Ph.D. Thesis

Information content models of objects for communication between engineers

Submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in the Doctoral School of Applied Informatics and Applied Mathematics, Budapest.

Author:
Yatish Bathla

Supervisor:
Dr. László Horváth

May 2020

Final Examination Committee:

Chair:
Dr. József Tar

Members:
Dr. Róbert Fullér
Dr. Márta Takács

Public Defense Committee:

Opponents:
Dr. Gyula Hermann
Dr. inż. Jan Nikodem (Wrocław University of Science and Technology)

Chair:
Dr. József Tar

Secretary:
Dr. Krisztián Kósi

Members of the committee:
Dr. Edit Laufer
Dr. Gábor Renner (MTA SZTAKI)
Dr. Szilveszter Kovács (Miskolci Egyetem)

Date of Public Defense:

Declaration of Authorship

I, Yatish Bathla, declare that this thesis titled, "**INFORMATION CONTENT MODELS OF OBJECTS FOR COMMUNICATION BETWEEN ENGINEERS**" and the work presented in it are my own. I confirm that:

- This work was done while in candidature for a research degree at this University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

PhD Candidate

Date:

"*The science of today is the technology of tomorrow.*"

- Edward Teller.

Contents

List of Figures

List of Tables

Acronyms

3DXML 3DEXPERIENCE file format. 69, 82

AC Activity Contexts. 44, 47, 48

AHP Analytic Hierarchy Process. 96

AIC Active Information Content. i, iv, vi, 5, 10, 16, 87, 88, 91–95, 111

ANFIS Adaptive Neuro Fuzzy Inference System. vii, 104–107

ANP Analytical Network Processes. 96

API Application Programming Interface. i, 4, 50, 52, 57, 72, 74, 85, 111, 113

BiC Behavior Info-Chunk. 4, 43–45, 47, 49, 50, 63, 65, 66, 68, 73, 78

BMX Behavioral Modeling Extension. 11

CAD Computer Aided Design. iii, vi, 11, 53, 69, 72, 73, 75–78, 81–83, 85

CiC Component Info-Chunk. 4, 31, 32, 34, 36–40, 45–48, 57, 58, 66, 70, 92, 94

CPM Classical Product Model. 2, 8, 10, 11, 13, 14, 111

CPS Cyber Physical System. i, vi, 5, 10, 15, 16, 87, 88, 90–95, 110, 111, 113

CSS Cascading Style Sheets. 4, 53

CxiC Context Info-Chunk. 4, 43, 44, 46–50, 63–66, 68, 73, 78

DC Drive Contexts. 44, 47, 48

DHA Discrete Hybrid Automata. 15

DKC Driving Knowledge Content. 14

EMA Enterprise Management Agent. 69

EMS Engineering Model System. i, iv, vi, 5, 10, 87–90, 93–95, 111

EO Engineering Object. 9, 16, 17, 22, 26, 28, 33, 34, 53, 54, 98, 110

EOs Engineering Objects. i, 3, 8, 16, 19–23, 25–28, 42, 98

ER Entity Relationship. 43, 57, 69–71, 78

ES Engineering structure. vi, 14, 88, 90

FC Feature Contexts. 44, 47, 48

FIS Fuzzy Inference System. i, vi, vii, 5, 97, 100–109, 112

GUI Graphical User Interface. 105, 107

HCI Human Computer Interaction. 3, 9, 16, 39, 73

HTML Hypertext Markup Language. 4, 53

HTTP Hypertext Transfer Protocol. 74

HYSDEL Hybrid System Description Language. 15

IBCA Initiative Behavior Context and Action. iii, 10, 14, 17, 42–44, 47–49, 77–79, 81, 85

IC Information Content. i, v, vi, 2–5, 9, 10, 14, 15, 17, 19–31, 37–40, 42–44, 48, 49, 52–58, 63, 69–79, 81–89, 94, 108, 110–113

IFP Interaction Feature Pair. 13

IP Intelligent Property. 10, 14–17, 42, 44, 47, 49, 75, 77–79, 81, 85, 86, 91, 111, 113

IVPS Intelligent Virtual Product Space. 12, 31, 39

JDBC Java Database Connectivity. 53

JSP Java Server Pages. 4, 53

JSTL Java Server Pages Standard Tag Library. 53

LiC Layer Info-Chunk. vi, 69, 72, 77–79, 88–90, 92–95, 111

LiCF Functional Layer Info-Chunk. 4, 31, 36, 37, 39, 40, 43–45, 47, 50, 52, 57, 58, 63, 66, 69, 70, 92, 94

LiCL Logical Layer Info-Chunk. 4, 31, 32, 34, 37–40, 43–48, 50, 52, 57, 58, 63, 64, 66, 69, 70, 92, 94

MAAD Multilevel Abstraction based Self-Adaptive Definition. i, iii, v, vi, 3, 4, 9, 10, 17, 42–46, 48, 50–52, 56, 57, 63, 72, 77–80, 85, 111

MBSE Model Based Systems Engineering. 2, 9

MTZ Modified Thorup Zwick. 17

OOP Object Oriented Principle. 4, 11, 43, 53, 56, 68, 73, 92, 94

OR Object Relationship. 43, 57

PBM Part Behavior Model. 11

PFM Part Function Model. 11

PLM Product Lifecycle Management. i, 2, 3, 5, 7–10, 17, 47, 112

PR Product Realization. 14

RBA Request Behavior and Actions. 14

RBCD Requirement Behavior Context Driven. 14

RFL Requirement Functional Logical. iv, 5, 96, 97

RFLP Requirement Functional Logical Physical. i, iii–vi, 2–5, 8–10, 14, 15, 17, 19, 24, 28–38, 40–50, 52, 56–58, 69, 70, 72, 73, 75, 77–79, 81, 86–94, 96, 97, 100, 102, 108–113

RPBD Request based Product Behavior Driven. 13

SB Situation defining behaviors. 47

SFiC Sub-Function Info-Chunk. 4, 31, 35–37, 39, 40, 45, 47, 57, 58, 66, 70, 92, 94

STEP Standard for the Exchange of Product Model Data. 8, 26

TS Takagi Sugeno. 11

TZ Thorup Zwick. 17

UML Unified Modeling Language. v, vi, 54, 56

VMS Vehicle Management System. 15

WMS Water Management System. 15

Acknowledgements

This thesis is the end of my journey to obtaining my doctorate studies. I have not traveled in a vacuum on this journey.This thesis has been kept on track and been seen through to completion with the support and encouragement of numerous people including my professors, family, friends and colleagues. At the end of my thesis, it is a pleasant task to express my thanks to all those who contributed in many ways to the success of this study.

At this moment of accomplishment, first of all I express my sincere gratitude to my supervisor, Prof. László Horváth.Thanks for giving me the opportunity.This work would not have been possible without his guidance, support and encouragement.Under his guidance, I successfully overcame many difficulties and learned a lot. The discussions I had with him were invaluable. Special thanks to the head of the Doctoral School *Prof. Aurél Galántai* for his kind support and guidance.I would like to say a big thanks to *Prof. Máráta Tákas*, who guides to write high-quality research papers, Prof. Joszef Tar, who helps to polish my concepts in system modeling, *Prof. Szénási Sándor* and *Prof. Vámossy Zoltán*,who provide the teaching opportunity and guidance.

I am grateful to all my friends for being the surrogate family during the four years I stayed in Budapest and for their continued love and support thereafter. My final words go to my family. My mother, *Sushma Bathla*, who always encourage me to study and achieve my dreams, my father *Pawan Bathla* for financial support, my sisters *Deepti Gera Bathla*, *Himani Gulyani Bathla* and brother *Sahil Bathla* for motivation and my wife, *Neha Bathla Batra* for love and care. On a different note, many people Sergio Sanchez Martinez, Daniel Bowley, Jose Evelio Martinez Saiz, Rituraj Gupta, Neerandra Yadav, Naina Mukesh, Mahmood Albkri, Zsolt Kovacshazi, Sister Dorni, Elder Allred have been a part of my Ph.D. Studies and I am highly grateful to all of them.

Abstract

The research is about Behavioral modeling and Behaviors representations of the multidisciplinary product for the contextual definition of selected parameters and closely connected product objects in the Product Lifecycle Management (PLM) system. Research starts with the Requirement Functional Logical Physical (RFLP) structure multidisciplinary product model and then focus on the Information Content (IC) to handle the RFLP structure indirectly through the Multilevel Abstraction based Self-Adaptive Definition (MAAD) structure. The Community zone and Process plane are proposed in the IC for the structured organization of Engineering Objects (EOs) in the Multidisciplinary product model. Then, Info-Chunk entities are proposed in the Functional level and Logical level of the RFLP structure for the Behavioral Modeling of the multidisciplinary product. Further, Info-Chunk objects are proposed in the RFLP and MAAD structure for the Behaviors representation of the Multidisciplinary product. For practical implementation, InfoChunkLib Application Programming Interface (API) is proposed in the IC for the Web application and the Content Web server is proposed to store the modeled behaviors information and communicate with the modern Engineering system based web servers. Later, the research work extends to the Cyber Physical System (CPS) modeling and Fuzzy Logic. Here, Info-Chunk entities based RFLP structure are proposed in the Active Information Content (AIC) and extended Engineering Model System (EMS) for the Behavioral modeling of the multidisciplinary CPS. Finally, the Requirement block, Functional block and Logical block are proposed in the RFLP structure for the multidisciplinary product behaviors evaluation using the Fuzzy logic. The MATLAB toolbox, Fuzzy Logic toolbox (Mamdani Fuzzy Inference System (FIS) and Sugeno FIS) is used to validate the concepts.

Part I

Motivation

Chapter 1

Introduction

1.1 Research Aim and Problems

At the modern ages of the industrial revolution, the traditional design will be upgraded to the smart design to enter a "smart era". Computer-based simulation is becoming an invaluable asset in the design of complex products. Classical Product Model (CPM) [1] relies upon the highly integrated physical level model of the product. This integration does not support multidisciplinary product engineering because capability cannot provide the demanded unified modeling mechanism for different engineering areas [2]. In product design, multidisciplinary engineering becomes the need of every product. Engineering design is increasingly becoming a collaborative set of tasks among multidisciplinary, distributed design teams [3]. Requirement Functional Logical Physical (RFLP) structure is a complex modeling methodology from the systems engineering and supports the Model Based Systems Engineering (MBSE) process [4]. Currently, a multidisciplinary product assembly is done in the specification tree of the RFLP structure. Behaviors are included in Functional level and Logical level elements of RFLP structure in the context of content on the level of engineering objectives [5]. There is three essential product behavior represented in the RFLP structure namely dynamic logical behavior, global dynamic behavior and discrete behavior as explained in the paper [6]. Discrete behavior is evaluated and represented in Functional and Logical level elements of the RFLP structure. Dynamic behavior analysis [7] is the area of investigation in the Product Lifecycle Management (PLM) structure [8] system. Here, the Behavioral modeling approach is used for the analysis of the dynamic behavior of a product.
To derive an intelligent product in the PLM system, the Product model handing procedures are controlled indirectly by the Information Content (IC)

structure sector [9] [10].This sector is placed between human and information-based product models to enhance the HCI at product development [11] [12] and derive an intelligent product as per the Industry 4.0 [13]. It is applied at the contextual definition of objectives and decisions rather than application as a direct engineering object relationship. For Multidisciplinary product modeling, IC controls the RFLP level by the Multilevel Abstraction based Self-Adaptive Definition (MAAD) structure [14] [15] [16]. While the ability to share product knowledge has increased, it is still inadequate for developing complex products by distributed design teams [17]. There are numerous problems with the current multidisciplinary product model such as structure is formal so that the causes and characteristics of connections are hard to reveal at the development or revision of an existing structure [1], critical for the effective assistance of decision making in the Behavioral modeling of a Multidisciplinary product and management of high number of changes of modeled Engineering Objects (EOs) and representation of background of modeled information in the multidisciplinary product model [10]. As there are multiple disciplines of EOs present in the model, there is an unstructured relationship that exists between them [18]. Hence, a fully integrated product model is required as a virtual prototype for the lifecycle of the multidisciplinary product [19] [14].
Hence, there is a need for a novel approach required for the Behavioral modeling, Behaviors representation, Structured processing of interrelated Engineering Objects (EOs) and a Simplified representation of the complex multidisciplinary product to take coordinated decisions during the modeling [20]. Therefore, this work focus on the above mentioned research problems and propose innovative concepts in the Information Content (IC) to drive the RFLP structure product model considering the Product Lifecycle Management (PLM) systems. The author makes the research to resolve the challenging issues in the multidisciplinary product modeling like behavior representation, behavior evaluation, correlated decision between the multiple disciplines and simplified Human Computer Interaction (HCI) for behavioral modeling.

