It uses the extended V-model based on the multi-disciplinary interface model to support the whole design process. The research objective is to achieve a multi-discipline based integrated design of mechatronic systems in which the “process-based problems” and the “design data-related problems” should be answered by the product model and design method respectively. As for the design data-related problems, an extension of existing product models (the multi-disciplinary interface model) in which the macro level and micro level interface can be achieved simultaneously is proposed. For the process-based problems, an extended V-model based on the multi-disciplinary interface model is proposed. Thus the designers can achieve the macro level and micro level collaboration.

The multi-disciplinary interface model provides a standard representation for the interfaces, which can be used to ensure the right integration of the different sub-systems designed by the project teams from different disciplines during the discipline-specific design sub-process. In addition, the proposed method **compatibility()** and the two solutions which is used to solve the incompatibility problems can help the designers to gradually improve the architectural model of a mechatronic system in order to achieve a complete and appropriate architecture during the system design sub-process. The proposed method **compatibility()** and the two solutions to the incompatibility problem will test the compatibility among the subsystems (or components) during the discipline-specific design subprocess. They help the designers to decide whether the design process can enter the next phase and to test whether the system realises the proposed function and satisfies all the requirements proposed before, otherwise the design process should go back to the previous design phases. In summary, with the support of the multi-disciplinary interface model, the integration of the components designed by the project teams of different disciplines is ensured and the collaboration among the design teams is achieved. As the interface compatibility test can detect the design errors at an early stage, unnecessary iterations can be greatly reduced so that development costs and the time-to-market are decrease.

The design methodology of mechatronic system also focuses on the collaboration of the different engineers or designers, such as communication among designers, data sharing and exchange. Such collaboration among the individuals is called in this micro-level collaboration. On the one hand, in this extended V-model, the specific design phases are detailed, where individual designers can structure design sub-tasks and proceed and react in unforeseen situations. On the other hand, the multi-disciplinary interface model support micro level interface which can help to achieve the micro level collaboration. The attributes **direction** holds a definition on which one is the master and which one is the slave of the two elements linked by the interface. The attribute **visibility** defines how the parameter and document linked with one port can be accessed. According to the proposed multi-disciplinary interface model, an IT platform based on 3DEXPERIENCE Platform which can realise the proposed design methodology has been developed to achieve the communication, data sharing and exchange among the designers. Therefore, the micro level collaboration can be achieved by the design methodology.



3.3 Summary

This chapter presents the propositions - the multi-disciplinary interface model and the design methodology based on the proposed interface model, which can be used during the design process of mechatronic systems to help the designers achieve the multi-discipline based integrated design. The multi-disciplinary interface model can provide the designers with a standard representation for the interfaces defined by project teams from different disciplines. The interface classification and the interface compatibility rules are respectively represented as the UML classes, relationships, attributes and methods in the multi-disciplinary interface model. The interface classification provides much more details of an interface to the designers and helps them to avoid the confusion by the misuse of interfaces. The interface compatibility rules are proposed to guarantee the different elements integrate correctly and reduce the unnecessary iterations in subsequent design steps during the design process. The proposed

multi-disciplinary interface model can be used as an extension of existing product models. LEGO Mindstorms is used for a better understanding of the first proposition on multi-disciplinary interface model.

The second proposition of the thesis is a new design methodology based on the proposed multi-disciplinary interface model. This design methodology uses an extended V-model to describe the generic process for the design of mechatronic systems from the clarification of all requirements on the whole system to a user-validated system. In this extended V-model, the specific design phases are detailed, where individual designers can structure design sub-tasks and proceed and react in unforeseen situations. The multi-disciplinary interface model is used to support the macro level and micro level collaboration in order to help the designers achieve the multi-discipline based integrated design. The design of LEGO robot is proposed in order to illustrate the second proposition on design methodology more clearly.

Next chapter will present a case study detailing the design process of a three dimensional (3D) measurement system to demonstrate the proposed multi-disciplinary interface model and design methodology.

Case study

The case study chosen to demonstrate the propositions in this chapter is a 3D measurement system. This measurement system is designed for reconstruction of the measured object's surface based on optical measurement. Because this 3D measurement system is considered as a mechatronic system integrating synergistically the electrical/electronic system, mechanical parts, information processing and optical technology, the system requires a multi-discipline based integrated design.

In this chapter, the demonstrator implementing the propositions based on 3DEXPERIENCE Platform will be first presented. The 3D measurement system will be subsequently introduced. The propositions will be then demonstrated by the design of the 3D measurement system with the demonstrator. The design of

the 3D measurement system will show that the propositions of the multi-disciplinary interface model and the design methodology can help the project teams to achieve the multi-disciplinary integration. Summary of the case study will be finally given.

□ 4.1 Demonstrator development

A demonstrator based on 3DEXPERIENCE Platform 2013x is developed to implement the propositions. This sub-section will firstly introduce the VPM Functional Logical Editor workbench in 3DEXPERIENCE Platform which implements the interface classification and the interface data model. The interface compatibility rules are implemented by using CATIA

Knowledge Based Engineering (KBE) solution named Knowledgeware. As for the design methodology, the Requirement engineering view, Functional view, Logical view of the RFLP approach in 3DEXPERIENCE Platform are used to represent the three phases of system design subprocess of the design methodology.

4.1.1 Introduction of 3DEXPERIENCE Platform

The VPM Functional Logical Editor workbench of 3DEXPERIENCE Platform (Figure 4.1) manages the system within the entire development lifecycle from requirements, functional, logical, and physical definition to simulation. In one window, the user can access the requirements of the product, its functional decomposition, its logical architecture and its physical definition (Kleiner & Kramer, 2013). Next sub-section will introduce how to implement the proposed multi-disciplinary interface model by using the VPM Functional Logical Editor workbench of 3DEXPERIENCE Platform.

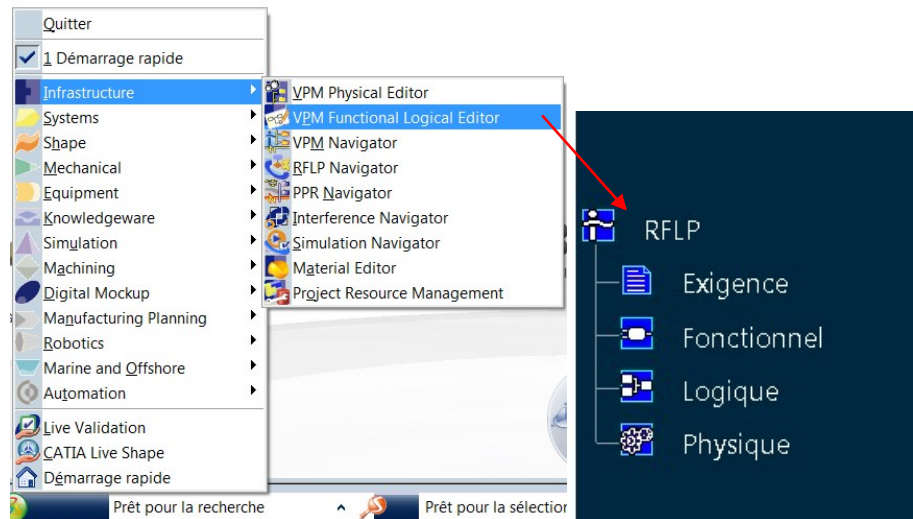


Figure 4. 1 VPM Functional Logical Editor

4.1.2 Implementation of the design methodology

The RFLP approach reviewed in Chapter 2 has been implemented as a basis for Systems Engineering in 3DEXPERIENCE Platform environment. In the system design sub-process of the design methodology, requirements specification phase, functional modelling phase and architectural definition phase can be respectively demonstrated by the Requirement engineering view, Functional view, Logical view of RFLP approach implemented in 3DEXPERIENCE Platform.

- Requirements specification: the first phase of the design methodology is requirements specification. By analysing the customers' needs, the requirements specifications can be imported directly into ENOVIA from Microsoft Word or Excel documents, for example. Once the requirements specification is created in the ENOVIA environment, such requirements are managed in a hierarchical structure simultaneously available in the Requirement engineering view of 3DEXPERIENCE Platform (Figure 4.6).

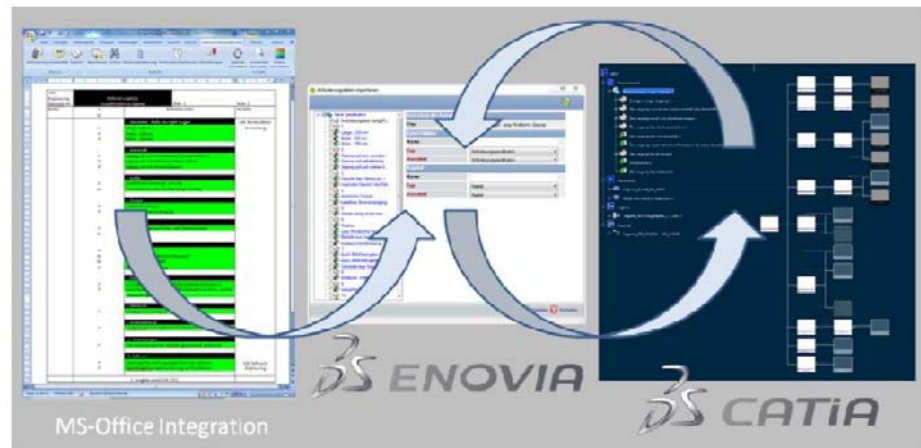


Figure 4.2 Requirements specification using Word, ENOVIA and 3DEXPERIENCE Platform (Kleiner, 2013)

- Functional modelling phase: the VPM Functional Logical Editor can be used to demonstrate the functional modelling phase of the design methodology in the Functional view. The functions are derived from requirements specification, and thus expand the requirements model. The main and sub-functions can be displayed graphically and structured hierarchically in 3DEXPERIENCE Platform.
- Architectural definition phase: in the Logical view of RFLP approach implemented in 3DEXPERIENCE Platform, the logical architecture of the system can be defined using logical components and the relations between them. However, the environment, the interface and the port, proposed in multi-disciplinary interface model, have not been detailed in the Logical view. In order to demonstrate the design methodology, the Logical view has been further developed. As detailed in Section 4.1.2, the environment, the interface and the port can be represented by making use of the logical component and the Knowledgeware of 3DEXPERIENCE Platform.

During the discipline-specific design sub-process of the design methodology, the interface compatibility rules created by 3DEXPERIENCE Platform Knowledgeware can help the designers to guarantee the subsystems assemble

correctly and to ensure the multi-disciplinary integration among different project teams. Figure 4.7 shows the interface compatibility rules and the test result.

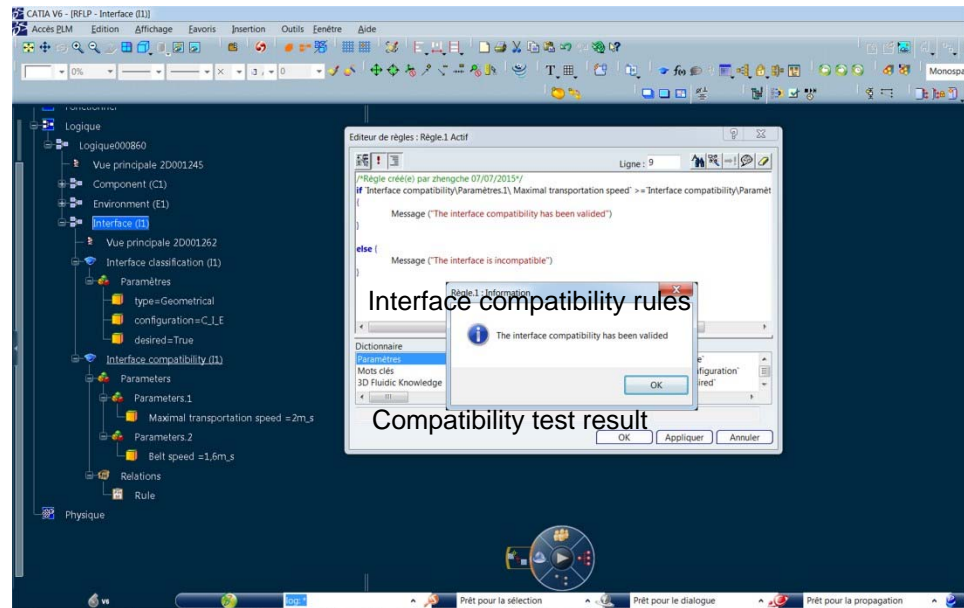


Figure 4.3 Interface compatibility rule and the test result

By now the thesis has shown how the proposed design methodology can be implemented in 3DEXPERIENCE Platform. Next sub-section will present the implementation of the multi-disciplinary interface model in 3DEXPERIENCE Platform.

4.1.3 Implementation of the multi-disciplinary interface model

The architectural model of mechatronic systems can be defined by the VPM Functional Logical Editor workbench because it enables users to create, modify and delete logical data structures and to link those elements with implemented relations.

However, in the VPM Functional Logical Editor workbench, the architectural model of mechatronic system can only be represented by the components, the environments and the interfaces are not defined. A port is represented as a feature of a logical model that is used to define the input/output of functions or to provide connections between logical components, but the attributes of ports presented in

Chapter 3 have not been described in the port proposed in 3DEXPERIENCE Platform. In order to implement the proposition of multi-disciplinary interface model, the VPM Functional Logical Editor workbench is further developed so that the classes **Interface**, **Environment** and **Port** and the relationship among them can be represented in the demonstrator.

- **Component:** components (or sub-systems) of mechatronic systems can be represented by the logical components in the VPM Functional Logical Editor workbench. The sub-systems or components in the Logical category can be created. The 2D representation of the component is shown in Figure 4.2. In the demonstrator, the component is represented by blue box.