1.2 Outline of the Theses

The research work started with the structured organization of the engineering objects in the multidisciplinary product model by introducing the concepts of Community zone and Community diagram [Y1] in the Representation layer of Information Content (IC). The Community zones categorized the multidisciplinary product model based on the discipline involved during the product

modeling and Community diagram is the visualization of the Community zone depends on the zone type and discipline. Then, the Process plane [Y2] is proposed in the Engineering objective layer, Contexts layer and Decisions layer of Information Content (IC) to classify the process involved during the multidisciplinary product modeling.
Then, the concepts of Info-Chunk entities [Y3] are proposed in the Functional level and Logical level of the RFLP structure for behavioral modeling of a multidisciplinary product model. Here, Functional Layer Info-Chunk (LiCF) is the high-level entity and retrieves the information from the Functional layer. The Sub-Function Info-Chunk (SFiC) is the low-level entity retrieve the information from the LiCF. Similarly, Logical Layer Info-Chunk (LiCL) is the high-level entity and retrieves the information from the Logical layer of the RFLP structure. Component Info-Chunk (CiC) retrieves the information from the Logical component and placed in the LiCL. The entities of the RFLP structure are mapped with the IC to handle the product behaviors of the multidisciplinary product model. This is done by classifying the IC based on the Engineering discipline and System behavior. In other words, Conceptual models categorized the IC into Discipline content and Behavior content. Behavioral modeling of a multidisciplinary product is done by the contents, which controls the Info-Chunk entities. Hence, it can be an integrated model and can be realized in industrial professional modeling systems by using their functionality for open architecture. For Behaviors representation of a Multidisciplinary product, Info-Chunk entities are converted into the Info-Chunk objects, which are based on the Object Oriented Principle (OOP) [Y4] concepts. Here, Behavior Info-Chunk (BiC) objects and Context Info-Chunk (CxiC) objects are proposed in the Behaviors level and Contexts level of the Multilevel Abstraction based Self-Adaptive Definition (MAAD) structure to model the behaviors of the multidisciplinary product. The Info-Chunk objects are used to store the behaviors related data of the multidisciplinary product and exchange the information with SFiC, LiCF, CiC, LiCL objects of the RFLP structure.
For the practical approach of the Behavioral representations, An Application Programming Interface (API) called "InfoChunkLib" is proposed in the IC by using Info-Chunk objects for the web application and a content web server to store and represent the modeled behaviors information of the IC web Application. The API is coded by using the Java programming language with the JavaFX software platform. The IC web application is coded by using the programming language Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), Java Server Pages (JSP). The Content web server is Apache based. The backend is coded by using the Java Servlets and the database used is PostgreSQL. Therefore, the proposed model is flexible

enough to integrate with the existing modeling in industrially applied engineering systems. Then, an active knowledge model of a multidisciplinary product is proposed for behavioral modeling. Here, various scenarios are proposed to handle the multidisciplinary product application of the PLM system through the IC application [Y5]. Therefore, this model can establish better communication between engineers and modeling procedures during the lifecycle management of product information in computer-based engineering systems.
Then, the author enhances the scope of the research by proposing the Info-Chunk entities driven RFLP structure for multidisciplinary Cyber Physical System (CPS) and Requirement Functional Logical (RFL) blocks by using fuzzy logic concepts. Here, the Discrete behavior of a multidisciplinary product in the real environment is monitored by using Fuzzy logic [Y6]. In this case, the behavior of a product system is monitored by the conceptual model of the IC. Here, Info-Chunk concepts are proposed in the Active Information Content (AIC) and extended Engineering Model System (EMS) [Y7] for the behavioral modeling of the multidisciplinary CPS. Using the Fuzzy logic, Mamdani Fuzzy Inference System (FIS) is used to monitor the behaviors of a multidisciplinary product whereas Adaptive Neuro FIS is used to monitor and improves the behaviors of a multidisciplinary product in the real environment.

Part II

Main Areas of Research

Chapter 2

RFLP Structure for Multidisciplinary Product Modeling

This chapter explains the preliminary research work in the product modeling. Product can be defined as something that is produced by industrial process and it is not possible to create every product in the real world. Some of the influencing factors are investment, feasibility, flexibility, performance, market value of the product. Technology like computer made human life easier and gave shape to the imagination. Evolution of computer started in the mid of 20th century and era of product structure modeling started in the late of 20th century. It is defined as an environment where a product could be modeled in the virtual space. The outcomes are savings of resources and time, less investment and efficient design of a product.

The chapter is structured as follows:
Section 2.1: Introduction
Section 2.2: Multidisciplinary Product Modeling
Section 2.3: Survey of Approaches in the Behavioral Modeling
Section 2.4: Summary of the chapter

2.1 Introduction

Product Lifecycle Management (PLM) is the business activity of managing a company's products effectively all the way across their life cycles from the very first idea for a product until it disposed. It is defined as lifecycle of a product from the very first idea for a product until it disposed. The objective is to increase product revenues and reduce product-related cost [8]. This activity is very popular in industries that it has led to highly integrated engineering systems [21] [22]. Product design is one of the phase of PLM system. The next step was creating standards for the product model. This was achieved in the industries by using the Standard for the Exchange of Product Model Data (STEP) product model which is standardized by ISO 10303 [23]. It is also called as Classical Product Model (CPM). Product modeling structure could be complex for example modeling of aircraft, rocket or a car, where there are huge number of product components and their relationships. It is intended to handle wider range of product-related data and product types (electronic, mechanical, sheet metal, fiber composites, ships, architectural) covering the entire life-cycle of a product [24]. It established the standardized basis of feature based product models [25]. It is standardization of exchange and archival of PLM data, including 3D models and cover all needs of interchangeability. The feature driven CPM [9] is most commonly used for discipline specific product modeling. It consist of Engineering Objects (EOs) and their relationships [12]. Engineer places object, parameter, and relationship definitions in the virtual product space using model construction functions available in the procedural environment [26]. It relies upon highly integrated physical level model of product. But, product modeling is not limited to the physical layer. The separated or only slightly integrated mechanical engineering modeling increasingly demanded multidisciplinary integration [27].

2.2 Multidisciplinary Product Modeling

It is the combination of EOs of multiple discipline and their relationship. Modeling of a multidisciplinary product must have means for the integration of discipline specific models into a model with a unified structure. Contextual connections between model representations of two different discipline related units is possible on the physical level of modeling, integrated definition must be raised to conceptual level of product design. Therefore, a four-leveled structure of the product model using Requirement Functional Logical Physical (RFLP) structure [14] was introduced for multidisciplinary product modeling. It is compliant with the IEEE 1220 standard. It is well known method in systems engineering and well suitable for modeling of industrial product as system [28] because it increases the efficiency of the overall ecosystem used to develop, model, simulate, and deliver the system by providing the deployment solution of the multi-disciplinary collaboration and provides approach for better developing and understanding the complex products. This model considers the product as a system [14]. It is a framework that supports the MBSE process [4]. This structure has four layers i.e.*Requirement layer* for the requirements against the product,*Functional layer* for the functions to fulfill requirements,*Logical layer* for the product wide logical connections, and *Physical layer* for the representations of physically existing objects was organized in the highly contextual connections [27]. Behavior representations accommodate on the Functional and Logical levels of the RFLP structure [14].

2.2.1 Information Content

To derive an intelligent product in the PLM system, Product modeling procedures is controlled by introducing new sector in the product model called Information Content (IC). It is applied at contextual definition of objectives and decisions rather than application as direct EO relationship [12]. The purpose of IC is supporting decision making on EOs. Human can record the required amount of decision background as IC. It replace direct application of knowledge by an indirect method where knowledge is mapped to contextual chain of human intent, engineering objective, contextual connection, and status driven decision. In this way, knowledge controls EOs indirectly through information content. It assists effective communication between engineers of different discipline and information oriented modeling procedures. This sector is placed between engineer and information based product model to enhance HCI at product development [9]. IC was introduced to record and apply content of information that is represented in the product model space [10]. IC based Product Model can be characterized in five levels i.e. *Level of intent of humans*, *Level of meaning of concepts*, *Level of engineering objectives*, *Level of contexts*, *Level of decisions*. The main essence is that engineer controls definition of engineering objects by the definition of intent [12] [9]. It can handle concurrency of engineering activities. However, it cannot handle different influences of the same decision [29] [12]. Here, Influences control EO and their relationships and act through instructions, specifications, standards, and customer demands.
IC was necessary [16] when the RFLP structured product model was implemented in industrial engineering modeling from system engineering. IC [14] drives the RFLP level by the Multilevel Abstraction based Self-Adaptive Definition (MAAD) structure [14]. This structure is used for self adaptive modeling, where the level of *objectives and requests*, *product behaviors*, *contexts*, *actions*, and *feature objects* are applied in order to connect engineers with RFLP implementations [30]. The MAAD modeling [24] methods and model structures are introduced as generalized means for the support of higher level

abstraction based generation of RFLP elements. The MAAD modeling was based on the knowledge representation, contextual change propagation, and extended feature definition capabilities for advanced modeling systems [14]. The active knowledge in multidisciplinary product model has become organized in the form of intelligent property of company. Hence, the driving generation of the RFLP element is done by the Intelligent Property (IP) as well. Also, active knowledge content was represented by Initiative Behavior Context and Action (IBCA) structure and has driving actions on RFLP structure elements [27]. IP controls the RFLP level by the IBCA structure [27].It leads to the analysis of self-adaptive PLM modeling.

In case of Cyber Physical System (CPS) modeling, IBCA structure driving content structure was restructured and extended by Active Information Content (AIC) [31]. The main change was that it included new level for system related IC. Further, IC extension to CPS modeling includes all knowledge and information required to drive object parameter generation procedures on other levels of Extended Engineering Model System (EMS). It needs higher level abstraction than the conventional physical level modeling. The IC driven RFLP structured product model is suitable model for extension to CPS related contexts.

2.3 Survey of Approaches in the Behavioral Modeling

The author has conducted the survey of behavioral modeling of the CPM, RFLP structure and CPS [Y8]. The survey data of CPM behavioral modeling approaches are arranged in the ascending order between the 1998 and 2014.

- In the year 1998, the behavioral modeling approach called modularized modeling approach [2] was defined which deals with models of subsystems and adopts the object-oriented approach in system modeling. Here, shape and behavior are encapsulated as an object in the classical product model as per the object-oriented principle. This approach was based on four steps: domain decomposition, interface definition, structuring, and modularization where each submodel was associative to the geometric master model. This approach reduced the complexity of systems modeling by dividing the system into submodels, called as behavior features. Furthermore, behavior module aggregated the behavior features, their relations, and the interfaces to other behavior models and defined the hierarchy of condensed models.

- In 2000, behavioral modeling of CPM [32] was redefined by three pillars: smart models, objective-driven design and an open extensible environment. Furthermore, adaptive process features were outlined in the advanced feature based modeling called Application features and Behavioral features. Here, smart models build the intelligent products using adaptive process features and results the product. The design of the product was optimized to meet the objectives in the objective-driven design process. Furthermore, existing external applications could be used by the product to increase design flexibility and reliability. Based on the above mentioned concepts, PTC launched Creo Behavioral Modeling Extension (BMX). Later in 2000, modeling approach called Port-based modeling paradigm [33] was delineated by component model and interaction model. This approach followed cyclic procedure between function, form and behavior. A component object was defined which is the combination of both forms and behavior that create a virtual prototype of the system.

- In 2002, a modeling approach was introduced on the basis of the behavior of a part in the CPM that was characterized by its function(s) and achieved through interactions with other part of assembly under a set of operating conditions [34]. Here, behavior of a component controlled the transformation between physical entities that was governed by part geometry and physical laws associated with the entities. Part Function Model (PFM)and Part Behavior Model (PBM) was defined, where PFM generated various product specifications of a part by constituting the set of spatial and design functional relationship and PBM explained the relationship between part function, part geometry and part behavior. Behavior-PFM map was described on the basis of part behavior, classification of geometric interaction and geometric nature of the surface. This model stored the PFM model information with an OOP based Computer Aided Design (CAD) system called Concentra's concept modeler. Later in the 2002, modeling approach was explained [35] using Takagi Sugeno (TS) and polytopic models in the CPM. The solution of this work concluded the trade- off between number of components and accuracy.

- In 2003 and 2004, behavioral modeling approached with intelligent content [36] [37] was outlined which involved specifications and knowledge for the design processes. Petri net model representation of engineering objects [36] in the CPM was explained where an active modeling was introduced in the year 2003. The active model is knowledge driven that has capability of reaction using behavior related knowledge and acts as an intelligent design of the modeled objects and captures the new decisions and intents. Furthermore, generic manufacturing process model was explained with four-leveled structure (levels of process (Level 1), setup (Level 2), operation (Level 3) and numerical control cycle (Level 4)) that described the set of process features and their relations. This modeling approach was extended in the year 2004, where author introduced the circumstance factor in the behavior of modeled objects [37]. Also, part model object was integrated into classical product model that involved analysis of structure and behavior features. The knowledge content of active modeling included the nonlinear mathematical optimization by numerical algorithms. Furthermore, a new feature of intelligent modeling called automatic contextual change of model was introduced in the approach that focused on mathematical correction of automatic fitting into the new environment and reconstruction of the old environment without any additional human interaction. In 2004, modeling approach called active part modeling [38] was applied to the classical product model that comprises knowledge from three sources, namely modeling procedure, generic part model and designer. It was defined as behavior of modeled objects in different circumstances and acted as agents after exchange them with other modeling systems at applications of models. Also, feature models were used for the specifications and knowledge in the design processes that simulate behavior of the modeled objects where comprehensive groups of features were defined. Here, feature definitions were stored in feature library in the modeling system. Furthermore, goal-directed behavioral representation in agent-based modeling of engineering objects was defined that advances the simulation by emulation of intelligence with the automatic contextual change of model feature.