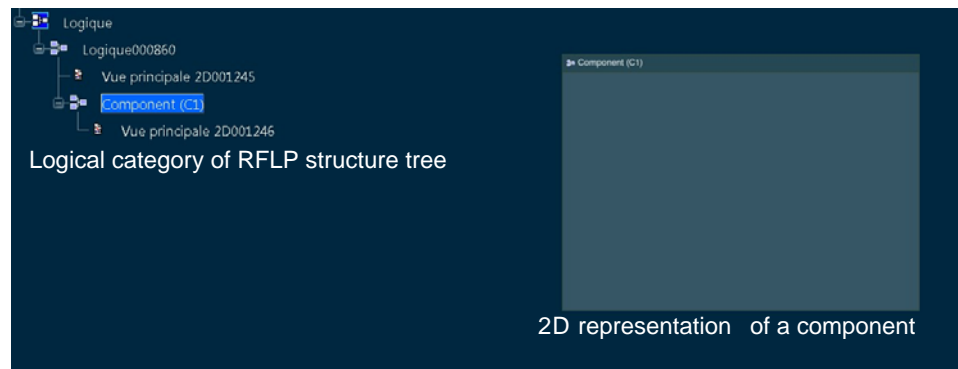


Figure 4.4 Component representation in 3DEXPERIENCE Platform

- **Environment:** 3DEXPERIENCE Platform does not provide the representation of environment. Logical component proposed in the VPM Functional Logical Editor workbench is used to represent the environment. As shown in Figure 4.3, the environment is represented by green box.



Figure 4.5 Environment representation in 3DEXPERIENCE Platform

- Interface:** the interface can be generally represented by the connections among logical components in the VPM Function Logical Editor workbench, but the details, such as the interface classification and interface compatibility rules, are not shown in such connections. Like environment, the logical component is used to represent the interface. As for the details, Knowledgeware solutions of 3DEXPERIENCE Platform is used to describe the interface classification and interface compatibility. Knowledgeware delivers a set of tools that helps designers to define, reuse and share their know-how throughout the extended enterprise. In other words, Knowledgeware supports the lifecycle management of the knowhow that is reused in engineering and manufacturing activities. The Set of Parameters enables the users to gather the parameters in a set and to describe the interface classification as shown (Figure 4.4). **Type**, **Configuration** and **Desired/undesired** proposed as three attributes of one interface are represented as the names of parameters, and the values of parameters are used to describe the details of classification. The rules, which are considered as a set of instructions based on the context and described by the rule instructions, are used to define the interface compatibility rules.

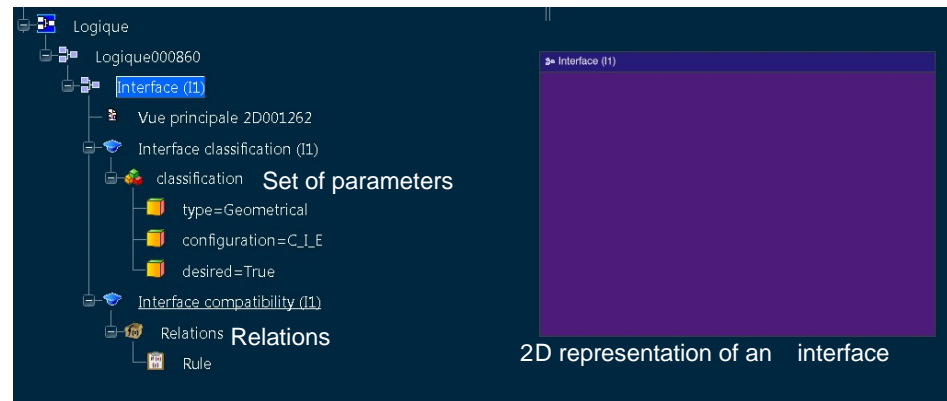


Figure 4.6 Interface representation in 3DEXPERIENCE Platform

- **Port:** according to the proposed multi-disciplinary interface model, component, environment and interface can all have ports. The attributes “**direction**” and “**visibility**” are proposed to describe one interface.

In 3DEXPERIENCE Platform, a port is an entity exposing the different data handled by a logical component. The three types of directions represented by the attribute “**direction**” of class **Port** in the multi-disciplinary interface model In, Out, In/out can be found in 3DEXPERIENCE Platform. In fact 3DEXPERIENCE Platform provides four types of direction: *In*, *Out*, *In/Out* and *No direction*. *No direction* is proposed to represent a port which exposes a variable. Such *No direction* port is used to model a mathematical equality between exposed variable (Kleiner & Kramer, 2013). However, only using a variable to describe the parameters linked with the port is not enough, so in the demonstrator of the proposition, such *No direction* port is not used. 3DEXPERIENCE Platform Knowledgeware is used to create a set of parameters for that port (Figure 4.5).

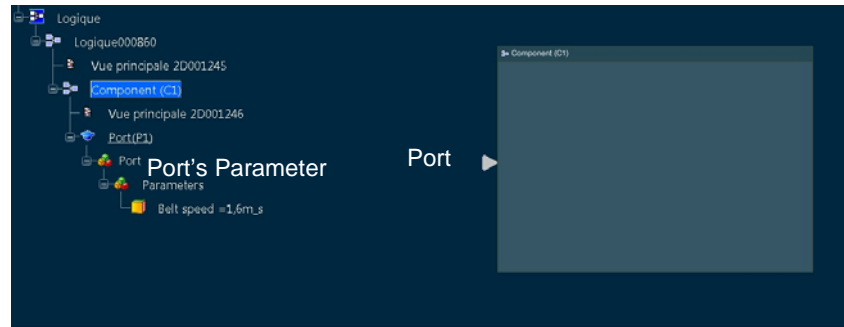


Figure 4.7 Representation of port and its parameters in 3DEXPERIENCE

Platform

The second attribute for the port in the multi-disciplinary interface is **visibility**. This attribute describes how the port can be accessed. The authorized values are *public*, *protected* and *private*. This attribute describes how the port can be accessed. However, the port is only defined as an entity possessed by a logical component in 3DEXPERIENCE Platform, and the visibility is not been represented. Thus, the three types of design projects provided by 3DEXPERIENCE Platform are used to represent this attribute directly. In 3DEXPERIENCE Platform, public project means that data can be accessible by any users during the design process. Protected project can be accessible but it becomes unchangeable. A project carrying the private property is accessible only from affected users. Such three types of projects can be perfectly used to represent the corresponding types of ports' visibilities. If a port is defined as a *public* port, this port and its associated documents should be considered as the items of a public project. A *protected* port and its associated document linked should be put into a *protected* project. The *private* port and its document are accessible only by those who design it; therefore they should be stored in a private project.

- **Parameter and document:** according to the definition of the multidisciplinary interface model, the classes **Parameter** and **Document** are of great significance to describe the port linked with them. The class **Parameter** specifies the parameter related to the port which can be quantified. As above mentioned, 3DEXPERIENCE Platform

Knowledgeware is used to create a set of parameters to represent the Class **Parameter** linked to the port. If parameters stored in the Class **Parameter** cannot fully represent a port, the class **Document** is used to store the documents which can describe the port. The CAD document can be adopted as the useful document to represent the geometrical details of a port. In 3DEXPERIENCE Platform, such CAD document used to describe the port can be defined and stored in VPM Physical Editor. The link between the CAD document in VPM Physical Editor and the port modelled in VPM Functional Logical Editor has been defined in 3DEXPERIENCE Platform, so the designers can refer one port's CAD document directly.

The demonstrator based on 3DEXPERIENCE Platform and implementing the propositions on design methodology and multi-disciplinary interface model has been presented in this sub-section. In next sub-section the demonstrator will be adopted to clarify the propositions by using the design process of 3D measurement system.

□ 4.2 3D measurement system

The design of 3D measurement system is chosen as a case study in this chapter to demonstrate the proposition. This system is a typical mechatronic system which integrates mechanical parts, electrical/electronic components, and information processing. And meanwhile, the optical technology also plays a significant role in this system.

The 3D measurement system can operate on 2 modes, which are called Active mode 1 and Active mode 2 (Figure 4.8). The two measurement modes are similar to each other, so here Active mode 1 will be taken as an example to introduce the principle of the 3D measurement system. The principle of the system on Active Mode 1 is shown in Figure 4.8 (a). A fringe patterns (Figure 4.8 (c)) is generated and projected on the measured object's surface. The deformed pattern (Figure 4.8 (d)) reflected by the object's surface is then captured and analysed. By comparing the original fringe patterns and the deformed image, the depth information of the

measured surface can be given. The measurement mode can be switched by changing the path of light (Dupont, Hou, Lamarque, & Redarce, 2013).

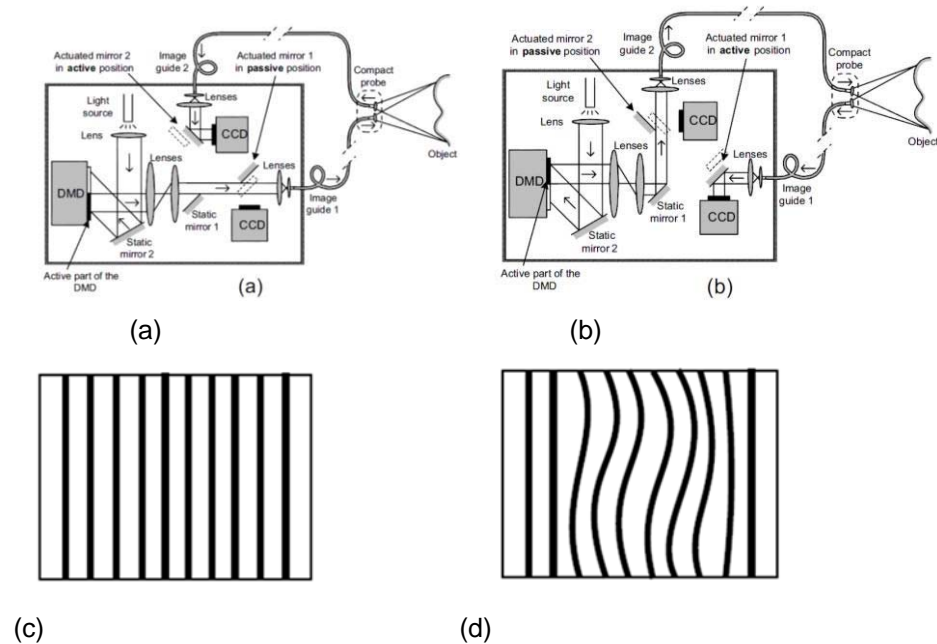


Figure 4.8 Principle of the measurement mode: (a) Active mode 1 (b) Active mode 2 (c) Projected pattern (d) Deformed pattern

4.2.1 System design sub-process

The first phase of the system design sub-process focuses on specifying the requirements specification. The general functional requirement proposed by the customers is that “the 3D measurement system should measure the object’s surface”. In order to fulfil the requirement of measurement, hierarchical structure of the requirement should be defined to detail this general requirement in the corresponding phase of the micro level process. When the requirements specification has been finished, the macro level process enters into the functional modelling phase. The functional modelling can be determined by analysing the requirements specification. A hierarchical functional model should be proposed according to the requirements specification. For example, the requirement “A

fringe pattern should be projected on the object's surface" can be fulfilled by the subfunction "Project the fringe pattern".

The last phase of the system design sub-process is the architectural definition phase. According to the functional model, the system's general architecture formed by sub-systems or components can be defined. The 3D measurement system can be generally decomposed into six sub-systems. The pattern projection sub-system (C1.1) projects the fringe patterns on the measured object, and the deformed image reflected by the measured object is received by the deformed image reception sub-system (C1.2). The original fringe patterns and the deformed image will be compared and analysed by the 3D image reconstruction sub-system (C1.3). The measurement modes can be switched by the mode switch sub-system (C1.4). The whole system is supported by the mechanical support subsystem (C1.5) while the power is supplied by the power supply sub-system (C1.6). The decomposition process of the system should be applied recursively. The designers can further decompose every sub-system into mechatronic modules and try to use the past designs or standard components to fulfil the functions of these mechatronic modules. For example, the DMD (Digital Micro-mirror Device), considered as a standard component, can realise the function "Generate fringe pattern". The actuator to switch the measurement mode can be realised by the past design developed by other project team. If such standard components or the past designs do not exist, the designers can use the proposed pillar design model for the further decomposition. The pattern projection sub-system (C1.1), for example, should be further broken down into C1.11.light source, C1.12.DMD, C1.13.image guide I and C1.14.compact probe I. The interfaces among the sub-systems should be also clarified in this phase. The multi-disciplinary interface model can be instantiated in this phase because it provides a standard representation of interfaces by using three attributes to describe an interface of mechatronic systems. It also helps the designers from different disciplines to overcome the lack of commonality in interface definitions from different decomposition levels or different disciplines. The instances of interface data model will be shown in Section 4.2.2.

Before entering the detailed design phase, the mapping among the requirements specification phase, the functional modelling phase and the architectural definition phase can be established in order to ensure the consistency during the design process.

Table 4.1 shows the relationship among the requirements, the functions, the sub-systems and the solution principles. The first three columns of this table present the requirements, functions and the sub-systems by which the functions can be fulfilled, while the last column shows the standard components, past designs or the discipline-specific teams which can realise the corresponding sub-systems.

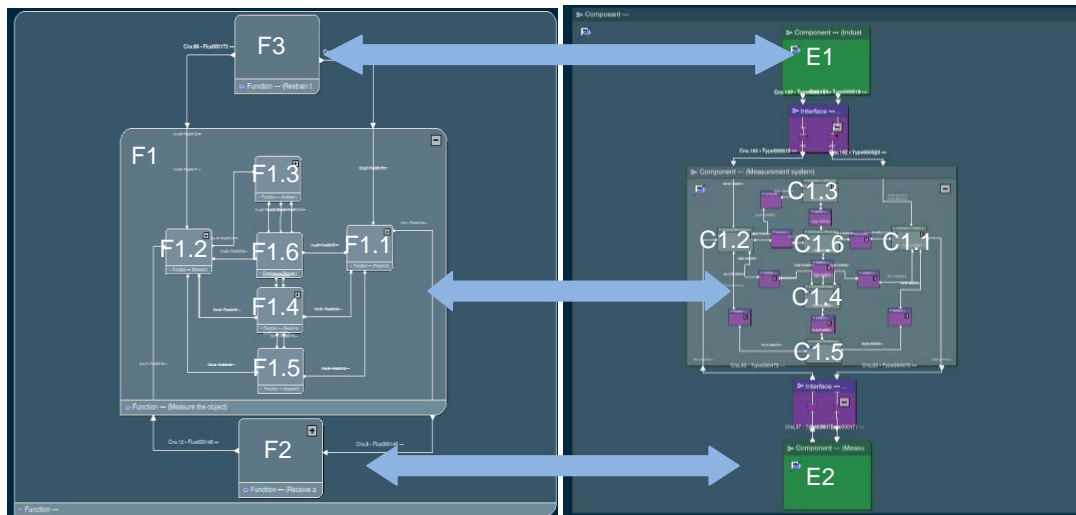
PROPOSITIONS

Table 4.1 Requirements, functions, components and design teams/standard component

Requirement (sub-requirement)	Function (Sub-function)	System (Sub-system)	Solution principles
R1. The system should measure the object's surface.	F1. Measure the object's surface	C1. Measurement system	
R1.1. The system should project the fringe pattern.	F1.1. Project the fringe pattern	C1.1. Pattern projection subsystem	
R1.11. The system should provide the light source.	F1.11. Provide the light source	C1.11. Light source	Standard component
R1.12. The system should generate fringe patterns.	F1.12. Generate fringe patterns	C1.12. DMD	Standard component
R1.13. The system should transmit the fringe pattern.	F1.13. Transmit the fringe pattern	C1.13. Image guide I	Standard component
R1.14. The system should project the fringe pattern.	F1.14. Project the fringe pattern	C1.14. Compact probe I	Optical team
R1.2. The system should receive the reflected light.	F1.2. Receive the reflected light	C1.2. Deformed image reception sub-system	
R1.21. The system should capture the deformed image	F1.21. Capture the deformed image	C1.21. Compact probe II	Optical team
R1.22. The system should transmit the deformed image	F1.22. Transmit the deformed image	C1.22. Image guide II	Optical team
R1.23. The system should store the digital data of the deformed image	F1.23. Store the digital data of the deformed image	C1.23. CCD (Charge-Coupled Device) sensor	Standard component
R1.3. The system should reconstruct the 3D image	F1.3. Reconstruct the 3D image	C1.3. 3D image reconstruction sub-system	
R1.31. The system should store the reconstruction data	F1.31. Store the reconstruction data	C1.31. PC	Standard component
R1.32. The system should analyse the deformed pattern	F1.32. Analyse the deformed pattern	C1.32. Algorithm	Software team
R1.33. The system should show the reconstruction result	F1.33. Show the reconstruction result	C1.31. PC	Standard component
R1.4. The system should switch the measurement mode		C1.4. Mode switch sub-system	
	F1.4. Switch the measurement mode	C1.41. Actuator	Past design
		C1.42. Mirror	Standard component
R1.5. The system should support all the components of the 3D measurement system	F1.5. Support all the components of the 3D measurement system	C1.5. Mechanical support subsystem	Mechanical team
R1.6. The system should supply power for the system	F1.6. Supply power for the system	C1.6. Power supply subsystem	Electrical team
R2. The object should receive and reflect the light	F2. Receive and reflect the light	E1. Measured object	
R3. The environment should restrain the system	F3. Restrain the system	E2. Industrial environment	

PROPOSITIONS

Figure 4.9 shows the hierarchical structure of functional modelling and architectural definition for the 3D measurement system. Figure 4.9 (a) describes the main functions of the measurement system, the measured object and the industrial environment (**F1**.Measure the object's surface, **F2**.Receive and reflect the light and **F3**.Restrain the system) and six basic sub-functions of F1 (**F1.1**.Project the fringe pattern, **F1.2**.Receive the reflected light, **F1.3**.Reconstruct the 3D image, **F1.4**.Switch the measurement mode, **F1.5**.Support all the components of the 3D measurement system and **F1.6**.Supply power for the system). The links between individual sub-functions are used to transfer the data or control flows. Figure 4.9 (b) presents the **C1**.3D measurement system, **E1**.Industrial environment and **E2**.Measured object. Six subsystems of **C1** (**C1.1**.Pattern projection subsystem, **C1.2**.Deformed image reception subsystem, **C1.3**.3D image reconstruction subsystem, **C1.4**.Mode switch subsystem, **C1.5**.Mechanical support subsystem and **C1.6**.Power supply subsystem) and the interfaces among them are designed as the solutions to the functions proposed at the Functional design stage. The red arrows in Figure 4.9 show the mapping between the basic functional model and the architectural model by which the functions can be accomplished.



(a) Functional modelling

(b) Architectural definition

Figure 4. 9 Functional and architectural model for 3D measurement system

4.2.2 Discipline-specific design sub-process

PROPOSITIONS

In order to demonstrate the propositions more clearly, the pattern projection sub-systems (**C1.1**) is chosen to show the discipline-specific design sub-process.

The pattern projection sub-system is one of the most important parts in the 3D measurement system. Figure 4.10 shows the representation of the sub-system with its components in the Logical design stage of 3DEXPERIENCE Platform. A white light source (**C1.11**) illuminates the DMD (**C1.12**) whose power is supplied by the power supply subsystem. The fringe patterns are generated by the DMD and injected into the image guide (**C1.13**). The image guide consists of a fibre bundle composed of optical fibres. It is connected to the compact probe (**C1.14**) composed of a diaphragm and objective lenses. The pattern coming from the compact probe is projected on the object surface.

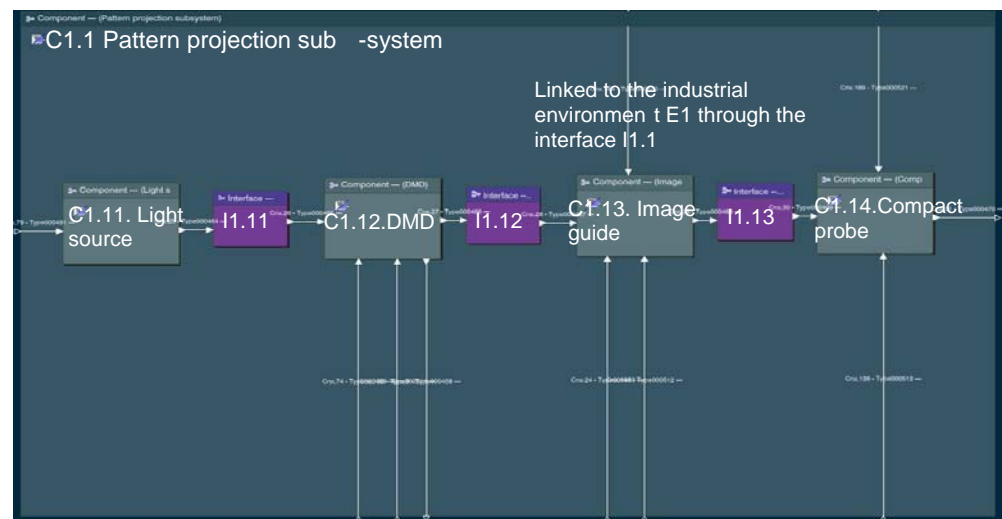


Figure 4.10 Representation of pattern projection sub-system

The solution principles in Table 4.1 shows that the sub-systems and components have been attributed to discipline-specific design teams. In the Pattern projection sub-system, the designers of optical team should choose the right components by referring the data sheet (e.g., light source, DMD and image guide) or design them by using the hierarchy of the design parameters proposed in the design methodology (e.g., compact probe).

This light transmission between the DMD and the compact probe is realized by an image guide. This image guide has to be taken into consideration very carefully. On the one hand, an image guide with a high resolution is needed to meet the requirement of the inspection for the industrial equipments. On

PROPOSITIONS

the other hand, the image guide has to be flexible enough to accommodate the industrial environment (**E1**). Figure 4.11 shows the interface (**I1.1**) between the industrial environment (**E1**) and the image guide (**C1.13**).

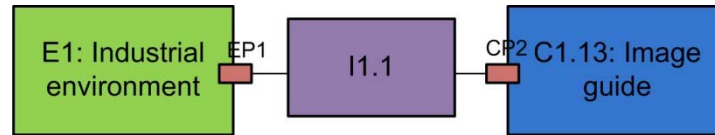


Figure 4.11 Interface (I1.1) between the industrial environment (E1) and the image guide (C1.13)

Figure 4.12 shows the data sheet of the image guides. In order to obtain a better reconstruction result, the designers of the optical team choose the image guide with the highest resolution (FIGH-1001500N), whose minimum bending radius is 200mm.

Image fiber
 ● Silica based image fiber / FIGH Series N Type

	FIGH-10-500N	FIGH-15-600N	FIGH-30-850N	FIGH-50-1100N	FIGH-70-1300N	FIGH-100-1500N
Number of Pixels (Nominal)	10,000	15,000	30,000	50,000	70,000	100,000
Image Circle Diameter (μm)	460	550	790	1,025	1,200	1,400
Fiber Diameter (μm)	500	600	850	1,100	1,300	1,500
Coating Diameter (μm)	600	700	950	1,200	1,450	1,700
Minimum Bending Radius (mm)	50	60	90	110	150	200
Coating Material (*)	Silicone Resin (Black)					
Lattice Defect (%)	<0.1					
Non-circularity (%)	<5					

Figure 4.12 Image guide data sheet

By analysing the scale of the whole measurement system and the industrial environment, the designers of the mechanical team propose the maximum scale of system. The value of the maximum scale (350mm) is represented as maxValue in the instance Industrial scale. The interface I1.1 links the image guide and the industrial environment which are defined by the designers from two different project teams - the optical team and the mechanical team. The multi-disciplinary interface provides the designers of different disciplines a standard representation which can help them to overcome the lack of commonality in interface definition and to achieve the multidisciplinary integration.

The data model of the interface I1.1 with the ports **EP1** and **CP2** can be created according to the above analysis. The UML object diagram in Figure

4.13 shows the instance of the interface I1.1 and the two ports **EP1** and **CP2** linked by I1.1. The parameters of each port can be stored in the demonstrator based on 3DEXPERIENCE Platform. As introduced before, the interface compatibility rules can be realised by the Knowledgeware functionalities. The test result of interface compatibility shows that the interface between the industrial environment and the image guide is incompatible, because the minimum bending diameter of the image guide is beyond the size limit proposed by the industrial environment.

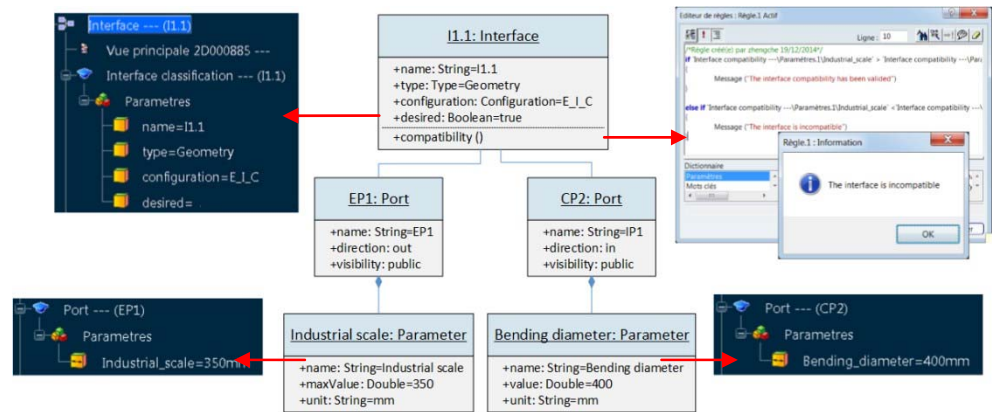


Figure 4.13 Instance of the incompatible interface I1.1

The incompatibility of the interface between the industrial environment and the image guide has been detected; therefore this incompatible interface has to be taken into consideration by the designers. The solutions proposed in Chapter 3 can help the designers to solve this incompatibility problem. As described in the component change solution, one element linked by the incompatible interface can be changed to solve this incompatibility problem (Figure 4.14). Thus, the designers of the optical team replace the image guide with FIGH70-1300N whose minimum bending radius is 150mm, and the interface compatibility can then be validated (Figure 4.15).

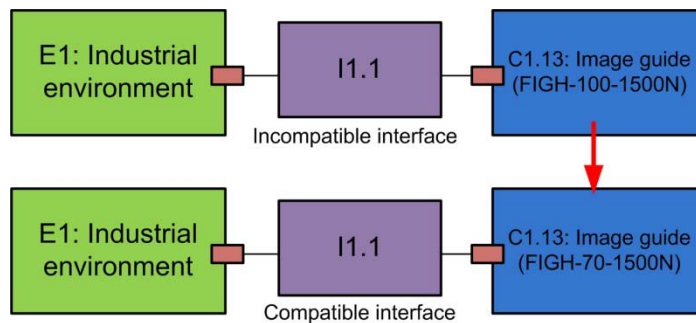


Figure 4.14 Solution 1 dealing with the incompatible interface I1.1

This compatible interface between the industrial environment and the image guide means that the size of the image guide accommodates the industrial environment.