- In 2007, behaviors were represented as design objectives and computational intelligence in the classical product model controlled the behaviors of modeled objects [5]. Here, modeled object was characterized by the inherent and specified behaviors where concept of affect zone was used so that any change in the object might affect other objects of its zone if the related situation was changed. Furthermore, Intelligent Virtual Product Space (IVPS) was introduced in the classical product model. IVPS composed by the four sectors, namely development sector, behavior sector, interface sector, and learning sector. Here, behaviors were created as the objects in the development sector, analysis and rules were applied to the analyzed behavior object in the behavior sector. Later, in 2007 [39], product lifecycle knowledge specific engineering activities were analyzed by describing the information content in the data oriented CPM. Here, the behavior of the modeled object was defined in the engineering objectives level. This content provided the situation for decision-making by evaluating the processes for modeled objects. In 2013, behavior design approach was applied to the CPM [40], where a software application was being developed that optimized product performance in the design phase by taking into account use conditions and requirements so that user technical system was designed. Here, a conceptual model was proposed to help the designer to define each required task to fulfill system functions to improve its performance.

- In 2013, modeling approach Interaction Feature Pair (IFP) was outlined [41] to the classical product model by utilizing the constituent elements of an IFP as state variables. Here, product modeling framework module based on a concept of IFP was developed that embodied information of interaction type, related feature pairs and behavioral information that fulfilled the interactions. Furthermore, S-Space spanned by six basic IFPs was defined that can model the structure of IFPs through operators and functions. Later in the year 2013, elements of behavioral modeling approach to a CPM was defined by using product request feature definition, product behavior feature definition, and context structure feature definition [42] where new behavior level product definition process controlled the process of the feature level product. Furthermore in the same year, situation based product feature to a CPM was proposed [43]. Here, feature level product definition considered the parameters of behaviors and composition of situations by the circumstances.

- In 2014, Request based Product Behavior Driven (RPBD) method of product definition and contextual connections of features was explained [44]. Here, behavior features drive the active knowledge features directly or actions on active knowledge features of knowledge content product model. Later in the year 2014, behavioral modeling of a switched reluctance generator for aerospace applications was applied [45] to the CPM. This model reproduced the average behavior of the input output variables that are required for system-level analysis of the aircraft power distribution system. Its advantage is that there was no need for a detailed knowledge of the equipment structure. Here, parameterization method was explained by obtaining the output impedance and applied to the experimental system and a set of load step tests have been carried out both experimentally and by simulation, and the results from both tests had been compared. In all cases, the model had reproduced with good accuracy the actual system response.

The survey data of RFLP structure behavioral modeling approaches are arranged in the ascending order between the 2014 and 2017.

- In 2014, behavioral modeling approach was introduced to the RFLP structure product model where new generation of intelligent engineering systems were defined by behavior definition for function (F) and logical (L) levels [46]. Here, behavior assisted F and L level product definition collected information from the requirements (R) level and drove generation of physical (P) level of product model. Also, Requirement Behavior Context Driven (RBCD) structure was applied to the levels of RFLP structured product model with the objective of flexible human control by active knowledge definition.

- In 2015, product behavior centered Initiative Behavior Context and Action (IBCA) structure for RFLP structure based product model was defined [47]. The main purpose of IBCA structure was to organize active Intelligent Property (IP) which was used by the Product Realization (PR) model structure in the P (Physical) level of RFLP structure based product model. Furthermore, representation and application of IC at RFLP structure based product definition was the contribution of IBCA structure [48] that lead to the self adaptive RFLP structure based product model. This is done by creating the substructures called situation substructure, Function substructure, Behavior substructures, and Context substructures in structure level of IBCA structure. The Request Behavior and Actions (RBA) knowledge content structure was introduced in the RFLP structure based product model [49]. It concentrated on modeling of request content driven product behavior that was driven by human request for the influence of product definition.

- In the year 2016, IBCA structure was proposed as representation of Driving Knowledge Content (DKC) for application at generation of elements and features in RFLP structured product model, which focused on the intelligent computing and multi-physics in product behavior centered engineering, and organizing knowledge background in IP [50]. Engineering structure (ES) was defined in the RFLP structured product model where content was defined as information or knowledge [19]. This content is also applicable to the CPM modeling. Also, IBCA structure was used for the representation of information and knowledge in the background of decisions on objects and their parameters which organized IP content.

- In the year 2017, IC structure was supported for contextual driving of element generation in the RFLP structured model [51] that achieved better support of IP sourced, product system model based, and multidisciplinary engineering grounded engineering activities for lifecycle of products.

CPS is defined as the physical system with the power of computing. Here, behavior is defined by both computational and physical parts of the system. The survey data of CPS structure behavioral modeling approaches are arranged in the ascending order between the 2012 and 2017.

- In the year 2012, challenges precisely related to the CPS behavioral modeling are [52] as follows: (i) Models with solver-dependent, nondeterminate, or zeno behavior (ii) Modeling interactions of functionality and implementation (iii) Modeling distributed behaviors. The suggested solution was hybrid system modeling and simulation, concurrent and heterogeneous models of computation, the use of domain-specific ontologies to enhance modularity, and the joint modeling of functionality and implementation architectures technologies that provided partial solution of CPS modeling. Here, authors of above mentioned paper considered Vehicle Management System (VMS) for their analysis and focused on the fuel management subsystem to illustrate the modeling challenges.

- In 2013, behavior model of a CPS [53] is build and analyzed by adopting the Discrete Hybrid Automata (DHA) modeling frame and Hybrid System Description Language (HYSDEL). Here, trajectories of the continuous states was simulated using MATLAB which overcomes the problems of uncertainty and concurrency of computing-physical interaction.

- In 2014, run-time behavior of an Water Management System (WMS) using EPANET as water network simulator [54] was discussed. The main goal of this CPS model was to optimize the system using computation like changing pipe sizing, system performance indicators, demand-driven and pressure-driven. This experiment was performed using tools WaterCAD, EPACAD and Matlab.

- In 2017, IP support provided for CPS modeling [31] was outlined by introducing the AIC structure in the RFLP structure product model to establish CPS enable information content and multilevel transfer structure to connect AIC structure to various organized IP environments.

2.3.1 Complex Issues

The major issues in behavioral modeling of product occurs due to its complexity. There are numerous problems with the current product model such as structure is formal so that the causes and characteristics of connections are hard to reveal at the development or revision of an existing structure [1], critical for the effective assistance of decision making in product modeling [10] and management of high number of changes of modeled EOs and representation of background of modeled information in product models [9]. Also, challenging tasks to organize the EOs defined by the product model as well as track relationship of their EOs with other objects. For existing product, Human wastes a lot of time calculating parameter and structure of modeled EO defined by the other engineer and understanding complex relationship between EOs. Moreover, it is difficult for human to interact with EOs which are not related to their discipline. Level of difficulty increases with the large and complex product. Any change can affect the process of whole product. Also, problem arises during the management of high changes in modeled EOs, representation of modeled information, processing of unstructured dependencies, tracking the creation of large and complex product model and revision of parts that influence the other parts of product model. There is strong requirement of efficient decision support on parameters of engineering object of associative connection. Also, it has limitation that manual tracking of changed EO and its connected chain of EOs consumes an lot of time and responsible for error and mistakes.
Also, it requires coordination of huge amounts of discipline specific model information. During the HCI, the difficulties faced by the humans are proper representation of complex product data gathered on the layers, unavailability of content interaction and structured processing of interrelated engineering objects to obtain coordinated decisions. Construction of a product in model space utilizes high number of modeling procedures for creating elements, their structural and dependencies. It also requires engineering activities with high level multidisciplinary and high number of areas of expertise. It is difficult for the engineer of certain discipline to analyze the product structure thoroughly because there are large number of multiple discipline EOs present in the model and unstructured relationship between them. It is more challenging for new engineer to analyze the existing product model because there is

no place to get the information about specification of engineering object and their relationship. As a result, Modification in any EO leads to affect the complete life cycle of a product.

2.4 Summary

This chapter explain basic concepts of Multidisciplinary product model. Further, IC was explained whose purpose is to record and apply content of information that is represented in the product model space. This content drive the RFLP level by MAAD structure. Also, generation of RFLP element is done by the IP of IBCA structure. In the early stage of the research work, I have proposed Modified Thorup Zwick (MTZ) routing algorithm [Z1] for the future internet considering the PLM systems. Here, I have tried to resolve the scalability problem of the internet by making modifications in the Thorup Zwick (TZ) [55] scheme. I have introduced community concept in the topology of the internet, re-define cluster nodes and replace landmark nodes with proposed community nodes in the network. The proposal efficiently organizes the internet by defining a new set of rules for inter-domain nodes and reduces routing table size in a prominent amount. The OMNeT++ discrete event simulator is used to verify the benefits of the proposed routing scheme. Then, I have conducted a survey [Y8] of the state of the art of system behavioral modeling and CPS behavioral modeling is provided in which most of the approaches are reviewed under the PLM system. Then, I have proposed a web application [Y9] that compares existing 3D product modeling software in the market. Here, the statistical study is conducted first to compare the best possible software of the same category in terms of features and operation performed on the product model. The application is used for the practical approach of IC based web application. It is important to note that this chapter is part of preliminary work.

Part III

Results of Research

Chapter 3

Structured Representation of the Engineering Objects

This chapter is the initial point of the research work. The author makes an effort to organize the complex multidisciplinary product model by proposing the concepts of Community zone and Process plane in the Information Content (IC).

The chapter is structured as follows:

Section 3.1: Introduction.
Section 3.2: Concepts of the Community Zone and Community Diagram.
Section 3.3: Concepts of the Process plane.
Section 3.4: Summary the chapter.

3.1 Introduction

Representing and arranging the data of a complex multidisciplinary product model is a challenging task. Therefore, Community zone, Community diagram, and Process plane are proposed in the IC to resolve the issue of the multidisciplinary product model. In this context, the Community zone organizes the EOs of a multidisciplinary product into various zones depending on the number of the disciplines. Then, the Community diagram provides a compact overview of a community zone. The Process plane organizes the behaviors of the multidisciplinary product by categorizing the process involved during the product modeling. In this context, the Analysis process, Contextual process, and Optimisation process are proposed. Then, the Requirement Functional Logical Physical (RFLP) Structure product model is used for the explanation of the Process plane.

3.2 Concepts of the Community Zone and Community Diagram

The research work starts with the multilevel organization of the IC [9]. The author proposes the structured organization of the EOs of a multidisciplinary product by dividing the model space into different zones as shown in Fig. 3.1.

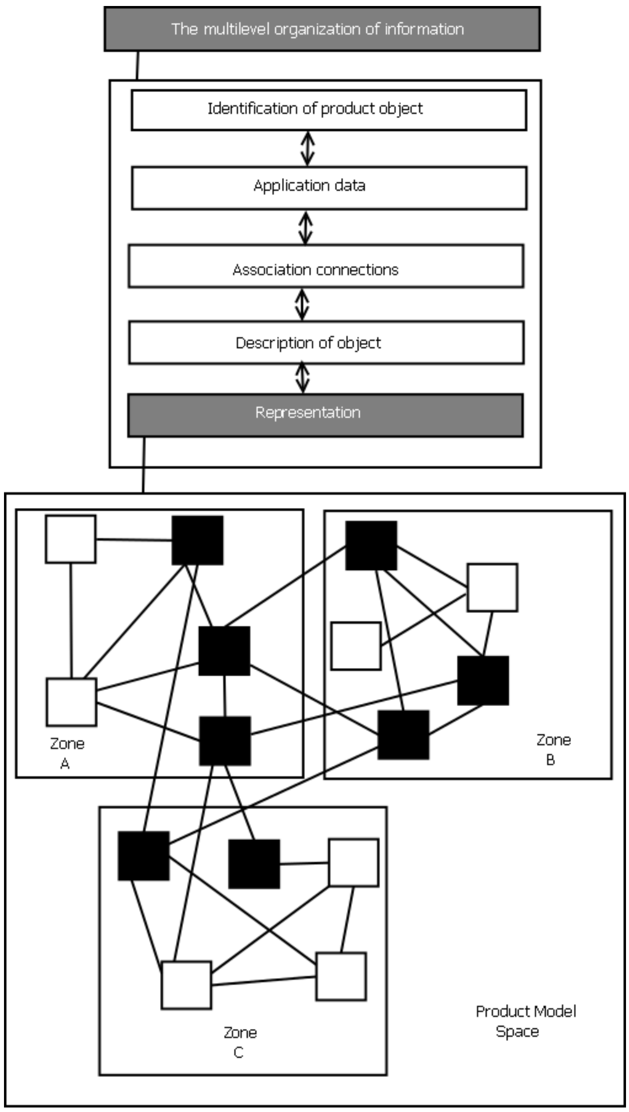


Figure 3.1: Structuring Representation Plane in the Information Content

3.2.1 Community Zone

As stated about the IC, it is the bridge between humans with the information-oriented procedures and product model space. Hence, there must be separate zonal areas in the product model space for engineers of various disciplines where they can conceptualize their work in the form of Engineering Objects (EOs). This lead to the zone concept called Community zone. The multidisciplinary product model space is divided into the various Community zones depends on the engineering discipline. A Community zone is involved with one or more zones depends on the requirements of the product design. It is initialized at the *Intent level* of the Information Content (IC). Fig. 3.2 illustrates the concepts of the Community zones in a multidisciplinary product model space.

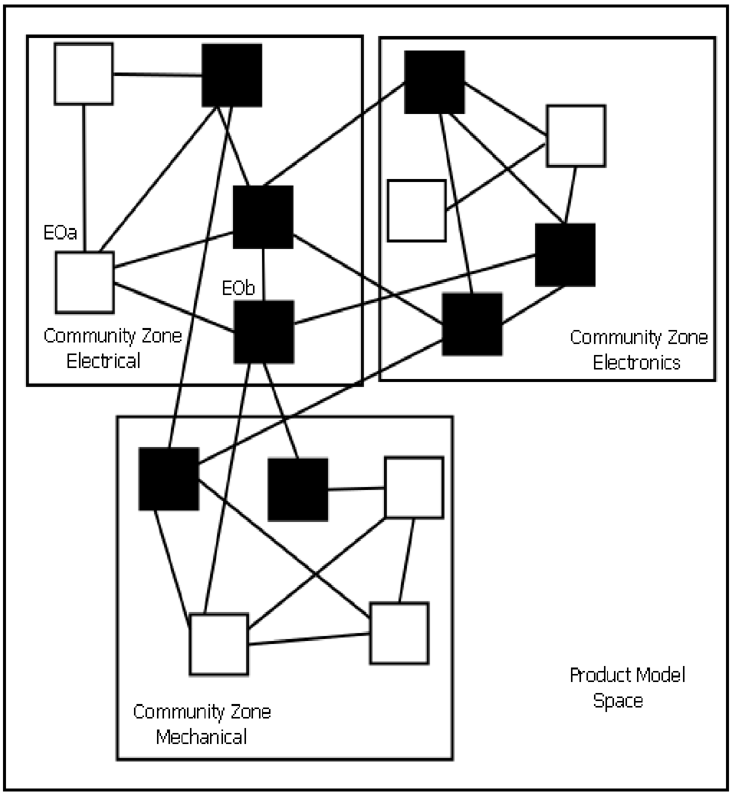


Figure 3.2: Community zones in Product Model space

A Community zone consists of:

- EOs
- Connection of EOs within the community zone

- Connection of EOs outside the community zone

Based on the concept of Community zone, EOs are categorized into two types as follow:

- Single zone EO: It has a relationship within their Community zone. It is represented by the White node as shown in Fig. 3.2 and EO_a is an example.

- Multiple zone EO: It has a relationship within and outside of their Community zone. The amount of visible areas in the neighboring community zones depends on the policies of that community. It is represented by black node as shown in Fig. 3.2 and EO_b is an example.

3.2.2 Community Diagram

After categorizing the EO, the next goal is a simplified representation of EOs of a multidisciplinary product based on the engineering discipline and system behavior. This is done by the community diagram. The Community diagram is the compact visualization of a multidisciplinary product model concerning a community zone in the context of an engineering discipline or system behavior. It defines how much area of a multidisciplinary product model is visible to reduce its complexity. It is represented on the *Representation plane* of the IC. Fig. 3.3 and Fig. 3.4 shows Community Diagram of single zone (EO_a) and multiple zone (EO_b) respectively. The advantage of the Community diagram is that it simplifies the structure of the multidisciplinary product model by visualizing only those areas of product model to the engineers that are required to complete their tasks.

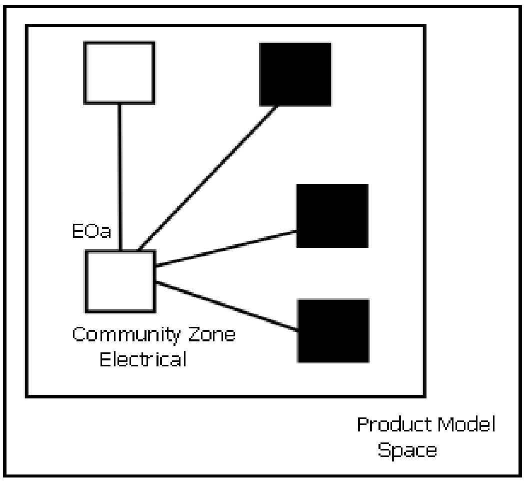


Figure 3.3: Community diagram of a Single Zone EO

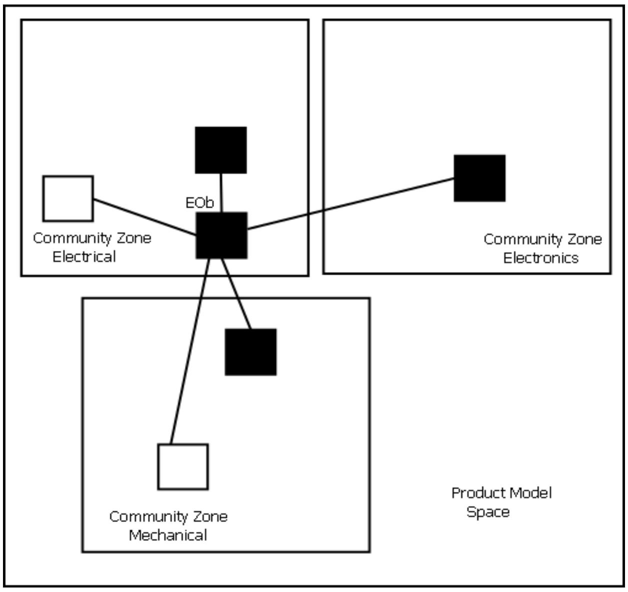


Figure 3.4: Community Diagram of a Multiple Zone *EO*

3.3 Process Plane in the Information Content

The Process plane is proposed in the IC to categorize the number of the process involved during the product modeling. As shown in Fig. 3.5, the Process plane is classified as the Analysis process, Contextual process, and Optimization process.

- Analysis Process: *Level of Engineering objectives* in the IC are described for the behaviors of EOs at certain situations. It is mainly concerned with the purpose of intent. The term, Analysis, is defined as an activity for Engineering objectives to fulfill the requirements of Intent. In the virtual environment, a multidisciplinary product is analyzed in many ways, so that it delivers the expected behavior in the real world. In the context of this work, the Analysis Process is proposed in the *level of Engineering objectives* to observe the behaviors of a multidisciplinary product under different situation and obtain the information from the related behavior.

- Contextual Process: This process is used to define contextual connections between the EOs of a multidisciplinary product depending on the type of analysis. As a result, corresponding changes should be made such that it can configure its parameter and satisfy all the conditions to avoid product failure. It is defined on the *Contexts* layer of the IC.

- Optimization process: After analyzing the behavior of the engineering object, the main challenge is to optimize the analyzed data and make the decision to obtain the best possible solution of data. The optimization process is used to define an activity which provides best the possible value of data according to the type of products. In the context of this work, optimization algorithm is required which computes solutions with the required accuracy. Analyzed data can be linear or nonlinear. Hence, linear or nonlinear optimization is required to obtain the desired data. It is defined on the *Decisions* layer of the IC.

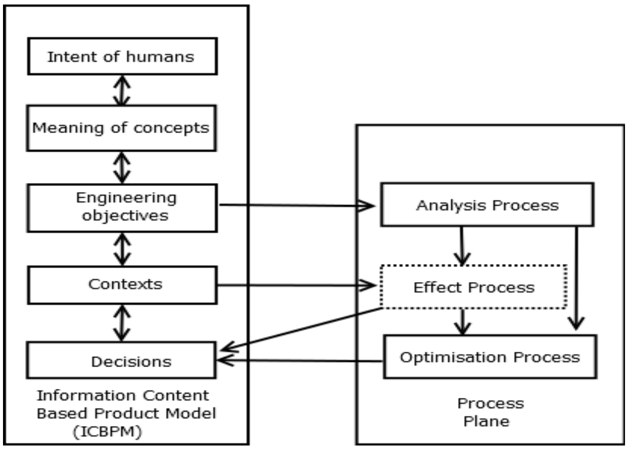


Figure 3.5: Process Plane in the Information Content

Most of the analyses performed on the multidisciplinary product using the RFLP structure are related to Multiphysics analysis, as it can investigate the product behavior. Multiphysics can analyze multi-disciplinary activities of RFLP structure. The multiphysics problem is nonlinear, but most of the solution is obtained by solving a series of linear subproblems. They are inevitably nonlinear. Therefore, related analyzed data are optimized by nonlinear programming. For multidisciplinary product analysis, objectives are to optimize the set of analyzed data as per the requirement of product type and constraints can depend on the system performance, environmental conditions, contextual connection dependency, etc. Basic approaches used to optimize the nonlinear problem are to find local optima and global optima respectively. As there are multiple disciplines involved in the multidisciplinary product, hence multi-discipline optimization is precisely achieved by finding global optima rather than discipline-specific optima [56]. For behavioral modeling, the Process plane is considered as a low-level entity, which stores the information of the process during the product design phase. This is explained in the Fig. 3.6 and Fig. 3.7.

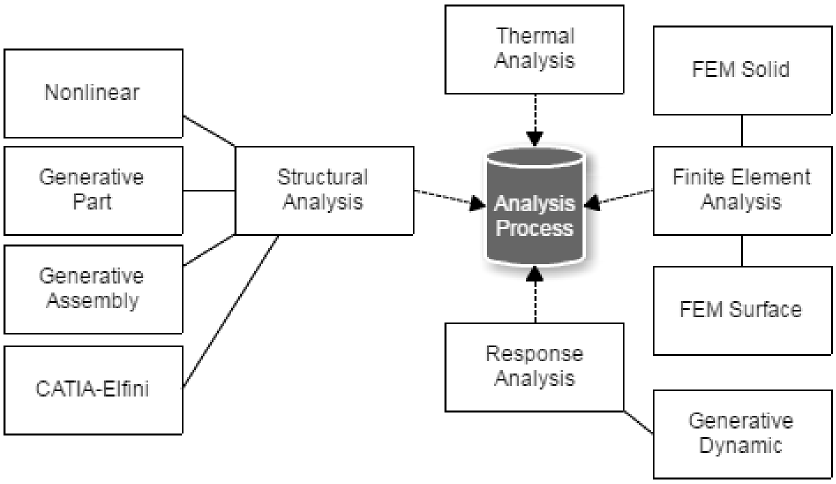


Figure 3.6: Elements in the Analysis Process entity

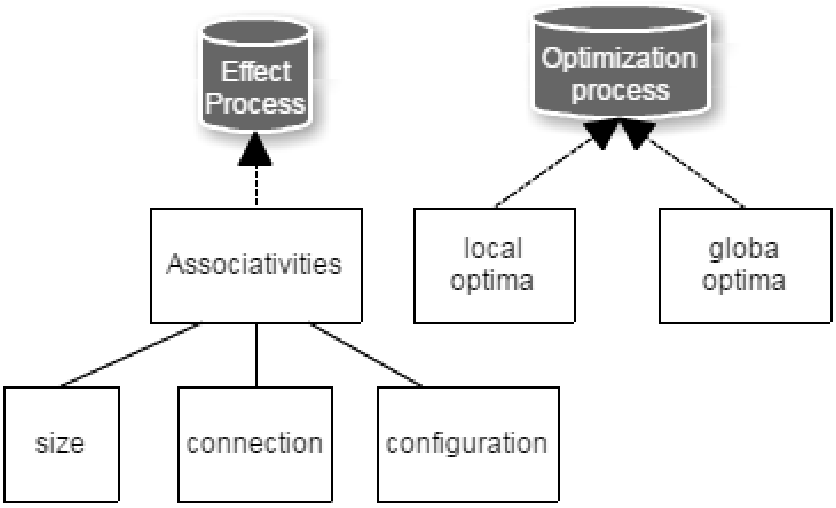


Figure 3.7: Elements in Contextual Process and Optimization Process entity

3.4 Summary

This chapter discusses the structured representation of a Multidisciplinary product model by organizing the EOs in the IC. This is done by introducing Community zones between the unstructured relationship of EO. Product model space is divided into different community zones depends on the disciplines or specifications of the product model. Inside a community zone, EOs are categorized into the single zone and multiple zone EO depends on their connection in the community zone. Community diagram is the compact overview of EO and their relationships belong to certain Community zones of the product model. Further, the Process plane is introduced in the IC. The Analysis process stores the analysis information done on the EOs, Contextual process store the contextual information of EOs and optimization process stores the optimization information of EOs in a product model. Thereafter, the output can be converted into a generic format. The outcomes of the research work will be used for the Behavioral modeling and Behaviors representation of the multidisciplinary product. It will be explained in the next chapters. This chapter covers the Thesis Group 1.