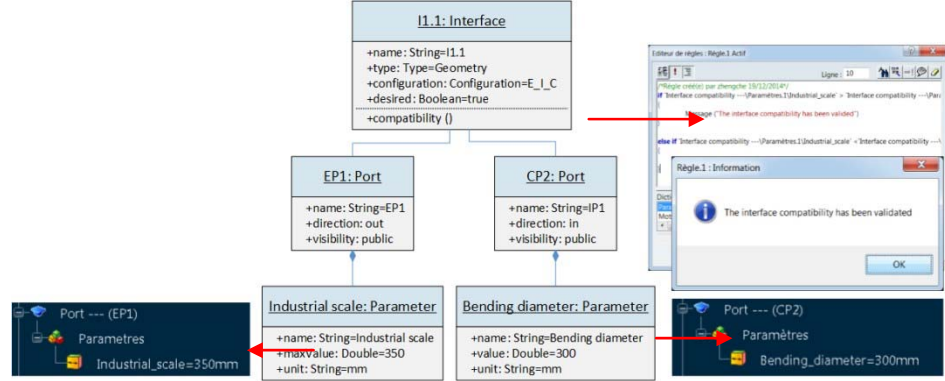


Figure 4.15 Instance of the compatible interface I1.1

Once the design of the image guide has been finished (FIGH-701300N), the attention should be paid to the interface (I1.12) between the DMD (C1.12) and the image guide (C1.13) (Figure 4.16). Two different kinds of transfers exist in the interface I1.21. The first one is related to the geometric shapes of the two components while the second one is used to transfer the energy. The interface where different types of transfers exit simultaneously requires a further decomposition.

As a result, the interface I1.12 is decomposed into I1.121 and I1.122.

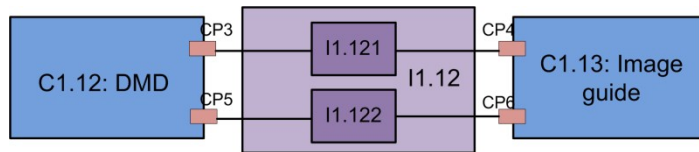


Figure 4.16 Interface (I1.12) between the DMD (C1.12) and the image guide (C1.13)

Attention should be paid to the geometric interface I1.121. The image guide data sheet (Figure 4.11) shows that the image circle diameter of the FIGH-70-1300N is 1200 μm (1.2mm). However, the maximum diameter of the image circle projected by the DMD should be the width of the DMD (8.3mm), which can be obtained by calculating from the data sheet shown in Figure 4.17.

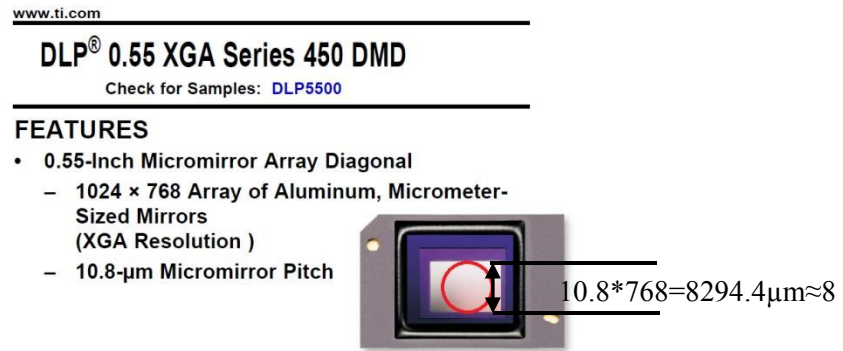


Figure 4.17 DMD data sheet and maximum image circle diameter

The UML object diagram in Figure 4.18 shows the instance of the interface I1.12 with the two ports CP3 and CP4. By applying the interface compatibility check method, the designers will find that the interface between the DMD and the image guide is incompatible because the image circle diameter of the DMD is different from that of the image guide.

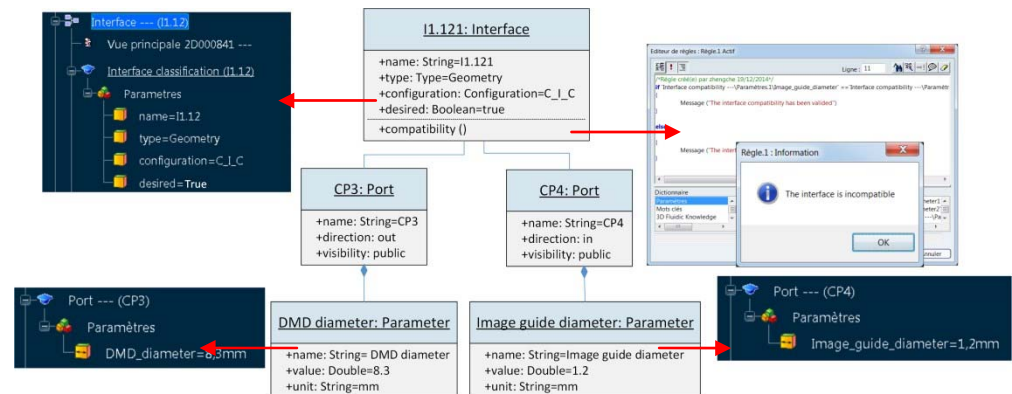


Figure 4.18 Instance of the incompatible interface I1.12

This incompatible interface between the DMD and the image guide indicates that these two components do not integrate correctly and cannot directly be connected with each other. The second solution interface decomposition solution, requires the incompatible interface to be decomposed into an Interface-Component-Interface structure. A lenses system which can change the image circle diameter will be defined by the designers of the optical team (Figure 4.19). The compatibility of the new interface (I1.1211 and I1.1212) will be checked.

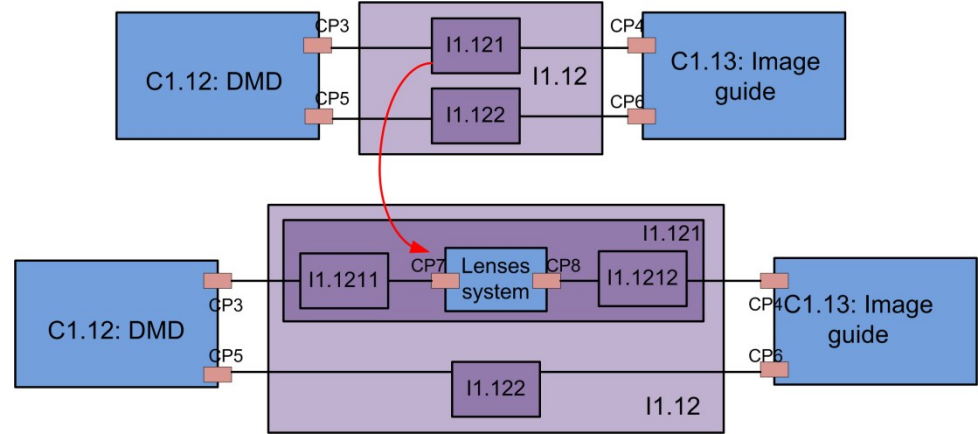


Figure 4.19 Solution 2 dealing with the incompatible interface I1.121

The lenses system can modify the image circle diameter from 8.3mm to 1.2mm. This modification process realised by the lenses system is modelled and simulated by the optical software Zemax EE (Dupont, Lamarque, & Prelle, 2011). The parameter of this lenses system's ports can be created according to the simulation result and stored in the interface data models of I1.1211 and I1.1212. Figure 4.20 takes the interface I1.1211 as an example to show that the incompatibility problem of I1.121 has been solved by the interface decomposition solution.

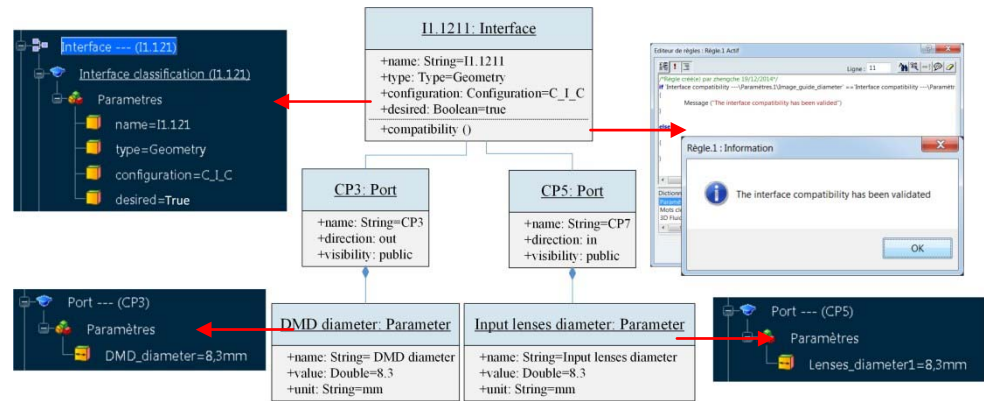


Figure 4.20 Instance of the compatible interface I1.1211

Now attention should be paid to the second sub-interface I1.122. This interface is used to transfer the light from the DMD to the image guide. By referring the data sheet of the DMD and the image guide chosen before, the designers of the optical design team find that the image guide has lower resolution than that of the DMD. Indeed the resolution of the image guide equivalent of 70k pixels, while the resolution of the DMD is almost 800k pixels. Such difference of resolution between the DMD and the image guide

will cause the loss of resolution during the image transfer process and create serious impact on the quality of the image reconstruction.

The designers of optical design team have already used the lenses system to change the image diameter to solve the incompatibility problem of the geometric interface II.1.121, but the optical measures such as lenses system cannot be used to change the resolution of the DMD or the image guide. As a result, the solution proposed by the project teams of other disciplines should be taken into consideration to solve this incompatibility problem.

The multi-disciplinary interface model can be used to support such multi-disciplinary collaboration and then achieve a more integrated design. The reconstruction algorithms should be carefully developed by the designers of software team to solve such incompatibility problem. The data sheet of the DMD shows that the DMD has a width of 768 pixels. However, the image guide has 300 pixels in the diameter direction by calculating from the data sheet of the chosen image guide. If the image guide is used to transfer the image projected from the DMD, every pixel of the image guide can transfer $768/300=2.56$ pixels of the DMD. This result indicates that if the fringe patterns projected by the DMD have a width of less than 2.56 pixels (e.g., 2 pixels). Such width of fringe patterns cannot be identified by the image guide, so serious impacts will be created on the quality of the image reconstruction. Therefore, a reconstruction algorithm which can analyse the fringe patterns whose width is more than 3 pixels should be proposed by the designers of software team. The phase shifting algorithm is finally adopted by the designers of software team because the minimum width of the fringe patterns is 8 pixels. Figure 4.21 show that the incompatibility problem of the interface II.1.122 has been well solved by using the Interface decomposition solution. The phase shifting algorithm proposed by the software team is added between the DMD and the image guide to eliminate the loss of resolution during the image transfer process. This example demonstrates that the multidisciplinary interface model can efficiently support the multidisciplinary collaboration during the design process to achieve a much more integrated design.

PROPOSITIONS

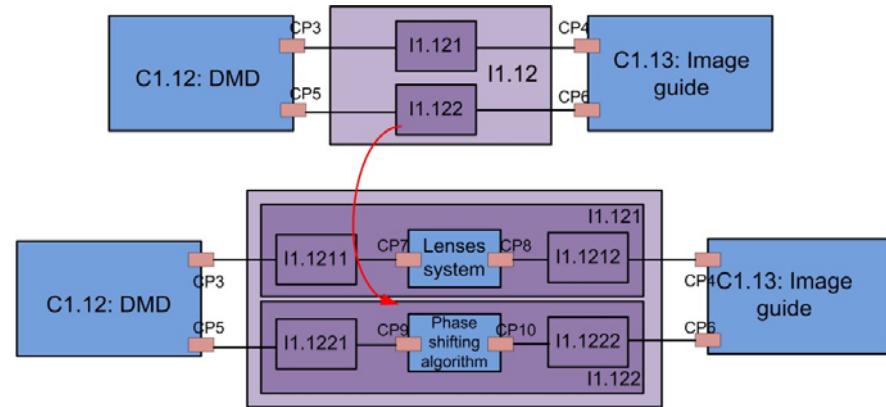


Figure 4.21 Solution 2 dealing with the incompatible interface I1.122

Figure 4.22 shows a part of the 3D measurement system in which the lenses system is design to change the image circle diameter of the DMD.

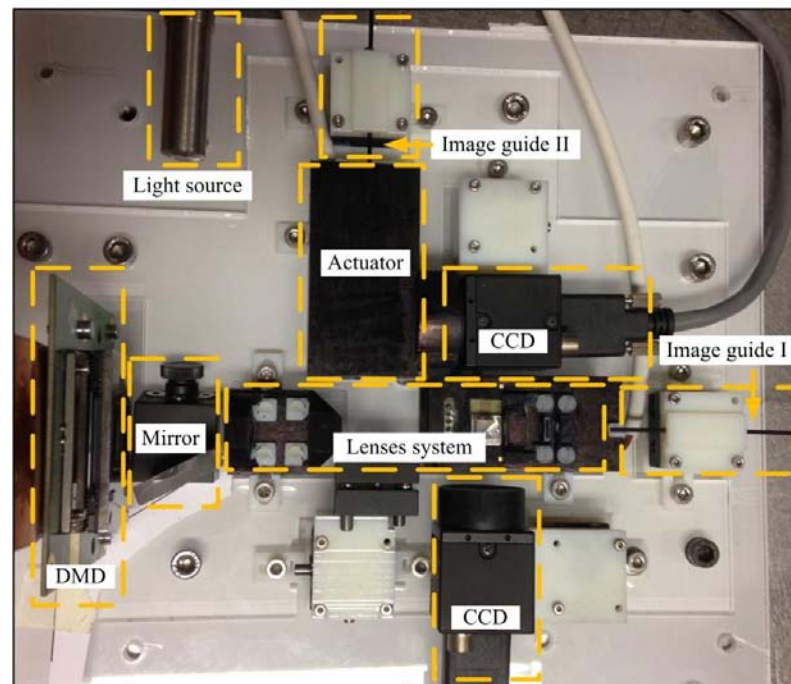


Figure 4.22 Part of 3D measurement system

4.2.3 Synthesis of case study

This sub-section has presented the design process of the measurement system carried out in line with the proposed design methodology. The three phases of the system design sub-process in the macro level process, including the requirements specification phase, the functional modelling phase and the architectural definition phase have been presented by the hierarchical structures and demonstrated by the Requirement engineering view, Functional view and Logical view of RFLP respectively implemented in the

3DEXPERIENCE Platform. The main sub-systems and the interfaces of the 3D measurement system have been specified and demonstrated thanks to Logical Editor workbench at the end of the architectural definition phase. In order to demonstrate the multi-disciplinary interface model more clearly, the pattern projection sub-system has been chosen. During the disciplinespecific design process, the image guide of the pattern projection subsystem should be chosen by the optical design team, while by analysing the industrial environment, the designers of the mechanical team should propose the maximum scale of the system. The data model of the interface between the image guide and the industrial environment can accordingly be defined in the demonstrator. The designers can test the interface compatibility by using the compatibility rules created by the CATIA Knowledgeware. The second example of the interface chosen in this case study is the interface between the DMD and the image guide. This interface proves to be incompatible by applying the interface compatibility rules on it. In order to solve this problem, the interface decomposition solution has been adopted. It means that the designers have to go back to the architectural definition phase and this incompatible interface should be further decomposed into an interface-component-interface structure. Therefore a lenses system is added between the DMD and the image guide. The compatibilities of the two new interfaces have been tested when the design of the lenses system has been finished.