Thesis Group 1

Thesis group 1: Community Diagram and Process Plane in the Information Content

Thesis 1.1

I have introduced the concepts of Community zone and Community diagram [Y1] in the Information Content (IC) for the structured organization of the multidisciplinary product model. The structure of a multidisciplinary product model is formal so that the causes and characteristics of connections are hard to reveal at the development or revision of an existing structure [1] and management of the high number of changes of modeled engineering objects and representation of background of modeled information in product models [9]. Therefore, Community zone is defined as the area in a multidisciplinary product where an engineering discipline can conceptualize their work by performing various tasks on the Engineering Objects (EOs). Hence, a complex ISO 10303 standardized STEP product model is divided into various Community zones depending on the engineering discipline. Inside the zones, EO is categorized as Single zone EO and Multiple zone EO. Finally, a Community zone is demonstrated by the Community diagram in the IC. It is the visualization of a community zone in terms of Single zone EO and Multiple zone EO. Therefore, an engineer can evaluate the multidisciplinary product model relevant to its discipline.

Thesis 1.2
I have introduced the Process plane [Y2] in the Information Content (IC) to organize the process activities for effective decision making during the multidisciplinary product modeling. As mentioned in the paper [10], there are critical issues occur for the effective assistance of decision making in product modeling. Therefore, the Process plane categorizes the Analysis process, Contextual process and Optimization process in the IC. The Analysis process is the set of activities applied for analyzing a multidisciplinary product model. Based on the analysis, the Contextual process stores the information about the process involved in the context of EOs in a multidisciplinary product model. The optimization process is the set of activities to optimize the set of analyzed data. It can be local optima or global optima. Here, Decision is related with the best possible output from the optimized data according to the specification of modeled product. This plane provides an effective decision methodology for representing the behaviors of the multidisciplinary product.

Relevant own publications about this thesis group: [Y1, Y2].

Chapter 4

Info-Chunk Entity for Behavioral Modeling

This chapter focus on the Behavioral modeling of the multidisciplinary product. It is done by proposing the Info-Chunk entities in the Requirement Functional Logical Physical (RFLP) structure. Further, conceptual models of Information Content (IC) are defined with the Info-Chunk entities. This is done in the context of the engineering discipline and system behavior. The chapter is structured as follows:
Section 4.1: Introduction
Section 4.2: Classification of the Information Content.
Section 4.3: Concept of Info-Chunk.
Section 4.4: Configuration for Driving the Model.
Section 4.5: Conceptual Models of Information Content.
Section 4.6: summary the chapter.

4.1 Introduction

A complex multidisciplinary product model generally faces difficulties in making decisions when there is a large number of EOs participating during behavioral modeling. The level of complexity increases in the case of the high number of dependencies among the EOs as there is a vast collection of data gathered from the various engineering disciplines participating in product modeling. In the virtual environment, some of the challenging tasks need definite and correlated information of an engineering discipline, tracking activities of the system behavior, among others. Hence, the establishment of effective assistance in engineering decisions is quite challenging. This chapter proposes the classification of IC as Discipline based content and

Behavior based content. It is done by classifying the Intent so that the decision processes of a complex multidisciplinary product model can take place efficiently. Based on the classification, conceptual models are generated to store the knowledge of engineering discipline and system behavior. It can store and organize the knowledge related to engineering disciplines and system behavior of the product model. This knowledge is stored and applied to take effective decisions so that the complex product model is represented in a simplified manner.

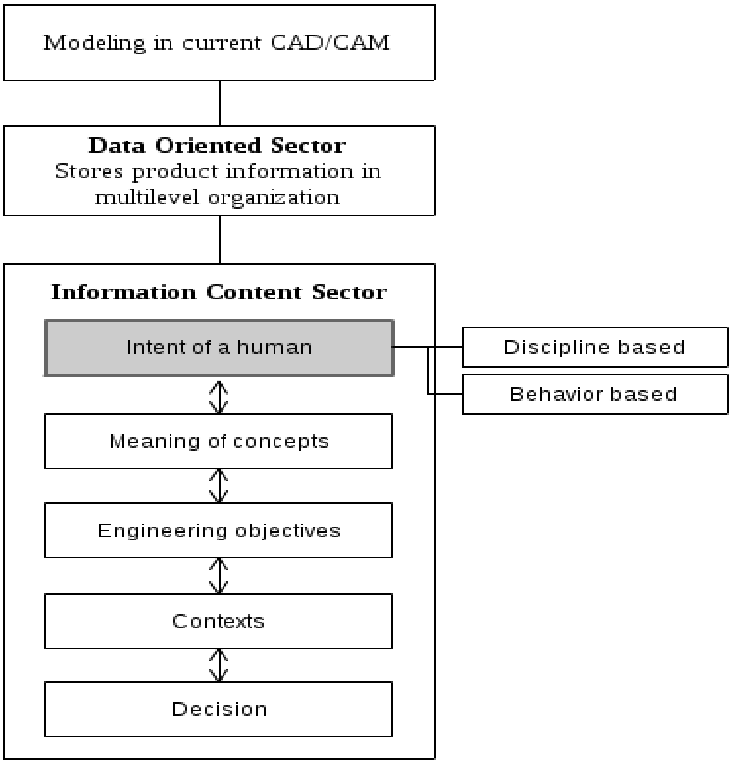


Figure 4.1: Category of content in the Information Content (IC) sector

The entity called Info-Chunk is introduced in the *Functional layer* and *Logical layer* of the RFLP structure for behavioral modeling of the multidisciplinary product. This entity is mapped with the IC to control the activities of the multidisciplinary product model. Further, Community diagrams [Y1] are used to visualize the parameters of the IC that assist with the engineering discipline and system behavior. The rest of the chapter is discussed as follows: IC is classified by engineering discipline and system behavior. After the classification, Info-Chunk is introduced in the RFLP structure for the conceptual models of the IC. Later, the configuration of Info-Chunk in the logical component and logical level is explained which drives the conceptual model. Finally, the conclusion is discussed.

4.2 Classification of Information Content

The RFLP structure product model is handled indirectly through the IC. The Conceptual models of IC are defined based on the Engineering discipline and System behavior. It is categorized as Discipline-based content and Behavior-based content as shown in Fig. 4.1. As the name indicates, Discipline based content stores the knowledge of engineering disciplines whereas, Behavior based content stores the knowledge of the system behavior. As shown in Fig. 4.2, RFLP structure is compliant with the IEEE 1220 standard. It is based on the V-cycle design process and allows concurrent engineering to coordinate the separate activities of a distributed design team. The conceptual models of IC are mapped with Logical and Physical levels of the RFLP structure. Both contents are interconnected so that any changes made in content affects the other. Furthermore, a different number of models can be constructed for a multidisciplinary product based on the classification. To explain the concept of the content, let as consider: the number of disciplines in Discipline based content, denoted by N_d. The number of behaviors in Behavior based content, denoted by N_b. The number of disciplines participated in the engineering activities, denoted by D. The number of expected behaviors of a system, denoted by B. In the case of the Behavior content, behaviors are categorized based on the priority. Some behaviors are important than others to implement a specific version of a multidisciplinary product.

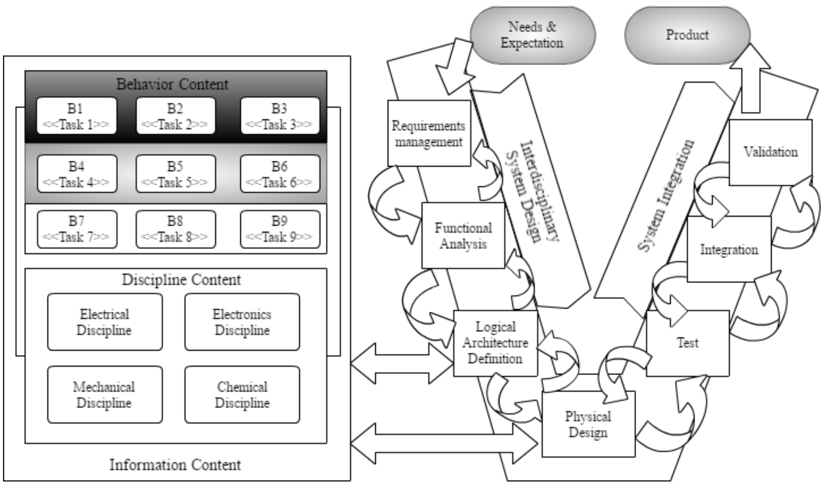


Figure 4.2: Interaction between the IC & RFLP structure

4.2.1 Discipline based Content

Discipline based content is defined by initializing the intent based on the number of participating engineering disciplines while modeling the product. This content is used to display the activity of discipline in the product model. According to the above mentioned nomenclature, N_d = D. The objective is to view the product in terms of various participating disciplines. Humans store the knowledge of the discipline in an Info-Chunk entity which will be discussed in the next section. This information is used to analyze a product.

4.2.2 Behavior based Content

The intent of behavior-based content is defined based on system behavior. The objective is to view the product in terms of the expected behaviors which is further based on the requirements. Priority [Y6] is assigned to an expected behavior in the content and arranged accordingly. According to the above mentioned nomenclature, $N_b \leq$ B, as some behaviors of a product has more priority than others.

4.3 Concepts of Info-Chunk

Info-Chunk is an entity that stores the correlated knowledge of *Functional level* and *Logical level* of the RFLP structure and organized the data-oriented sector. It is defined manually or automatically by the virtual space during the product design phase. The stored information is used by the IC for the behavioral modeling of the complex multidisciplinary product model. The Info-Chunk of the Logical layer of the RFLP structure is categorized as the Component Info-Chunk (CiC) and Logical Layer Info-Chunk (LiCL). CiC stores the knowledge of the logical component whereas LiCL stores the knowledge of the logical layer of the RFLP structure. The Info-Chunk of the Functional layer of the RFLP structure is categorized as the Sub-Function Info-Chunk (SFiC) and Functional Layer Info-Chunk (LiCF).SFiC stores the knowledge of the functional component whereas LiCF stores the knowledge of the functional layer of the RFLP structure.

4.3.0.1 Component Info-Chunk (CiC)

CiC is a low-level entity and store the information in the logical component. The block parameters of the CiC are described in the Fig. 4.3. It is used to store the knowledge of a component based on the configuration. According to the proposed rule, It is initialized either by an engineer or by the IVPS automatically. The *Component description* parameter is defined to store the component name and component number. It is an optional parameter. The *LiCL description* parameter stores the information of the Info-Chunk present on the logical layer of the RFLP structure. The *Community name* parameter stores the engineering discipline name of the component. It is the main parameter for the Discipline based content. The *Contribution in the logical component* parameter stores the role of CiC in the LiCL for the expected result. The *Connector* parameter stores the knowledge of the Inner connector and the Stream connector. Here, the Inner connector is concerned with the knowledge of input port and output port type while the Stream connector is concerned with knowledge of the material flow in the component as explained in [57].

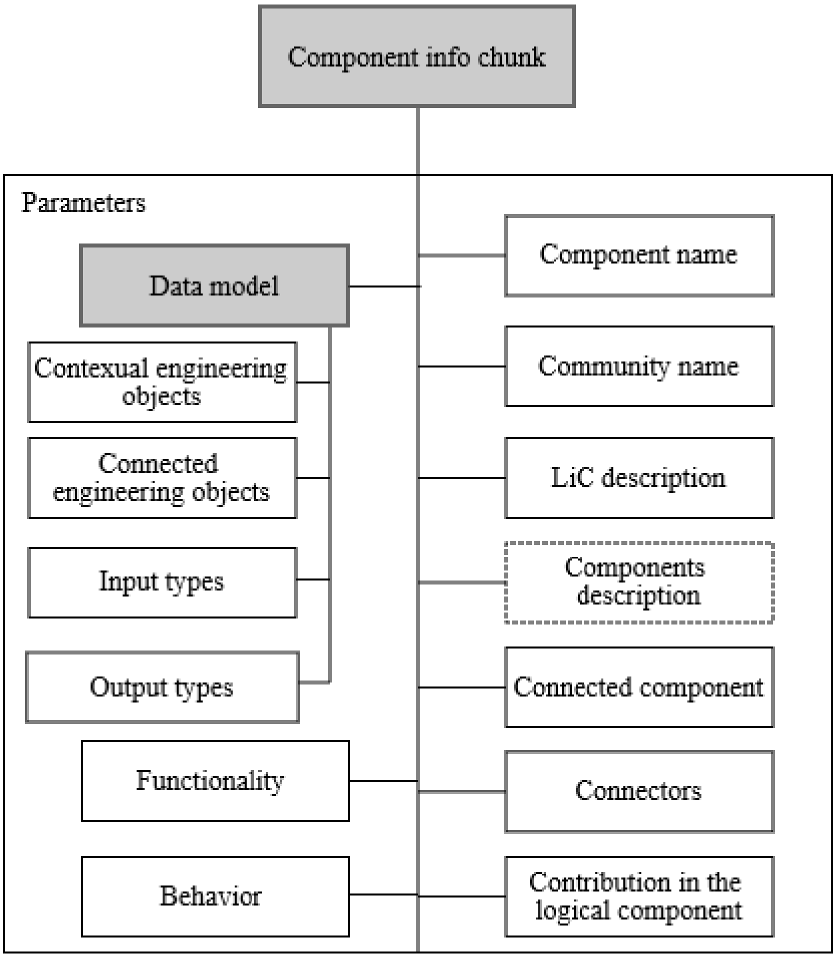


Figure 4.3: Parameter of Component Info-Chunk

The *Behavior* parameter stores the role of component behavior contributing to the behavior of the system. It is the main parameter for Behavior based content. The *Functionality* parameter stores the feature of the component. The *Data model* parameter stores knowledge of the contextual EOs for the physical level of a RFLP structure. The *Contextual engineering objects* parameter stores knowledge of the EO within the context of influenced EOs and relates to the component. The *Connected engineering objects* parameter stores the knowledge of EO in the context of connected EOs. The *Input types* and *Output type* parameters store the knowledge of connection type for input and output port. It depends on the discipline of the connected EOs for example if the connected engineering object at the input is a mechanical discipline then the input type parameter will be mechanical. Similarly, the output type parameter will be calculated.

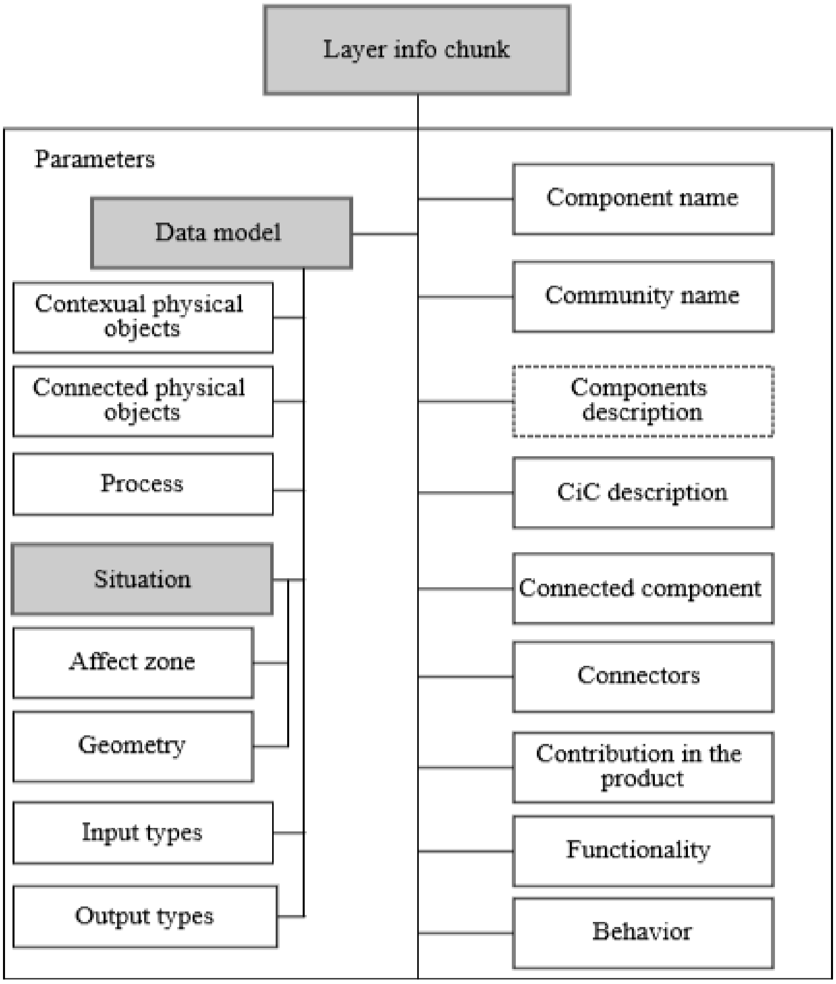


Figure 4.4: Parameter of Logical Layer Info-Chunk

4.3.0.2 Logical Layer Info Chunk (LiCL)

LiCL is a high-level entity and store the information of the logical layer component of the RFLP structure. The block parameters of the LiC are described in Fig. 4.4. The *Component name* parameter stores the name of the EO present in the logical layer of the RFLP structure. Like CiC, the parameters of LiCL such as community name, component description, functionality and behavior follow the same steps. The *Connector* parameter contains the knowledge of the Inner connector and Extended connector. The Inner connector parameter stores information of the input port and output port type while the Extended connector parameter stores information of input and output type of LiCL. In the case of Discipline content, the LiCL connector type refers to a discipline while in the case of Behavior content, the LiCL connector type refers to behavior.
The *Contribution in the product* parameter describes the role of LiCL in the logical level of the RFLP structure to deliver the expected result. The *Data model* parameter stores a detailed description of the engineering object. The *Affect zone* parameter stores the information of the EOs that are influenced due to changes that took place in the analyzed the EO. The *Geometry* parameter stores knowledge of the EO shape in a situation. It is considered a low level entity that stores the information of elements like parts, assemblies, form features, and others in this category. The element present in the geometry entity is the lower level entity as shown in Fig. 4.5. The *Process* parameter stores the information of the process involved during the product modeling. As explained in the previous chapter, the entities of the Process plane consists of the Analysis process, Contextual process, and Optimization process respectively. The entities of the process plane are defined in the previous chapter. The *Input type* parameter and *Output type* parameter store the input and output connection information of the contextual EO of a multidisciplinary product.

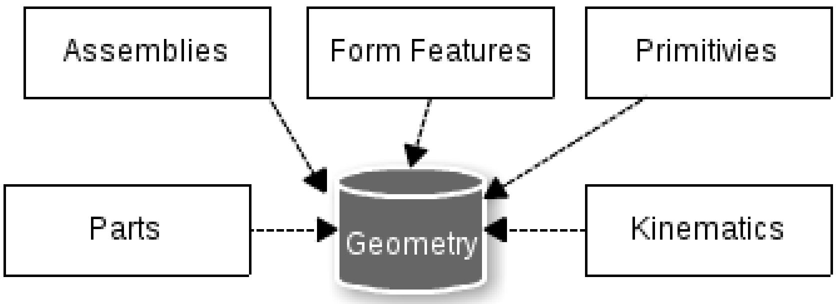


Figure 4.5: Elements in the Geometry entity

4.3.0.3 Sub-Function Info-Chunk (SFiC)

SFiC is a low-level entity and stores the knowledge of the sub function in the Functional component of the RFLP structure. It stores the information of individual elements that combine to form sub function or part function as shown in Fig. 4.6.

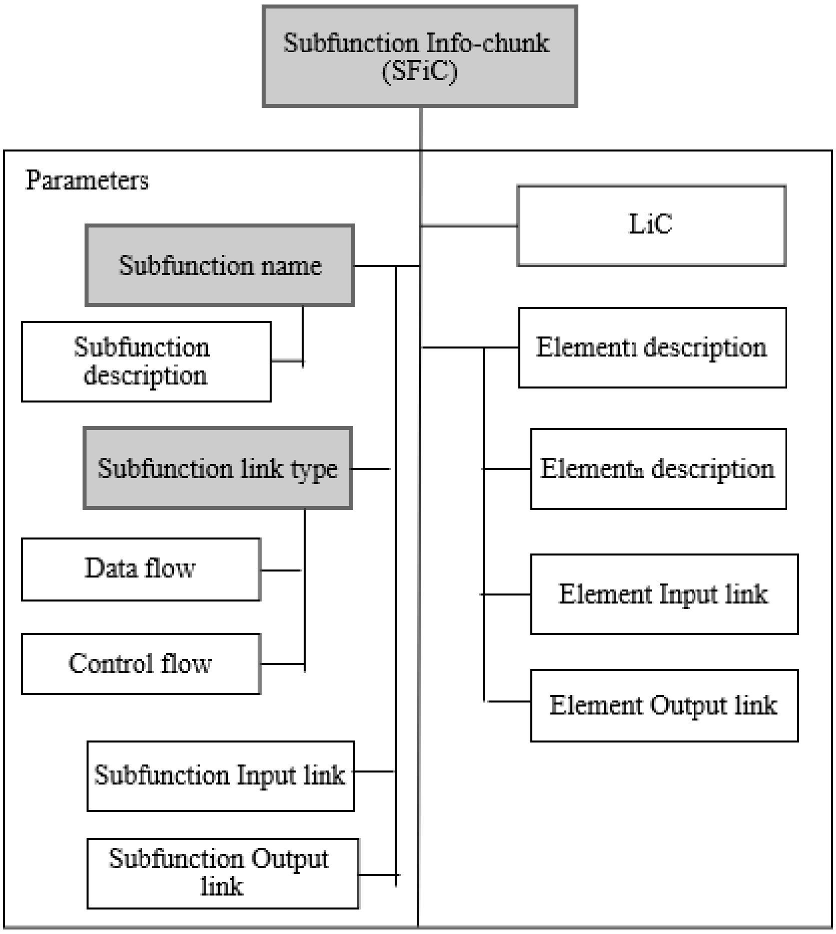


Figure 4.6: Parameters of Sub-Function Info-Chunk

The Subfunction name parameter stores the name and description of the sub-function. The link between individual sub-functions can comprise one of the two types: data flow or control flow [58]. Therefore, Subfunction type parameter store the flow information. The *Subfunction input link* parameter stores the information of SFiC connected to the input. The *Subfunction output link* parameter stores the information of SFiC connected to the output. The *LiC* parameter store the name of Functional layer Info-Chunk that consist of given SFiC. A Sub-function is a transaction broken into individual elements. The information of elements are stored in the *Element*$_1$, *Element*$_2$,...*Element*$_n$ *description* parameters where n is the number of elements. The *Element Input link* parameter and *Element output link* parameter stores the information of Input and output connection between elements respectively. Unlike CiC, only one SFiC is define for a sub function.

4.3.0.4 Functional Layer Info-Chunk (LiCF)

LiCF is a high-level entity and stores the information of the functional level of the RFLP structure. It stores the data of the main function which is the collection of SFiC and their connection as shown in Fig. 4.7.

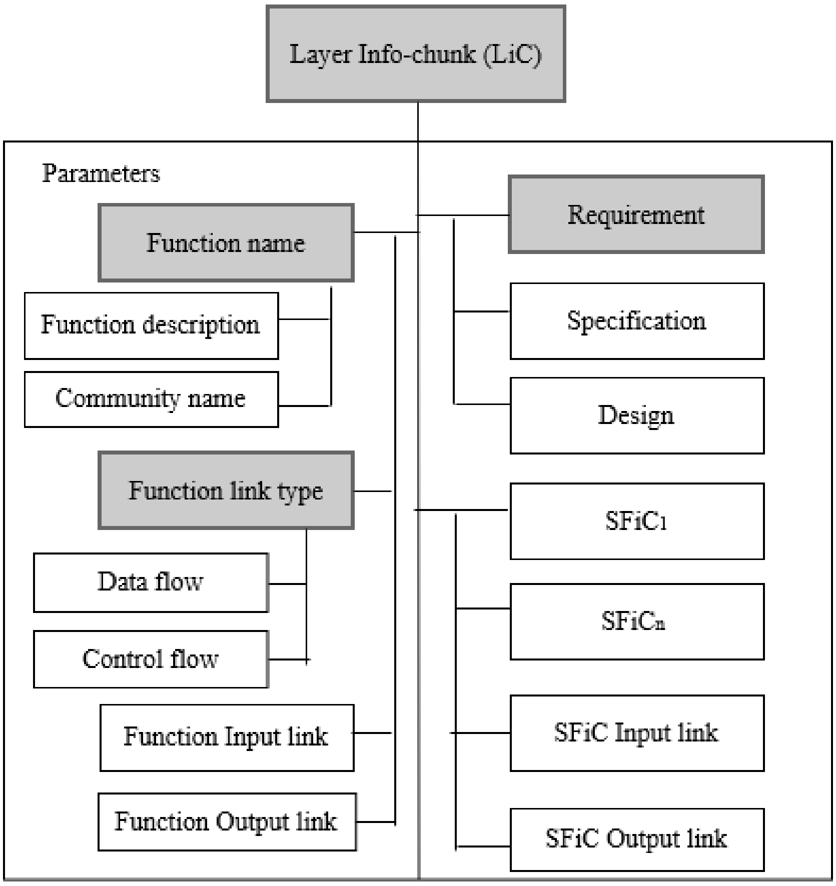


Figure 4.7: Functional Layer Info-Chunk

The *Function name* parameter stores the name, description and community name of the function. The *Function type* parameter store the flow information. The *Function input* parameter stores the information of LiCF connected to the input. The *Function output* parameter stores the information of LiCF connected to the output. As functions are derived from requirements in terms of specification and design. Therefore, *Requirement* parameter store the requirement information focus on the given LiCF. The main function is derived from the collection of sub-function. The information of sub-function are stored in the $SFiC_1$, $SFiC_2$,...$SFiC_m$ parameters where m is the number of sub function. The Input and output connection between SFiC are stored in the *SFiC Input link* parameter and *SFiC Output link* parameter respectively.