□ 4.3 Summary

In this chapter, the demonstrator based on 3DEXPERIENCE Platform is first presented to demonstrate the propositions on the multidisciplinary interface model and the design methodology based on it. As for the multi-disciplinary interface model, the VPM Functional Logical Editor workbench in 3DEXPERIENCE Platform is used to represent the interface classification and the interface data model. The interface compatibility rules are implemented by using the CATIA Knowledgeware. As for the design methodology, the Requirement engineering view, Functional view and Logical view of the RFLP approach in ENOVIA/3DEXPERIENCE Platform

are used to represent the three phases of system design sub-process of the design methodology.

The design of the 3D measurement system is then used as the case study and its design process is carried out in line with the proposed design methodology by using the demonstrator above presented. The interface between the industrial environment and the image guide is chosen as an example to demonstrate how the proposition can help the designers to achieve the multi-disciplinary integration. The image guide is chosen by the designers from the optical team while the maximum scale of system constrained by the industrial environment is decided by the designers of mechanical team. The multi-disciplinary interface model not only provides a standard representation for the interface to the designers of different disciplines, but also tests the compatibility of this interface to ensure the two components designed by different teams integrate with each other correctly. Moreover, another interface between the image guide and the DMD is chosen to demonstrate the propositions. Two different transfers exist in the interface between the image guide and the DMD. According to the interface decomposition method, such interface should be further decomposed into two sub-interfaces.

The first sub-interface is related to the geometric shapes of the two components. The interface decomposition solution is used to solve the incompatibility problem of this sub-interface. This example has shown that the interface decomposition solution can help the designers to refine the architectural definition and to release the architectural improvement during the design process. As to the second sub-interface which transfers the light between the DMD and the image guide, this sub-interface is proved to be incompatible due to the difference of resolution. Multi-disciplinary collaboration should be considered to solve this incompatibility problem because the designers of optical team cannot find any optical measures to change the resolution of the DMD or the image guide. Thanks to the standard representation proposed by the multi-disciplinary interface model, the designers of software team propose the phase shifting algorithm to eliminate the loss of resolution during the image transfer process.

In summary, the case study in the chapter has demonstrated that the proposed multi-disciplinary interface model and the design methodology can well support the multi-disciplinary collaboration during the design process. Then

PROPOSITIONS

a much more integrated design of mechatronic systems has been achieved based on the propositions.

Conclusions and future work

This thesis research started with two types of scientific problems - Process-based problems and Design data-related problems, which may be encountered by the designers when achieving the multi-discipline based integrated design of mechatronic system. These two types of scientific problems reveal the research objectives and research questions. This chapter firstly summarises the findings and contributions in this research regarding each of the research questions. Additionally, the learning from this thesis research has opened the door to other possible research directions. Thus the second part of this chapter, the possible future work will be pointed out.

□ 5.1 Summary on the research questions

The interest of this thesis is on the multi-disciplinary integration during the design process of mechatronic systems. According to the definition of mechatronics of (Kyura & Oho, 1996) - “the synergistic integration of mechanical engineering with electronics and intelligent computer control in the design and manufacturing of industrial products and processes”, which is considered as one of the most accurate definitions for mechatronics, the design of mechatronic systems should focus not only on the synergistic integration of products, but also on the integration of design processes. Therefore the thesis focuses on the two kinds of integration which are relevant to the multi-discipline based integrated design of mechatronic systems - integration related to the product and integration related to the process. Concerning the integration related to the product, traditionally, mechatronic system is considered as the resulting integration of electrical/electronic components, mechanical parts and information processing. However, much innovative functionality through their networking and their access to the cyber world has been integrated into mechatronic systems. From the systems in which mechanical part and

CONCLUSIONS AND FUTURE WORK

electronic components are combined, through the modern smart products involving various disciplines, to Cyber Physical System (CPS), Internet of Things (IoT) and System of systems (SoS) into which cyber world has been integrated, the mechatronic system has become increasingly integrated. In summary, the integration related to product shows that design of mechatronic system has become increasingly integrated due to many additional factors from various disciplines that must be considered. This kind of integration requires a standard representation of mechatronic systems from different disciplines and a shared interpretation of product modelling languages and terminologies. The product model is considered as an effective support for the design of mechatronic can be systems because the database of it is generated by the designers from different disciplines during the product development process.

Concerning the integration related to the process, it requires a high level collaboration from both the designers and the various disciplines during the design process of mechatronic systems.

In order to achieve the multi-discipline based integrated design, the thesis focuses on the two kinds of integration - integration related to product and integration related to process. These two kinds of integrations raise the scientific problems. The former raises “Design data-related problems” while the latter brings up “Process-based problems”. The research objectives and research questions can be accordingly proposed. How the propositions can answer these research questions will be presented hereafter.

Two research questions related to the design data are raised:

- How to realise the macro level interface?

The macro level interface in the thesis describes the associations between subsystems defined by the designers of different disciplines, both to indicate their inter-dependence and to provide high-level guidance for how sub-systems should be integrated in the final product. In order to answer this research question, the thesis proposes the multi-disciplinary interface model which provides a standard and shared representation for the interfaces among the sub-systems defined by project teams from different disciplines. On the one hand, the interface classification is represented as attributes in the proposed multi-disciplinary interface model and it provides much more details of an interface to the designers and helps them to

avoid the confusion by the misuse of interfaces. On the other hand, during the design process, the compatibility of the interfaces should be checked by the method **compatibility()** so that the different sub-systems can be assembled correctly and the designers can ensure the multi-disciplinary integration among project teams. With the support of the multi-disciplinary interface model, the macro level interface can be therefore achieved.

- How to realise the micro level interface?

The micro level interface defined in the thesis as “the interface which allows the designers to exchange and share information or data from other disciplines during the process of mechatronic design”. It intends to help designers to collaborate or coordinate by sharing information through formal or informal interaction. This research question can be also answered by the proposed multi-disciplinary interface model. First, the attributes **direction** of the class **Port** holds a definition on which one is the master and which one is the slave of the two elements linked by the interface, and the attribute **visibility** describes how the port and all the related information can be accessed by the designers. Second, a demonstrator has been developed based on 3DEXPERIENCE Platform. This demonstrator allows the designers to develop the prototype of a mechatronic system through the data sharing and exchange among the designers. To sum up, the multidisciplinary interface model can greatly help the designers to collaborate or coordinate during the design process of mechatronic systems, so it is considered as an effective support for the micro level interface.

Two research questions related to the process are presented as follows: □

How to achieve the macro level collaboration?

The macro level collaboration emphasises the homogeneous collaboration of disciplines during the concurrent engineering process. Figure 1.5 in Chapter 1 shows that the macro level interface is considered as an effective support for macro level collaboration. However, the multi-disciplinary interface model which realises the macro level interface cannot be used directly to manage the design process. Therefore the design methodology based on the multi-disciplinary interface model is proposed. The design methodology is

developed based on an extended V-model during which the concurrent engineering can be carried out. In this design process, sub-processes (system design, discipline-specific design and system integration) are presented in the extended V-model. During the system design sub-process, the designers can gradually improve the architectural definition of a mechatronic system thanks to the multi-disciplinary interface model which defines an interface from different decomposition levels or different disciplines with a same terminology. During the discipline-specific design sub-process, the compatibility of the interfaces between the components should be checked by the method **compatibility()** in order to ensure the right assembly of the sub-systems designed by the project teams of different disciplines. In summary, with the support of the multi-disciplinary interface model, the integration of the components designed by the project teams of different disciplines is ensured and the collaboration among the project teams is achieved. As a result, the macro level collaboration can be achieved by the design methodology.

- How to achieve the micro level collaboration?
The collaboration among the designers (or individuals) is called in the thesis micro level collaboration. The propositions aim to help the designers to achieve the micro level collaboration. First, as presented in Chapter 1, the micro level interface is considered as an effective support for micro level collaboration. The demonstrator in which the multi-disciplinary interface model is implemented enables the designers to share and exchange data. Second, the design methodology adopts the hierarchical structure in every design phase of the system design sub-process which proposes very specific views of the system. It not only describes the details in every design phase, but also helps the designers to ensure the consistency between the phases. Therefore, the micro level collaboration has been achieved by the design methodology.

The above discussion shows that the research questions related to the product data and design process can be answered by the propositions on multi-disciplinary interface model and the design methodology. In other words, the propositions can be considered as an effective support to help the designers

to achieve the multi-disciplinary integration for the design of mechatronic systems. Following sub-section will summarise the research contribution of the thesis to the design of mechatronic systems.

□ 5.2 Research contributions to design of mechatronic systems

The thesis contributes to both the academic research on design of mechatronic systems and current practices of mechatronic engineering. The first main proposition of the thesis can be concluded as “a multidisciplinary interface model as an extension of existing product models has been proposed”, whose contributions can be described as follows:

- The thesis defined a new interface classification for the interfaces existing in mechatronic systems. This interface classification is considered as the foundation of the multidisciplinary interface model. It not only gives much more details of an interface, but it also provides a standard representation of interfaces existing in mechatronic systems. This helps the designers to avoid the confusion by the misuse of interfaces in an early design phase of mechatronic systems.
- The thesis proposed a multi-disciplinary interface model which can describe the interfaces in mechatronic systems with a standard representation and a uniform interpretation of product modelling languages and terminologies. In other words, the proposed multi-disciplinary interface model provides a common terminology for the project teams dedicated to the design of mechatronic systems. It not only takes into account the information of proposed interface classification, but also represents the relationship between the interface and other entities of the product model. By analysing and comparing the entities of the multi-disciplinary interface model and those of existing product models, the multi-disciplinary interface model proves to be usable as an extension of existing product models and can be merged with them.
- The thesis proposed two interface compatibility rules and two solutions to solve the incompatibility problems. They both play an

important role to guarantee the right integration of the different elements in one mechatronic system and ensure the multi-disciplinary integration among disciplines.

The second main proposition of the thesis research is related to the design methodology based on the multi-disciplinary interface model. The contributions of this design methodology can be concluded hereafter:

- The thesis proposed the design methodology for the mechatronic systems based on an extended V-model. It defines an integrated design process starting with identification of all requirements on the total system and ending with a uservalidated system. In this design process, sub-processes (system design sub-process, discipline-specific design sub-process and system integration sub-process) and their design phases are presented.
- During the system design sub-process, the specific design phases are detailed, where individual designers can structure design sub-tasks and proceed and react in unforeseen situations. The multi-disciplinary interface model based decomposition method can help the designers to achieve a complete and appropriate architecture during architectural definition phase. By making use of the multi-disciplinary interface model, the component decomposition helps the system designers to decide the appropriate hierarchies for the system architecture; while the interfaces decomposition intends to support the evolution of the system architecture during the design process.
- During the discipline-specific design sub-process, the method **compatibility()** proposed based on the multi-disciplinary interface model will test the compatibility among the subsystems (or components) and help the designers to decide whether the design process can enter the next phase to test whether the system fulfils the proposed function and meets all the requirements previously specified or it should go back to the previous design phase. This iterative process can guarantee the sub-systems (or components) designed by the project teams of different disciplines to integrate correctly and to ensure the multi-disciplinary integration among different design teams.

□ 5.3 Future work

The research contributions of the thesis have been summarised. As for the future work, the propositions can be further expanded in several directions:

- As above depicted, during the architectural definition phase of the design process, systems with their interfaces should be decomposed into sub-systems and sub-interfaces. The decomposition of the mechatronic system will be applied recursively. However, there is always a research question, that is, what is the appropriate decomposition granularity for architectural definition of mechatronic systems? On the one hand, the decomposition should provide sufficient granularity so that the sub-systems can be allocated to discipline-specific teams or considered as standard components. On the other hand, a fine decomposition granularity will lead to time-consuming work during the sub-systems' integration process and could lead to less integrated systems due to separated and domain specific solutions. In order to solve the contradiction between the decomposition and integration level during the design process of mechatronic systems, more attention should be paid to the decomposition granularity in the future.
- 3DEXPERIENCE Platform allows the designers to define the hierarchical structures for the requirement specification, functional modelling and the architectural definition and a demonstrator of the multi-disciplinary interface model based on the Knowledgeware of 3DEXPERIENCE Platform has been developed. However, an intelligent IT platform means more than that. It should not only enable the collaboration and communication among the project teams of different disciplines, which have been partially implemented by the proposed demonstrator, but it should also provide an automation of these design activities or tasks according to the decisions made by the designers. Therefore, such intelligent IT platform which can fully support the entire design methodology should be further developed in the future.
- Clarification of all requirements on the whole system, as the early design phase of the proposed design methodology, depends largely

on the experience of the designers. Therefore different designers may propose diverse architectural models which are derived from the requirements specifications. In the case study presented in the thesis, the design of lenses system is used as an example to illustrate the method to solve the incompatible interface and the hierarchy of the design parameter in the discipline-specific design sub-process. However, as for an experienced mechatronic system designer, a lenses system may be taken into consideration when fixing the architectural model at the early stage of the system design subprocess and the iteration loop to solve the incompatibility problem can be reduced accordingly. But in most design cases, a complete architectural definition cannot be obtained without iterations and it needs to be improved along with the design process. The study case indicates that the proposed design methodology can well support such architectural evolution during the design process of mechatronic systems. And meanwhile, it is shown that the designers' experiences play significant roles in the design of mechatronic system. Much more attention should be paid to the designers' experiences (or knowledge) because the iteration loops can be greatly avoided so that the development costs and the time-to-market can be decreased.