4.4 Configuration of Info-Chunk to drive the Multidisciplinary Product Model

The information of a LiCF is accessed through the LiCL. LiCF consists a collection of the SFiC and connected in the random fashion. LiCL consists a collection of the CiC and connected by the logical and physical connection in a RFLP structure. The configuration of the LiCL is shown in the Fig. 4.8a. The logical connection is the connection between the logical components and is demonstrated by a straight line whereas the physical connection is the connection between the physical components and is demonstrated by a dashed line.

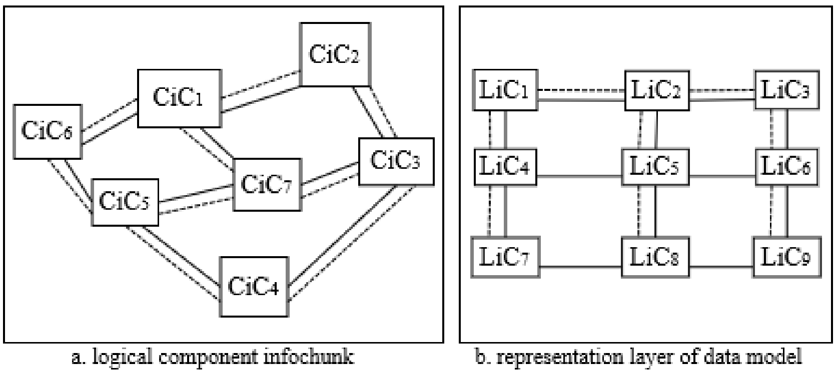


Figure 4.8: Configuration of Info-Chunk

The information in the LiCL depends on the type of IC. To access the information of a multidisciplinary product, LiCL are arranged in the representation layer of the data-oriented sector by physical connection and logical connection as shown in Fig. 4.8b. The CiC is placed in the logical component whereas the LiCL is placed at the logical level of the RFLP structure. For example, in the engineering system like CATIA V6, The CiC is a low-level entity that extracts information and is represented either corresponding to a Modelica component (MC) or to a group of Modelica components or the entire graph of a logical component as shown in the Fig. 4.9a. The human initializes the Info-Chunk entity as per the system specification. Like CiC, the LiCL is represented in the logical component (LC) by same steps as shown in the Fig. 4.9b. The logical level of the RFLP structure is mapped to the data-oriented sector by the LiCL that transfers the product related knowledge to the IC. The data-oriented sector is connected with the IC to take the correlating decisions during the product modeling.

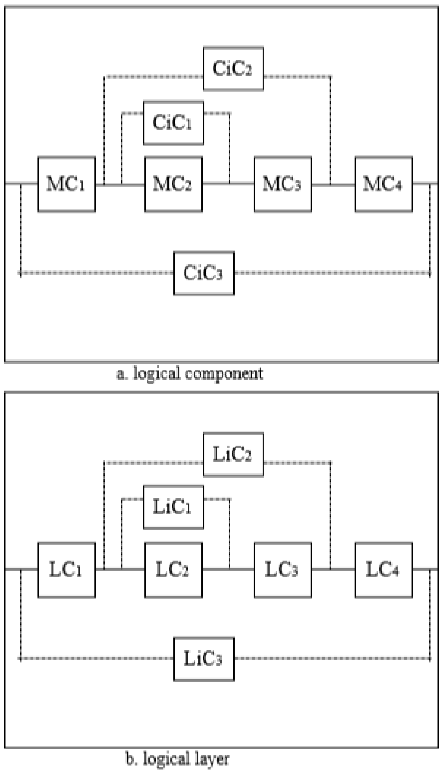


Figure 4.9: Representation of Info-Chunk

4.5 Conceptual Models of Information Content

After evaluating the parameters of Info-Chunk entities, the next step is to construct conceptual models of IC based on the System behavior and Engineering discipline. All the product related decisions take place at the decision level of the IC. It is important to note that data is accessible through the IC. In other words, it is not possible to make any changes in the Data-oriented sector directly. Therefore, the interface for the HCI is required to access the conceptual models.

4.5.1 Behavior Content

Behavior content is concerned with System behavior. It is focused on the customer demands. Priority is assigned to every behavior of the modeled product based on the requirements of the current and future versions during the multidisciplinary product modeling. The expected behavior of a product is evaluated in the IC by extracting the mapped information from the parameter of CiC,LiCL, SFiC, and LiCF respectively. Either human can initialize the parameters manually or Intelligent Virtual Product Space (IVPS) defines automatically the proposed parameters to modeled the behaviors of a multidisciplinary product model. The decision is visualized through the Community diagram in the *representation plane* of the IC. There are four scenarios to use the Community diagrams as shown in Fig. 4.10.

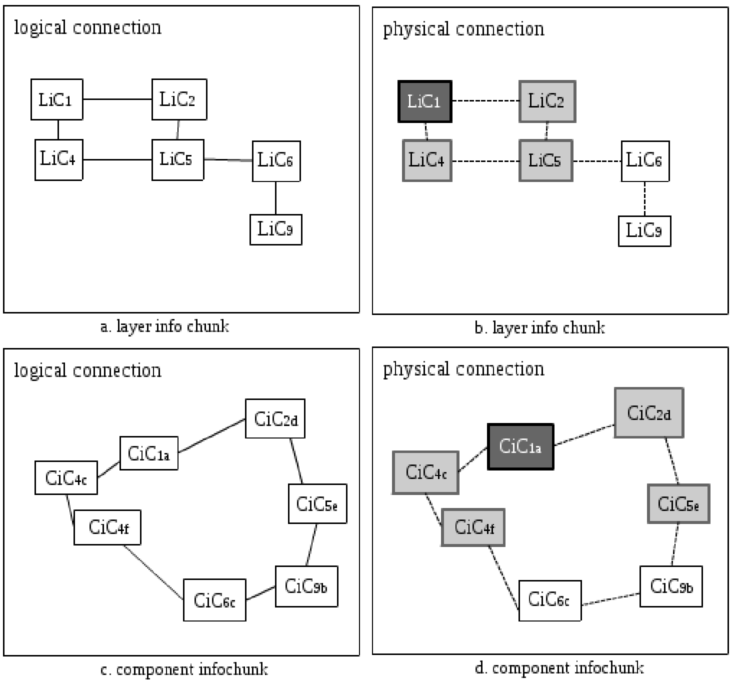


Figure 4.10: Community diagrams of the Behavior content

The logical and physical connection between the LiCL is explained in the Fig. 4.10a & Fig. 4.10b. Also, the logical and physical connection between the CiC is explained in Fig. 4.10c & Fig. 4.10d. To show the connections between the Info-Chunk entities using the Community diagram, let us assume the following nomenclature: Layer Info-Chunks are represented by $\{LiC_1, LiC_2, LiC_3, ..LiC_m\}$, where m is the total number. Similarly, Component Info-Chunks are represented by $CiC_x = \{CiC_{x1}, CiC_{x2}, CiC_{x3}, ..CiC_{xn}\}$ where x is the number of Layer Info-Chunk that consists of a specific set of Component Info-Chunk and n are the total number of the component in the Info-Chunk. A more detailed description is demonstrated in Fig. 4.10d, where information of the Component Info-Chunk is extracted which correlates with the Layer Info-Chunk. It is important that the nomenclature of LiCL and CiC are correlated. To analyze the information of a behavior, Info-Chunk is filtered by community names. For example, if any changes occurred in the LiC_1, it can influence the engineering objects of LiC_2, LiC_4 and LiC_5 as shown in the Fig. 4.10b. Based on the information obtained from the community diagram, the contribution of behavior is evaluated during the multidisciplinary product modeling. Later, the multidisciplinary product model is updated by initializing behaviors in the content or updating the existing behavior of the content.

4.5.2 Discipline-based Content

To define an engineering discipline in the Discipline content, CiC,LiCL, SFiC, and LiCF are defined. The approach is similar to Behavior content. The only difference is there is no priority set for the Discipline. The Community diagrams are used to extract the relevant information of discipline by using the Info-Chunk entities. Later, the multidisciplinary product model is updated by initializing disciplines in the content or updating the existing discipline of a content.

4.6 Summary

This chapter defines behavioral modeling of the multidisciplinary product by proposing the Info-Chunk entities. Here, LiCL stores the information of the logical level, CiC stores the information of the logical component, LiCF stores the information of the functional level & SFiC stores the information of a sub-function in the RFLP structure. Info-Chunk entities are used to describe the parameters for the conceptual model of IC. The classification is done based on System behavior and Engineering discipline. The final step is mapping between LiCL with the Information Content to take the correlating

decisions. The main purpose of the conceptual model is to store and represent the information of the complex multidisciplinary product model into a simplified form so that the human can more effectively analyze aspects of the multidisciplinary product. The models are used to guide the human to take the precise correlating decision of the complex product model. For the practical approach, Behavioral modeling of the multidisciplinary product using active content will be explained in the consecutive chapters.
This chapter covers the Thesis Group 2.

Thesis Group 2

Thesis group 2: Info-Chunk driven Information Content for the Multidisciplinary Product Modeling

Thesis 2.1
I have introduced the entities called Info-Chunk [Y3] in the Functional and Logical layer of the Requirement Functional Logical Physical (RFLP) structure for the behavioral modeling of the multidisciplinary product. In the paper [38], Behavior based models with intelligent content were emphasized, where feature models for specification and knowledge representations were conceptualized. It is based on the classical product modeling and has limited knowledge to simulate the behavior of the modeled objects. Whereas, this research work proposes the behavior models for the multidisciplinary product with Behavior content and Discipline content based on the Info-Chunk Entities. They can simulate the behavior of the modeled objects based on the information stored on the Functional and Logical layer of the RFLP structure. The conceptual models are used to guide the engineer for the simplified representation of the multidisciplinary product to take the correlating decisions.

Relevant own publications about this thesis group: [Y3].

Chapter 5

Behaviors Representation of the Multidisciplinary Product

This chapter proposes the Info-Chunk objects in the Multilevel Abstraction based Self-Adaptive Definition (MAAD) Structure and Requirement Functional Logical Physical (RFLP) Structure for behaviors representation of the multidisciplinary product where the Info-Chunk entity is considered as an object. Further, the concepts of Info-Chunk objects are extended to Intelligent Property (IP) that uses IBCA structure to handle the RFLP structure for the self adaptive multidisciplinary product modeling. The chapter is structured as follows:
Section 5.1: Introduction.
Section 5.2: Info-Chunk as an Object.
Section 5.3: Behavior Storing Techniques using Info-Chunk objects.
Section 5.4: Rules for the Generation of Info-Chunk Objects.
Section 5.5: Comparison with existing Behavioral Modeling Techniques.
Section 5.6: Summary the chapter.

5.1 Introduction

As mentioned in the previous chapters, Community zones are used in the IC to organize the product model entities and their relationship. Further, behaviors of the modeled entities are evaluated in the IC by the Process plane. Also, the Info-Chunk entity is introduced in the Functional and Logical layer of the RFLP structure for Behavioral modeling of a multidisciplinary product. Further, there is need of behavior representations and structured processing of interrelated Engineering Objects (EOs) to obtain the coordinated decisions. To solve the above-mentioned issues, this chapter proposes the Info-Chunk objects in the RFLP structure and MAAD structure for the behaviors representation of multidisciplinary product. Info-Chunk objects are based on the Object Oriented Principle (OOP) concepts that are used in software programming. Here, the Info-Chunk entity is converted into the Info-Chunk object first. Then, Behavior Info-Chunk (BiC) object and Context Info-Chunk (CxiC) object are introduced in the MAAD structure and the IBCA structure to store the behaviors of the multidisciplinary product. The proposed Info-Chunk objects are used to establish a link with the LiCL and LiCF objects of the RFLP structure. The multidisciplinary product model could be handled efficiently through the IC instead of the Specification tree of the RFLP structure. This chapter begins with the conversion of the Info-Chunk entity into the object. Further, BiC objects and CxiC objects are proposed in the MAAD structure and the IBCA structure. Then, behavior storing techniques for the multidisciplinary product are explained with the aforementioned concepts. Finally, rules for generating the BiC objects and CxiC objects in the MAAD structure are defined by using the pseudo-codes.