- Like product models, ontology has been recently applied to product modelling from the perspective of semantic representation and it can be also considered as a potential solution to the design data-related problems. The proposition on multi-disciplinary interface model has been also inspired from the current works on the ontology based product modelling approaches. Compared with product models, ontology defines a semantic representation (Abdul-ghafour et al., 2012). It can give clearer meaning to product models by interpreting them as physical objects, rather than linguistic constructions such as modelling languages used by product models. The multi-disciplinary interface model may be transferred and integrated with the ontology based product modelling approaches in the future. They provide a semantic representation to the designers of different disciplines enabling data exchange over the internet.

CONCLUSIONS AND FUTURE WORK

- As mechatronic systems become increasingly integrated, the multi-disciplinary collaboration not only takes place between different design teams of one company, but between entire supply chains as well. Enhanced supply chain management can create efficiencies and cost savings across a wide range of development process (L. Horvath, 2011). Multi-disciplinary collaboration makes more and more sensitive information (e.g. engineering, financial or customer data) and product models coming from the global supply chain available to the designers, and meanwhile, accessing to such sensitive information and product models can help the designers to make fast and accurate decisions. As a result, nowadays the security of product model and team information plays an increasing role. The proposed multi-disciplinary interface model has begun to consider the information security. The attribute visibility of the class Port, with its authorized values *public*, *protected* and *private*, is used to describe how the parameter and document linked with one port can be accessed. However, besides the parameter and document linked with one port, the security of the information provided by the supply chain should be also taken into consideration. Much more attention should be paid to the security of product model and information coming from global supply chain during the design process of mechatronic systems.
- As above introduced, nowadays, based on the information and communication technology, various kinds of complex systems, such as Cyber Physical System (CPS), Internet of Things (IoT), System of systems (SoS) have attracted increasing attentions from both the academia and industry. As a result, attention should be paid to some new research directions by considering the above internet-enabled systems. The three systems are quite similar with each other and sometimes the frontier among them has not been clearly identified. For example, CPS is automated system that enables connection of the operations of the physical reality with computing and communication infrastructures (Baheti & Gill, 2011). The CPS connected to the internet is often referred to as the “Internet of Things” (Jazdi, 2014) and the IoT can be engaged in complex relationships including the composition and collaboration with other

independent and heterogeneous systems in order to provide new functionalities, thus leading to the so-called systems-of-systems (SoS) (Maia et al., 2014). Different from the design of mechatronic systems, when the project teams face the design of such Internet-enabled systems, they should focus on the integrating multiple, asynchronous (but interdependent) complex systems, rather than a single complex system. However, like the design of mechatronic systems, the key challenges associated with the engineering process of such Internet-enabled systems concerns needs to manage the interfaces among components systems that are generally individually acquired and integrated (Wells & Sage, 2009). Due to the increasing complexity of such systems, the designers should not only deal with the interfaces defining how a sub-systems (or components) interacts with each other or with the environment (as what have been discussed in the thesis), but also manage the interfaces with other complex systems and the complex human-information interfaces (Jolly & Muirhead, 2009). In order to overcome the challenges of interfaces in the internet-enabled systems, (Watson, 2003) suggests the possibility by making use of model-based representations. As a result, the multi-disciplinary interface model can be further developed by considering the specialties of the internet-enabled systems to help the designers to clarify, understand, define, realise and manage the interfaces between different disciplines, different subsystems, and/or between people and organisations.

References

- Abdul-ghafour, S., Ghodous, P., & Shariat, B. (2012). Integration of product models by ontology development. In *IEEE 13th International Conference on Information Reuse and Integration (IRI)* (pp. 548–555). Las Vegas, USA. 8-10 August.
- Abid, H., Pernelle, P., Noterman, D., Campagne, J.-P., & Amar, C. Ben. (2014). SysML approach for the integration of mechatronics system within PLM systems. *International Journal of Computer Integrated Manufacturing*, 28(9), 972–987.
- Abrahamsson, P., Conboy, K., & Wang, X. (2009). “Lots done, more to do”: the current state of agile systems development research. *European Journal of Information Systems*, 18(4), 281–284.
- Abramovici, M., & Bellalouna, F. (2007). Integration and complexity management within the mechatronics product development. In *Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses* (pp. 113–118).
- Alvarez Cabrera, A. A., Foeken, M. J., Tekin, O. A., Woestenenk, K., Erden, M. S., De Schutter, B., ... Tomiyama, T. (2010). Towards automation of control software: a review of challenges in mechatronic design. *Mechatronics*, 20(8), 876–886.
- Ameri, F., & Dutta, D. (2005). Product Lifecycle Management: Closing the Knowledge Loops. *Computer-Aided Design and Applications*, 2(5), 577–590.
- Arnold, S., & Lawson, H. W. (2004). Viewing systems from a business management perspective: The ISO/IEC 15288 standard. *Systems Engineering*, 7(3), 229–242.
- Bagnall, B. (2002). *Core Lego Mindstorms programming*. New Jersey, USA: Prentice Hall Upper Saddle River.
- Baheti, R., & Gill, H. (2011). Cyber-physical Systems. *The impact of control technology*, 12(1), 161–166.
- Barbieri, G., Fantuzzi, C., & Borsari, R. (2014). A model-based design methodology for the development of mechatronic systems. *Mechatronics*, 24(7), 833–843.
- Bathelt, J., Jonsson, A., Bacs, C., Dierssen, S., & Meier, M. (2005). Applying the new VDI Design Guideline 2206 on mechatronic systems controlled by a PLC. In *International Conference on Engineering Design (ICED 05)*. Melbourne, Australia. 15-18 August.
- Beier, G., Figge, A., Müller, R., Rothenburg, U., & Stark, R. (2013). Supporting product development through cross-discipline dependency-modeling – novel approaches for traceability-usage. *Lecture Notes on Information Theory*, 1(1), 21–28.

REFERENCES

- Bettig, B., & Gershenson, J. K. (2010). The representation of module interfaces. *International Journal of Product Development*, 10(4), 291–317.
- Blanchard, B. S. (2008). *Systems engineering management*. New Jersey, USA: Wiley.
- Bloebaum, C. ., Hajela, P., & Sobieski-Sobieszczanski, J. (1993). Decomposition Methods for Multidisciplinary Synthesis. In *Multidisciplinary engineering systems: design and optimisation techniques and their application*. San Diego, USA: Academic Press.
- Blyler, J. (2004). Interface management. *Instrumentation & Measurement Magazine*, 7(1), 32–37.
- Boardman, J. (2006). System of Systems - the meaning of of. In *IEEE/SMC International Conference on System of Systems Engineering* (pp. 118–123). Los Angeles, USA. 24-26 April.
- Bock, C., Zha, X., Suh, H., & Lee, J.-H. (2010). Ontological product modeling for collaborative design. *Advanced Engineering Informatics*, 24(4), 510–524.
- Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5), 61–72.
- Boehm, B. W. (2002). Get ready for agile methods, with care. *Computer*, 35(1), 64– 69.
- Boland, D., & Fitzgerald, B. (2004). Transitioning from a co-located to a globallydistributed software development team: A case study at Analog Devices, Inc. In *3rd Workshop on Global Software Development*. Edinburgh, Scotland. 24 May.
- Booch, G., Rumbaugh, J., & Jacobson, I. (1998). *The unified modeling language user guide*. Massachusetts, USA: Addison-Wesley.
- Bricogne, M. (2015). *Méthode agile pour la conception collaborative multidisciplinaire de systèmes intégrés: application à la mécatronique*. PhD thesis, Université de Technologie de Compiègne.
- Bricogne, M., Petit, L., Meyine, A. M., Troussier, N., Eynard, B., & Schweitzer, D. (2010). Mechatronics issues for specification of PLM functionalities and features. In *MECATRONICS2010*. Yokohama, Japan. 22-24 November.
- Bruun, H. P. L., Mortensen, N. H., Harlou, U., Wörösch, M., & Proschowsky, M. (2015). PLM system support for modular product development. *Computers in Industry*, 67, 97–111.
- Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2004). How extreme does extreme programming have to be? adapting xp practices to large-scale projects. In *Hawaii International Conference on System Sciences*. Hawaii, USA. 5-8 January.

REFERENCES

- Carlock, P. G., & Fenton, R. E. (2001). System of Systems (SoS) enterprise systems for information-intensive organizations. *Systems Engineering*, 4(4), 242–261.
- Carrier, J. E., Ohline, R. M., & Kenny, T. W. (2011). *Introduction to mechatronic design* (pp. 21–37). Boston, USA: Prentice Hall.
- Casner, D., Renaud, J., Houssin, R., & Knittel, D. (2012). A novel design approach for mechatronic systems based on multidisciplinary design optimization. In *International Conference on Automation and Mechatronics*. Oslo, Norway. 1314, August.
- Cawley, O., Wang, X., & Richardson, I. (2010). Lean/Agile software development methodologies in regulated environments – state of the art. In *Lean enterprise software and systems* (pp. 31–36). Berlin, Germany: Springer.
- Chen, K., Bankston, J., Panchal, J. H., & Schaefer, D. (2009). A framework for the integrated design of mechatronic systems. In *Collaborative Design and Planning for Digital Manufacturing* (pp. 37–70). London, UK: Springer.
- Chungoora, N., Young, R. I., Gunendran, G., Palmer, C., Usman, Z., Anjum, N. a., ... Case, K. (2013). A model-driven ontology approach for manufacturing system interoperability and knowledge sharing. *Computers in Industry*, 64(4), 392–401.
- Cooke, A., Maarten Bonnema, G., & Poelman, W. (2012). Agile development for a multidisciplinary bicycle stability test bench. In *13th Mechatronics forum international conference MECHATRONICS 2012*. Linz, Austria. 17-19 September.
- Counsell, J., Ian, P., David, D., & Duffy, M. (1999). Schemebuilder: computer aided knowledge based design of mechatronic systems. *Assembly Automation*, 19(2), 129–138.
- Craig, K. C., & Marchi, J. A. de. (1996). Mechatronic system design at Rensselaer. *Workshop on Mechatronics Education*, 4(1), 67–78.
- Department of Transportation. (2007). *Systems engineering for intelligent transportation systems* (p. 11). Washington, D.C., USA.
- Dimensionsguide. (2011). Dimensions Guide. *WWW document*. Retrieved from <http://www.dimensionsinfo.com/dimensions-of-a-standard-lego-brick>
- Doran, T. (2004). IEEE 1220: For Practical Systems Engineering. *Computer*, 39(5), 92–94.
- Dorfman, M. (1990). System and software requirements engineering. In *IEEE Computer Society Press Tutorial* (pp. 7–22). IEEE CS Press.
- Dupont, E., Hou, Y., Lamarque, F., & Redarce, T. (2013). Binary pattern codification strategies in an active stereoscopic system based on flexible image guides. In *SPIE*

REFERENCES

- MOEMS-MEMS. International Society for Optics and Photonics*. San Francisco, USA. 2 February.
- Dupont, E., Lamarque, F., & Prelle, C. (2011). 3D triangulation system based on outof-axis aperture configuration for micro-scaled objects shape measurement. In *Proceedings of SPIE, the International Society for Optical Engineering*. San Francisco, USA. 22 January.
- Elezi, F., Graebisch, M., Hellenbrand, D., & Lindemann, U. (2011). Application of multi-domain matrix waste reduction methodology in mechatronic product development. In A. Chakrabarti (Ed.), *Research into Design- Supporting Sustainable Product Development*.
- Eynard, B., Gallet, T., Nowak, P., & Roucoules, L. (2004). UML based specifications of PDM product structure and workflow. *Computers in Industry*, 55(3), 301–316.
- Fenves, S., Foufou, S., Bock, C., & Sriram, R. D. (2006). CPM : a core model for product data. *Journal of Computing and Information Science in Engineering*, 8(1), 1–14.
- Ferreira, H. C., Lampe, L., Newbury, J., & Swart, T. G. (2010). *Power line communications: theory and applications for narrowband and broadband communications over power lines*. West Sussex, UK: John Wiley & Sons, Inc.
- Fitzgerald, B., Stol, K.-J., O’Sullivan, R., & O’Brien, D. (2013). Scaling agile methods to regulated environments: an industry case study. In *35th International Conference on Software Engineering (ICSE)* (pp. 863 – 872). San Francisco, USA. 18-26 May.
- Forsberg, K., & Mooz, H. (1998). System engineering for faster , cheaper , better. In *INCOSE1998*. Vancouver, Canada. UK. 6-11 June.
- Fosse, E., & Delp, C. L. (2013). Systems engineering interfaces : a model based approach. In *2013 IEEE Aerospace Conference*. Big Sky, USA. 2-9 March.
- Fotso, A. B., Wasgint, R., & Achim, R. (2012). State of the art for mechatronic design concepts. In *8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications* (pp. 232–240). Suzhou, China. 8-10 July.
- Garcia, C. A. B., Kunz, J., Ekstrom, M., & Kiviniemi, A. (2004). Building a Project Ontology with Extreme Collaboration and Virtual Design & Construction. *Advanced Engineering Informatics*, 18(2), 71–83.
- Gausemeier, J., Dumitrescu, R., Kahl, S., & Nordsiek, D. (2011). Integrative development of product and production system for mechatronic products. *Robotics and Computer-Integrated Manufacturing*, 27(4), 772–778.

REFERENCES

- Gausemeier, J., Frank, U., Donoth, J., & Kahl, S. (2009). Specification technique for the description of self-optimizing mechatronic systems. *Research in Engineering Design*, 20(4), 201–223.
- Gausemeier, J., & Moehring, S. (2003). New Guideline VDI 2206 – A flexible procedure model for the design of mechatronic systems. In *14th International Conference on Engineering Design*. Stockholm, Sweden. 19-21 August.
- Gautam, N., & Singh, N. (2008). Lean product development: maximizing the customer perceived value through design change (redesign). *International Journal of Production Economics*, 114(1), 313–332.
- Gil, R. J., & Martin-Bautista, M. J. (2012). A novel integrated knowledge support system based on ontology learning: Model specification and a case study. *Knowledge-Based Systems*, 36(1), 340–352.
- Girard, P., & Doumeingts, G. (2004). Modelling the engineering design system to improve performance. *Computers and Industrial Engineering*, 46(1), 43–67.
- Gómez-Pérez, A., Fernandez-Lopez, M., & Corcho, O. (2004). Ontological engineering: with examples from the areas of knowledge management. In *Ecommerce and the Semantic Web*. London, England: Springer-Verlag.
- Grimheden, M. E. (2013). Can agile methods enhance mechatronics design education? *Mechatronics*, 23(8), 967–973.
- Gruber, T. R. (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2), 199–220.
- Gu, P., & Chan, K. (1995). Product modelling using STEP. *Computer-Aided Design*, 27(3), 163–179.
- Guarino, N. (1998). Formal ontology and information systems. In *Proceedings of the FOIS'98-Formal Ontology in Information Systems*. Trento, Italy. 6-8 June.
- Haller, S., Karnouskos, S., & Schroth, C. (2009). The Internet of things in an enterprise context. In *Future Internet – FIS 2008* (pp. 14–28). Berlin, Germany: Springer.
- Hazelrigg, G. A. (1996). *Systems engineering: an approach to information-based design*. New Jersey, USA: Prentice Hall Upper Saddle River.
- Hehenberger, P. (2014). Perspectives on hierarchical modeling in mechatronic design. *Advanced Engineering Informatics*, 28(3), 188–197.
- Hehenberger, P., Poltschak, F., Zeman, K., & Amrhein, W. (2010). Hierarchical design models in the mechatronic product development process of synchronous machines. *Mechatronics*, 20(8), 864–875.

REFERENCES

- Hellenbrand, D., & Lindemann, U. (2011). A framework for integrated process modelling and planning of mechatronic products. In *International Conference on Engineering Design (ICED11)*. Copenhagen, Denmark.
- Hermann, M., Pentek, T., & Otto, B. (2015). *Design Principles for Industrie 4.0 Scenarios: A Literature Review*.
- Highsmith, J. (2002). *Agile software development ecosystems*. Indianapolis, USA: Addison Wesley.
- Hoffman, D. (1990). On criteria for module interfaces. *IEEE Transactions on Software Engineering*, 16(5), 527–542.
- Hofmann, D., Kopp, M., & Bertsche, B. (2010). Development in mechatronics – enhancing reliability by means of a sustainable use of information. In *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Montréal, Canada.
- Horvath, I., & van Der Vegte, W. F. (2003). Nucleus-based product conceptualization-part 1: principles and formalization. In *International Conference on Engineering Design*. Stockholm, Sweden.
- Horvath, L. (2011). Collaboration: the key to value creation in supply chain management. *Supply Chain Management: An International Journal*, 6(5), 205–207.
- Howe, J. (2008). *Crowdsourcing: why the Power of the Crowd is Driving the Future of Business*. Crown Business.
- Hribernik, K. A., Ghrairi, Z., Hans, C., & Thoben, K. (2011). Co-creating the Internet of Things - first experiences in the participatory design of intelligent products with arduino. In *17th International Conference on Concurrent Enterprising (ICE 2011)* (pp. 1–9). Aachen, Germany.
- INCOSE. (2015). *INCOSE systems engineering handbook: a guide for system Life cycle processes and activities*. John Wiley & Sons.
- Isermann, R. (2007). Mechatronic design approach. In *The Mechatronics Handbook* (pp. 2–3). CRC Press.
- ISO 10303. (1994). *Industrial automation systems and integration – Product data representation and exchange*. International Organization for Standardization.
- ISO/IEC 15288. (2002). *Systems engineering-system life cycle processes*. International Standardization Organization/International Electrotechnical Commission.
- ISO/IEC 81346-1. (2012). *Industrial systems, installations and equipment and industrial products -- Structuring principles and reference designations -- Part 1: Basic rules*. International Organization for Standardization.

REFERENCES

- ISO10303-210. (2011). *Industrial automation systems and integration - Product data representation and exchange - Part 210: Application protocol: Electronic assembly, interconnect, and packaging design*. International Organization for Standardization.
- ISO10303-214. (2010). *Industrial automation systems and integration - Product data representation and exchange - Part 214: Application protocol: Core data for automotive mechanical design processes*. International Organization for Standardization.
- ISO10303-233. (2012). *Industrial automation systems and integration - Product data representation and exchange - Part 233: Application protocol: Systems engineering*. International Organization for Standardization.
- Jamshidi, M. (2009). *System of systems engineering: innovations for the 21st century*. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Jardim-Goncalves, R., Figay, N., & Steiger-Garção, A. (2006). Enabling interoperability of STEP application protocols at meta-data and knowledge level. *International Journal of Technology Management*, 36(4), 402–421.
- Jazdi, N. (2014). Cyber physical systems in the context of Industry 4.0. In *IEEE International Conference on Automation, Quality and Testing, Robotics*. ClujNapoca, Romania. 22-24 May.
- Jolly, S. D., & Muirhead, B. K. (2009). System of systems engineering in space exploration. In *System of systems engineering, innovations for the 21st century*. Boca Raton, USA: John Wiley & Sons, Inc.
- Jönsson, A. (2004). *Lean Prototyping of multi-body and mechatronic systems*. PhD thesis, Blekinge Institute of Technology.
- Kahkonen, T. (2004). Agile methods for large organizations - building communities of practice. In *Agile Development Conference*. Calgary, Canada. 22-26 June.
- Kapurch, S. J. (2010). *NASA Systems engineering handbook*. DIANE Publishing.
- Kemmerer, S. J. (1999). *STEP: The grand experience*. Gaithersburg, USA: US Department of Commerce, Technology Administration, National Institute of Standards and Technology.
- Kircher, M., Jain, P., Corsaro, A., & Levine, D. (2001). Distributed extreme programming. In *Proceeding of Extreme Programming and Flexible Processes in Software Engineering*. Sardinia, Italy. 20-23 May.
- Kitamura, Y., & Mizoguchi, R. (2003). An Ontological schema for sharing conceptual engineering knowledge. In *International Workshop on Semantic Web Foundations and Application Technologies*,. Nara, Japan. 12 March.

REFERENCES

- Kleiner, S., & Kramer, C. (2013). Model based design with systems engineering based on RFLP using V6. In *Proceedings of the 23rd CIRP Design Conference* (pp. 93–102). Bochum, Germany. 11-13 March.
- Koch, C., Spröwitz, A., & Ströhla, T. (2006). Project course “Design of Mechatronic Systems.” In *IEEE International Conference on Mechatronics (ICM 2006)*. Budapest, Hungary. 3-5 July.
- Komoto, H., & Tomiyama, T. (2012). A framework for computer-aided conceptual design and its application to system architecting of mechatronics products. *Computer-Aided Design*, 44(10), 931–946.
- Koubaa, A., & Andersson, B. (2009). A version of cyber-physical internet. In *International Workshop of Real-time Networks*. Dublin, Ireland. 30 June.
- Krause, F.-L., Kimura, F., Kjellberg, T., Lu, S. C.-Y., Wolf, van der, Alting, L., ... Week, M. (1993). Product modelling. *CIRP Annals - Manufacturing Technology*, 42(2).
- Kyura, N., & Oho, H. (1996). Mechatronics-an industrial perspective, 1(March), 10–15.
- Lardeur, E. (2006). *Improvement of the engineering efficiency in a systems engineering context, case of automotive products and related manufacturing systems engineering*. PhD thesis, Ecole Centrale Paris.
- Le Duigou, J., Bernard, A., & Perry, N. (2011). Framework for product lifecycle management integration in small and medium enterprises networks. *ComputerAided Design and Applications*, 8(4), 531–544.
- Lee, E. a. (2008). Cyber physical systems: design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. Orlando, USA. 5-7 May: Ieee.
- Lefèvre, J., Charles, S., Bosch-Mauchand, M., Eynard, B., & Padiolleau, E. (2014). Multidisciplinary Modelling and Simulation for Mechatronic Design. *Journal of Design Research*, 12(1-2), 127–144.
- Leonard, J. (1999). *Systems engineering fundamentals*. Michigan, USA: Defense Systems Management College. Press.
- Li, Q., Zhang, W. J., & Chen, L. (2001). Design for control-a concurrent engineering approach for mechatronic systems design. *IEEE/ASME Transactions on Mechatronics*, 6(2), 161–169.
- Liang, V.-C., & Paredis, C. J. J. (2004). A port ontology for conceptual design of systems. *Journal of Computing and Information Science in Engineering*, 4(3), 206–217.

REFERENCES

- Liu, D. T., & Xu, X. W. (2001). A review of web-based product data management systems. *Computers in Industry*, 44(3), 251–262.
- Long, F. (2009). Real or imaginary: the effectiveness of using personas in product design. In *Irish Ergonomics Society Annual Conference*. Dublin, Ireland.
- Maia, P., Cavalcante, E., Gomes, P., Batista, T., Delicato, F. C., & Pires, P. F. (2014). On the development of systems-of-systems based on the Internet of Things : a systematic mapping. In *Proceedings of the 2014 European Conference on Software Architecture Workshops*. Vienna, Austria. 25-29 August.
- Martin, J. N. (1998). Overview of the EIA 632 standard: processes for engineering a system. In *17th DASC. The AIAA/IEEE/SAE Digital Avionics Systems Conference*. Bellevue, USA. 31 October -7 November.
- Martin, J. N. (2000). Processes for engineering a system : an overview of the ANSI / EIA 632 standard and its heritage. *Systems Engineering*, 3(1), 1–26.
- Marwedel, P. (2011). *Embedded system design: embedded systems foundations of cyber-physical systems*. Dordrecht, Netherland: Springer.
- Matsokis, A., & Kiritsis, D. (2010). An ontology-based approach for product lifecycle management. *Computers in Industry*, 61(8), 787–797.
- McFarlane, D., Sarma, S., Chirn, L. J., Wong, C. Y., & Ashton, K. (2003). Auto ID systems and intelligent manufacturing control. *Engineering Applications of Artificial Intelligence*, 16(4), 365–376.
- Miller, E. (1998). PDM moves to the mainstream. *Mechanical Engineering*, 120(10), 74–79.
- Nattermann, R., & Anderl, R. (2013). The W-model – using systems engineering for adaptronics. *Procedia Computer Science*, 16, 937–946.
- Noël, F., & Roucoules, L. (2008). The PPO design model with respect to digital enterprise technologies among product life cycle. *International Journal of Computer Integrated Manufacturing*, 21(2), 139–145.
- Noël, F., Roucoules, L., & Teissandier, D. (2005). Specification of product modelling concepts dedicated to information sharing in a collaborative design contexte. In *Advances in Integrated Design and Manufacturing in Mechanical Engineering* (pp. 135–146). Dordrecht, Netherland.
- Nowak, P., Rose, B., Saint-Marc, L., Callot, M., Eynard, B., Gzara-Yesilbas, L., & Lombard, M. (2004). Towards a design process model enabling the integration of product, process and organisation. In *5th International Conference on Integrated Design and Manufacturing in Mechanical Engineering (IDMME'04)* (pp. 91–103). Bath, UK.

REFERENCES

- Oestersötebier, F., Just, V., Trächtler, A., Bauer, F., & Dziwok, S. (2012). Modelbased design of mechatronic systems by means of semantic web ontologies and reusable solution elements. In *ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis*. Nantes, France. 2–4 July.
- Orionrobots. (2011). LEGO Specification. *WWW document*. Retrieved from <http://orionrobots.co.uk/pages/lego-specifications.html>
- Pahl, G., Beitz, W., Feldhusen, J., & Grote, K.-H. (2007). *Engineering design: a systematic approach* (3rd editio.). London: Springer-Verlag.
- Parwekar, P. (2011). From internet of things towards cloud of things. In *International Conference on Computer & Communication Technology*. Allahabad, India. 1517 September.
- Patil, L., Member, S., Dutta, D., Sriram, R., & Member, S. (2005). Ontology-Based Exchange of Product Data Semantics. *IEEE Transactions on Automation Science and Engineering*, 2(3), 213–225.
- Paul, C. R. (2006). *Introduction to electromagnetic compatibility* (pp. 5–7). New Jersey, USA: John Wiley & Sons.
- Paviot, T., Cheutet, V., & Lamouri, S. (2011). A PLCS framework for PDM/ERP interoperability. *International Journal of Product Lifecycle Management*, 5(2), 295–313.
- Penas, O., Plateaux, R., Choley, J.-Y., & Rivière, A. (2009). The different complexity levels in mechatronic design process. In *3rd International Conference on Software, Knowledge, Information Management and Applications (SKIMA2009)*. Fes, France. 21-23 October.
- Poppendieck, M., & Poppendieck, T. (2003). *Lean software development – an agile toolkit for software development managers*. Boston: Addison-Wesley.
- Pratt, M. J. (2001). Introduction to ISO 10303 - the STEP standard for product data exchange. *Journal of Computing Information Science in Engineering*, 1(1), 102–103.
- Rachuri, S., Fenves, S., Sriram, R., & Wang, F. (2005). A product information modeling framework for product lifecycle management. *Computer-Aided Design*, 37(13), 1399–1411.
- Rahmani, K., & Thomson, V. (2012). Ontology based interface design and control methodology for collaborative product development. *Computer-Aided Design*, 44(5), 432–444.
- Robin, V., & Girard, P. (2006). An integrated product-process-organisation model to manage design system. In *IMACS Multiconference on Computational Engineering in Systems Applications* (pp. 1287–1293). Beijing, China. 4-6 October.

REFERENCES

- Robin, V., Rose, B., & Girard, P. (2007). Modelling collaborative knowledge to support engineering design project manager. *Computers in Industry*, 58(2), 188–198.
- Royce, W. W. (1970). Managing the development of large software systems. *IEEE WESCON*, 26(8), 328–288.
- Saaksvuori, A., & Immonen, A. (2002). *Product lifecycle management* (Springer.). Berlin, Germany.
- Samad, T., & Parisini, T. (2011). Systems of Systems. *The Impact of Control Technology*, 12(1), 175–183.
- Sanders, E. B.-N., & Stappers, P. J. (2008). Co-creation and the new landscapes of design. *CoDesign*, 4(1), 5–18.
- Schöner, H.-P. (2004). Automotive mechatronics. *Control Engineering Practice*, 12(11), 1343–1351.
- Sellgren, U. (2006). Interface modeling - a modular approach to identify and assess unintended product behavior. In *NAFEMS 2nd Nordic Seminar: Prediction and Modelling of Failure Using FEA*. Roskilde, Denmark. 31 May-1 June.
- Sellgren, U., Törngren, M., Malvius, D., & Biehl, M. (2009). PLM for Mechatronics integration. In *International Conference on Product Lifecycle Management*. Bath, UK. 6-8 July 2009.
- Seyff, N., Maiden, N., Karlsen, K., Lockerbie, J., Grünbacher, P., Graf, F., & Ncube, C. (2009). Exploring how to use scenarios to discover requirements. *Requirements Engineering*, 14(2), 91–111.
- Shen, W., & Barthès, J. (2008). Special Issue on collaborative design and manufacturing, Advanced Engineering Informatics. *Advanced Engineering Informatics*, 22(3), 281.
- Shetty, D., & Kolk, R. A. (2010). *Mechatronics system design, SI version. Mechatronics System Design: SI*. Boston, USA: Cengage Learning.
- Smith, G. L. (2002). Utilization of STEP AP 210 at The Boeing Company. *ComputerAided Design*, 34(14), 1055–1062.
- Sohlenius, G. (1992). Concurrent engineering. *CIRP Annals - Manufacturing Technology*, 41(2), 645–655.
- Sosa, M. E., Eppinger, S. D., & Rowles, C. M. (2000). Designing modular and integrative systems. In *ASME 2000 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Baltimore, USA. 10-13 September.
- Stark, J. (2011). *Product lifecycle management* (pp. 1–16). Springer London.

REFERENCES

- Stelzmann, E. (2011). Contextualizing agile systems engineering. *IEEE Aerospace and Electronic Systems Magazine*, 27(5), 163–167.
- Steward, D. V. (1965). Partitioning and tearing systems of equations. *Journal of the Society for Industrial & Applied Mathematics*, 2(2), 345–365.
- Stone, R., & Wood, K. (2000). Development of a functional basis for design. *Journal of Mechanical design*, 122(4), 359–370.
- Sun, H., Fan, W., Shen, W., & Xiao, T. (2012). Ontology-based interoperation model of collaborative product development. *Journal of Network and Computer Applications*, 35(1), 132–144.
- Tang, H., Guo, S., Huang, L., Li, L., & Li, Y. (2015). Research and development on key models and technology of PDM system. *The International Journal of Advanced Manufacturing Technology*, 78(9-12), 1865–1878.
- Tang, X., & Yun, H. (2008). Data model for quality in product lifecycle. *Computers in Industry*, 59(2-3), 167–179.
- Terzi, S., Bouras, A., Dutta, D., Garetti, M., & Kiritsis, D. (2010). Product lifecycle management—from its history to its new role. *International Journal of Product Lifecycle Management*, 4(4), 360–389.
- Thramboulidis, K. (2005). Model-integrated mechatronics—toward a new paradigm in the development of manufacturing systems. *IEEE Transactions on Industrial Informatics*, 1(1), 54–61.
- Tichkiewitch, S. (1994). De la CFAO à la conception intégrée. *Revue internationale de C.F.A.O. et d'infographie*, 9(5), 609–621.
- Tomizuka, M. (2002). Mechatronics: from the 20th to 21st century. *Control Engineering Practice*, 10(8), 877–886.
- Torry-Smith, J. M., Mortensen, N. H., & Achiche, S. (2013). A proposal for a classification of product-related dependencies in development of mechatronic products. *Research in Engineering Design*, 25(1), 53–74.
- Tucoulou, J. C., & Gouyon, D. (2013). RobAFIS student competition: from systems engineering to the engineering of one system. *Insight Journal of INCOSE*, 16(4), 7–10.
- Van Beek, T. J., Erden, M. S., & Tomiyama, T. (2010). Modular design of mechatronic systems with function modeling. *Mechatronics*, 20(8), 850–863.
- Vasi, V. S., & Lazarevic, M. P. (2008). Standard industrial guideline for mechatronic product design. *FME Transactions*, 36(3), 103–108.

REFERENCES

- VDI 2206. (2003). *Design handbook 2206. Design methodology for mechatronic systems*. Düsseldorf: VDI Publishing Group.
- Vegetti, M., Leone, H., & Henning, G. (2011). PRONTO: An ontology for comprehensive and consistent representation of product information. *Engineering Applications of Artificial Intelligence*, 24(8), 1305–1327.
- Wang, L., Shen, W., Xie, H., Neelamkavil, J., & Pardasani, A. (2002). Collaborative conceptual design state of the art and future trends. *Computer-Aided Design*, 34(13), 981–996.
- Warmer, J. B., & Kleppe, A. G. (1998). *The object constraint language: precise modeling with UML*. Boston, USA: Addison-Wesley Object Technology Series.
- Watson, D. (2003). Model-based autonomy in deep space missions. *IEEE Intelligent Systems*, 18(3), 8–11.
- Weber, R. H., & Weber, R. (2010). *Internet of things*. Berlin, Germany: Springer.
- Wells, G. D., & Sage, A. P. (2009). Engineering of a System of Systems. In *System of systems engineering, innovations for the 21st century*. Boca Raton, USA: John Wiley & Sons, Inc.
- Wikander, J., Törngren, M., & Hanson, M. (2001). Emphasizing team building in a problem- and project-based curriculum to meet the challenges of the interdisciplinary nature of this field. *Robotics & Automation Magazine*, 8(2), 20–26.
- Womack, J. P., Jones, D. T., & Roos, D. (2007). *The Machine that Changed the World: The Story of Lean Production – Toyota’s Secret Weapon in the Global Car Wars that is Now Revolutionizing World Industry*. Free Press.
- Xu, X. W., & Liu, T. (2003). A web-enabled PDM system in a collaborative design environment. *Robotics and Computer-Integrated Manufacturing*, 19(4), 315–328.
- Yoo, S. B., & Kim, Y. (2002). Web-based knowledge management for sharing product data in virtual enterprises. *International Journal of Production Economics*, 75(1-2), 173–183.
- Zha, X. F., & Du, H. (2006). Knowledge-intensive collaborative design modeling and support. *Computers in Industry*, 57(1), 39–55.
- Zha, X. F., Fenves, S. J., & Sriram, R. D. (2005). A feature-based approach to embedded system hardware and software co-design. In *ASME Design Engineering Technical Conference*. Long Beach, USA. 24-28 September.
- Zheng, C., Bricogne, M., Le Duigou, J., & Eynard, B. (2014a). Mechatronic design process: a survey of product model. In *24th CIRP Design Conference*. Milano, Italy. 14-16 April.

REFERENCES

- Zheng, C., Bricogne, M., Le Duigou, J., & Eynard, B. (2014b). Survey on mechatronic engineering: a focus on design methods and product models. *Advanced Engineering Informatics*, 28(3), 241–257.
- Zheng, C., Le Duigou, J., Bricogne, M., & Eynard, B. (2013). Survey of design process models for mechatronic systems engineering. In *10ème Congrès International de Génie Industriel CIGI2013*. La Rochelle, France. 12-14 June.

List of publications

Journal papers

Zheng, C., Bricogne, M., Le Duigou, J., & Eynard, B. (2014). Survey on mechatronic engineering: a focus on design methods and product models. *Advanced Engineering Informatics*, 28(3), 241–257.

Zheng, C., Bricogne, M., Le Duigou, J., & Eynard, B. (2015). Multidisciplinary interface model for design of mechatronic systems. *Computers in Industry*, doi: 10.1016/j.compind.2015.12.002.

Zheng, C., Bricogne, M., Le Duigou, J., & Eynard, B. (2015). Design process for complex systems engineering based on interface model. *INCOSE Insight*, in press.

International conference papers

Zheng, C., Le Duigou, J., Bricogne, M., & Eynard, B. (2015). Multidisciplinary interface modelling: a case study on the design of 3D measurement system. In *12th International Conference on Product Lifecycle Management*. Doha, Qatar. 19-21 October.

Zheng, C., Bricogne, M., Le Duigou, J., & Eynard, B. (2014). Mechatronic design process: a survey of product model. In *24th CIRP Design Conference*. Milano, Italy. 14-16 April.

Zheng, C., Le Duigou, J., Bricogne, M., & Eynard, B. (2013). Survey of design process models for mechatronic systems engineering. In *10ème Congrès International de Génie Industriel CIGI2013*. La Rochelle, France. 14-16 April.

Workshop and seminar papers

Zheng, C., Le Duigou, J., Bricogne, M., & Eynard, B. (2015). Interface modelling for multi-disciplinary integration: application to design of mechatronic systems. In *Product and Asset Lifecycle Management PALM2015*. Belfort-Montbéliard, France. 1-4 July.

Zheng, C., Le Duigou, J., Bricogne, M., & Eynard (2015), An interface modelling approach for the integrated design of mechatronic systems. In: *20èmes Journées STP du GdR MACS*, Troyes, France. 5-6 February.

Zheng, C., Le Duigou, J., Bricogne, M., & Eynard (2014). Design process for complex systems engineering based on interface modelling. In: *6ème Forum Académie Industrie de l'AFIS*, Cergy Pontoise, France. 11-12 December.

ABSTRACT/RÉSUMÉ

Abstract

Mechatronic system is considered as a synergetic combination of mechanical engineering, electronic engineering and computer engineering. Such mechatronic system has its own lifecycle and should integrate different disciplines and various technologies. Therefore the design of mechatronic systems becomes increasing complex. In order to propose an approach to achieve a better functional and spatial integration of mechatronic systems, especially to achieve a higher integration of different disciplines during the design process of mechatronic systems, two kinds of problems must be overcome. The first problem is related to design data of mechatronic systems while the second is related to the design process.

The contribution of the thesis is based on two complementary concepts. The first contribution, the multi-disciplinary interface model, is proposed to address the issue of design data. These interfaces are based on the system architecture and specify which transfers exist between components designed by the project teams of different disciplines. Instantiated in the data model, multi-disciplinary interfaces enables a better data exchange and sharing among the engineers of different disciplines. The second concept concerns the design method based on the multi-disciplinary interface model. This method is proposed to establish the process for mechatronic engineering in order to achieve a better multi-disciplinary integration for the design of mechatronic systems.

Finally, the two propositions are then implemented in a demonstrator developed based on 3DEXPERIENCE Platform. A 3D measurement system, considered as a synergistic combination of mechanical engineering, electronic engineering, computer engineering and optical engineering, is used to demonstrate and validate the propositions of the thesis in terms of multi-disciplinary integration for the design of mechatronic systems.

Key words: Mechatronic system, product model, design method, multi-disciplinary interface, systems engineering

ABSTRACT/RÉSUMÉ

Résumé

Les systèmes mécatroniques sont caractérisés par la combinaison synergique de la mécanique, de l'électronique et de l'informatique en temps réel. Ils possèdent leur propre cycle de vie et doivent associer des expertises métiers et des technologies très variées, ce qui rend leur conception plus complexe et nécessairement plus intégrée.

Afin de mettre en œuvre une approche permettant d'assurer une meilleure intégration fonctionnelle et spatiale des systèmes mécatroniques, et plus particulièrement sur l'axe développement de produit en assurant une meilleure combinaison des expertises métier, deux types de problématiques doivent être surmontés. La première problématique a trait aux données de conception alors que la seconde est relative aux processus.

La contribution de nos travaux de thèse s'appuie sur deux concepts complémentaires. Le premier, un modèle d'interfaces multidisciplinaires, est proposé pour répondre à la problématique relative aux données de conception. Ces interfaces s'appuient sur l'architecture du système et précisent quels transferts existent entre les composants conçus par les différentes disciplines. Instanciées dans le modèle de données, les interfaces multidisciplinaires permettent d'échanger et de partager les informations entre les différentes disciplines. Le second concept concerne la méthode de conception basée sur le modèle d'interfaces multidisciplinaires. Cette méthode est définie pour pouvoir établir le processus d'ingénierie et permettre une meilleure intégration des expertises métiers tout au long de la conception des systèmes mécatroniques.

Enfin, les deux propositions sont implémentées par l'intermédiaire d'un démonstrateur basé sur 3DEXPERIENCE Platform. Un système de mesure 3D, combinaison synergique de la mécanique, de l'électronique, de l'informatique et l'optique, est utilisé afin de démontrer et de valider les contributions par nos travaux en termes d'intégration multidisciplinaire des expertises métier lors de la conception des systèmes mécatroniques.

Mots clés: Système mécatronique, modèle de données, méthode de conception, interface multidisciplinaire, ingénierie système