5.2 Info-Chunk as an Object

This chapter proposes Object Oriented Principle (OOP) concepts in the multidisciplinary product modeling. Encapsulation, inheritance, and polymorphism are the three principles of OOP methodology. This work starts with the entities and their relationship. Info-Chunk is an entity defined in the RFLP structure. However, OOP concepts are not directly applicable to an entity. The Info-Chunk entity must be converted into the Info-Chunk object for communication between IC and RFLP structure. Based on the entity-object conversion process by Ou Y. [59] and Bernhard Thalheim [60] :

- The parameters of an Info-Chunk entity is equivalent to the attribute of an Info-Chunk object
- Entity Relationship (ER) between Info-Chunk is equivalent to the Object Relationship (OR). Here, the method of an Info-Chunk object is derived from the OR as per the requirement of a specific discipline

Then, BiC object and CxiC object are proposed in the MAAD structure and IBCA structure. According to the proposed concept of Info-Chunk objects:

- In the RFLP structure, LiCL object consist of the attributes and methods of the Logical level and LiCF objects consist of the attributes and methods of Functional level

- In the MAAD structure, BiC object consist of the attributes and methods of Behaviors level and CxiC objects consist of attribute and method of Contexts level

- In the IBCA structure, BiC objects consist of the attributes and methods of *Situation defining behaviors (SB)* level of Behavior substructures and CxiC objects consist of attribute and method of *Product definition Activity Contexts (AC)* level, *Adaptive Drive Contexts (DC)* level, *Product Feature Contexts (FC)* level of Contexts substructures

5.3 Behavior Storing Techniques Using Info-Chunk Objects

Behavior is based on well-defined situations for sets of circumstances. It is represented in the Functional level and Logical level of the RFLP structure. BiC objects and CxiC objects represent dynamic behavior information. They are stored in the MAAD structure and IBCA structure to communicate with the LiCL objects and LiCF objects of the RFLP structure. IC operates the RFLP structure by the MAAD structure. Also, IP operates the RFLP structure by IBCA structure. The behavior storing techniques are classified as the operation performed by the BiC objects and CxiC objects in the MAAD structure and IBCA structure.

5.3.1 Info-Chunk Objects based MAAD Structure

The Behavior level of the MAAD structure drives the Functional level and Logical level of the RFLP structure. In the MAAD structure, a behavior is represented at *Behaviors* level and *Contexts* level. For behavior representation, communication between the RFLP structure and the MAAD structure is done by using the proposed BiC objects and CxiC objects. The BiC objects communicate with the LiCL objects as shown in Fig. 5.1, where the main contextual connections of the MAAD structure are organized as follow:

- The solid line is the inside contexts (C) of Behaviors levels for the MAAD structure. It is explained in the paper [38], where the contextual connection of model entities in the MAAD level is defined.

- The bold line is the driving contexts (D) of Behaviors levels for the MAAD structure. It drives the Functional level and Logical level of the RFLP structure. The dashed lines are the information retrieved by the BiC objects from the LiCL objects of the Functional level and Logical level.

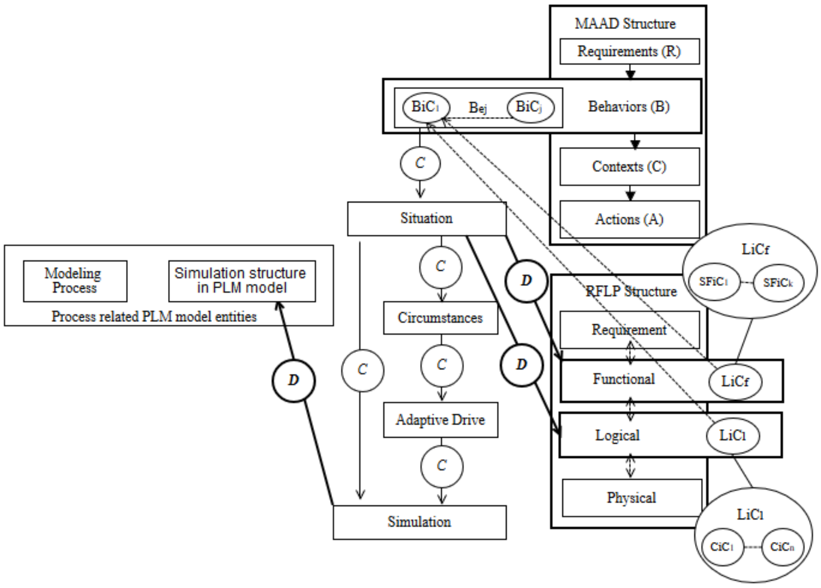


Figure 5.1: Communication between MAAD and RFLP structure at Behaviors level

In the case of the Logical layer of RFLP structure, it retrieves the *situation* attribute of the LiCL object $\{LiCL_1, LiCL_2, LiCL_3, ...LiCL_o\}$ and corresponding *behavior* attribute of their CiC objects $\{CiC_1, CiC_2, CiC_3, ..CiC_n\}$. It is represented inside the oval shape in the diagram. The information retrieved by the driving contexts populates the BiC objects in the *Behaviors* level of MAAD structure. Here, n is the number of CiC objects in a LiCL object and o is the total number of LiCL objects in the logical layer. The information retrieved is the actual situation, circumstances for the situation and the adaptive drive to drive context definitions. In the case of Functional layer of RFLP structure, driving contexts (D) retrieves the *Requirement class* attribute of the LiCF object $\{LiCF_1, LiCF_2, LiCF_3, ..LiCF_l\}$ and corresponding *Elements description* attributes of the SFiC objects $\{SFiC_1, SFiC_2, SFiC_3, ..SFiC_k\}$. It is represented inside the oval shape in the diagram. The information retrieved by the driving contexts populates the BiC objects in the *Behaviors* level of MAAD structure. Here, k is the number of SFiC objects in a LiCF object and l is the number of LiCF objects in the Functional layer of RFLP structure. The retrieved BiC objects are represented as $\{BiC_1, BiC_2, BiC_3, ..BiC_j\}$. Here, j is the number of BiC objects in the Behaviors substructure. The CxiC objects communicate with the LiCL objects is shown in Fig. 5.2, where the main contextual connections of the MAAD structure is organized as follows:

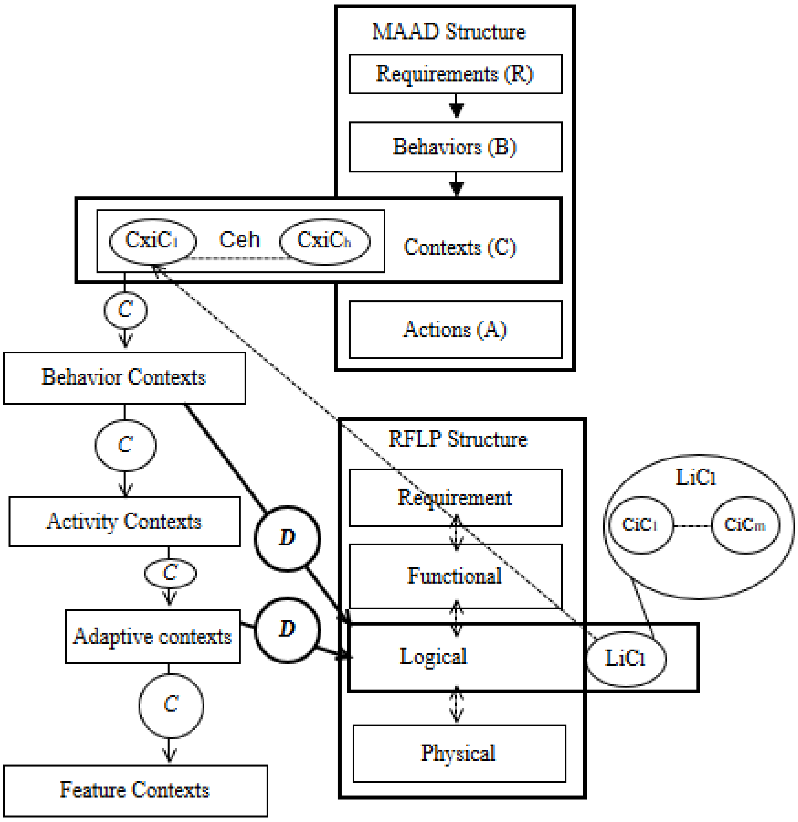


Figure 5.2: Communication between RFLP and MAAD structure at Contexts level

- The solid line is the inside contexts (C) of *Contexts* levels for the MAAD structure. It is explained in the paper [38], where the contextual connection of model entities in the MAAD level is defined.
- The bold line is the driving contexts (D) of *Behaviors* levels for the MAAD structure. It drives the Logical level of the RFLP structure. The dashed line is the information retrieved by the CxiC objects from the LiCL objects of the Logical level.

In the case of the Logical layer of RFLP structure, it retrieves the data model attribute of the LiCL object $\{LiCL_1, LiCL_2, LiCL_3, ...LiCL_o\}$ and corresponding data model attribute of CiC objects $\{CiC_1, CiC_2, CiC_3, ..CiC_m\}$.

Here, m is the number of CiC objects in a LiCL object and o is the total number of LiCL objects in the Logical layer. The information retrieved is the concept behavior, activity, adaptive and product feature contexts, connection behavior definitions, model definition activities, contexts for an adaptive drive, and context for physical level product and knowledge features. The retrieved CxiC objects are represented as $\{CxiC_1, CxiC_2, CxiC_3, ..CxiC_h\}$, where, h is the number of CxiC objects in the Contexts substructure.

5.3.2 Info-Chunk Objects based IBCA Structure

The driving generation of the RFLP element is done by the IP. Human-initiated engineering activities with the company IP by using IBCA structure for the generation of RFLP elements. It leads to the analysis of self-adaptive PLM modeling. The Info-Chunk objects based IBCA structure drives the RFLP structure as shown in in Fig. 5.3. The solid lines are the interaction between the IBCA structure and RFLP structure. The dashed lines are the information retrieved by the BiC objects and CxiC objects from the LiCF objects of the Functional level and LiCL objects of the Logical level. On the Behavior (B) level of the IBCA structure, SB substructure are configured to define behaviors by a set of BiC objects.

- In the Logical level of the RFLP structure, the *situation* attribute & *behavior* attribute of the LiCL object $\{LiCL_1, LiCL_2, LiCL_3, ...LiCL_d\}$ and the corresponding *behavior* attribute of the CiC objects $\{CiC_1, CiC_2, CiC_3, ..CiC_a\}$ are stored in the BiC objects of the SB element. Here, a is the number of CiC objects in a LiCL object and d is the total number of LiCL objects.

- In the Functional level of the RFLP structure, *Requirement* attribute of the LiCF object $\{LiCF_1, LiCF_2, LiCF_3, ...LiCF_c\}$ and corresponding *Elements description* attributes of the SFiC objects $\{SFiC_1, SFiC_2, SFiC_3, ...SFiC_b\}$ are stored in the BiC objects of the SB element. Here, b is the number of SFiC objects in a LiCF object and c is the number of LiCF objects in the functional level.

The stored information in the BiC objects is behavior definition (IEBD) and the related situation (IEBT) [47]. The total BiC objects obtained from the LiCF objects of the functional layer and LiCL objects of the logical layer is represented as $\{BiC_1, BiC_2, BiC_3, ...BiC_n\}$. Here, j is the number of BiC objects in the SB element. On the Contexts (C) level of the IBCA structure, product definition AC level, adaptive DC level, and product FC level are configured to define behaviors by a set of CxiC objects.

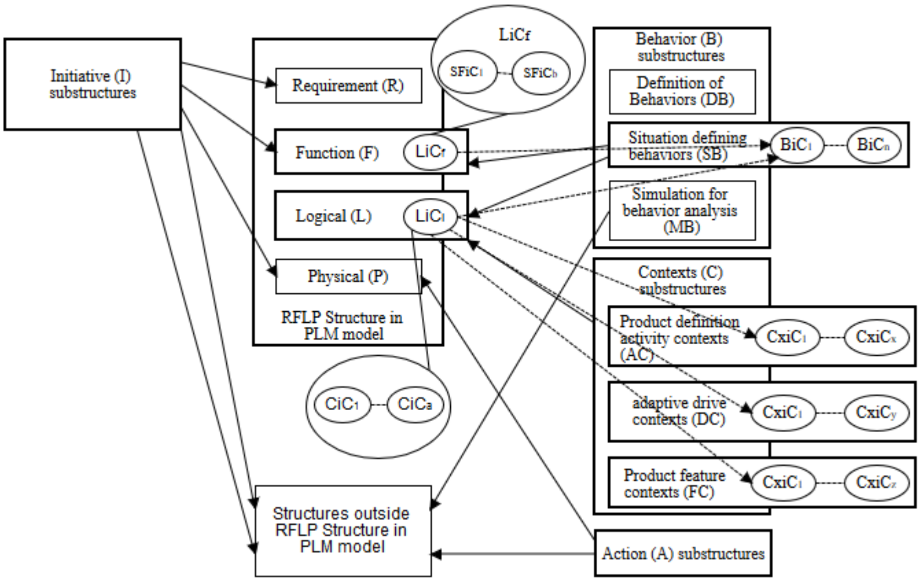


Figure 5.3: Communication between RFLP and MAAD structure at Contexts level

In the Logical level of the RFLP structure, the *Data model* attributes of LiCL objects $\{LiCL_1, LiCL_2, LiCL_3, ...LiCL_d\}$ & CiC objects $\{CiC_1, CiC_2, CiC_3, ..CiC_a\}$ are stored by the CxiC objects of AC, DC and FC elements. Here, a is the number of CiC objects in a LiCL object and d is the total number of LiCL objects. The stored information in the CxiC objects is the product behavior (IECB). The total CxiC objects obtained from the LiCL objects and represented as $\{CxiC_1, CxiC_2, CxiC_3, ..CxiC_x\}$, $\{CxiC_1, CxiC_2, CxiC_3, ..CxiC_y\}$, $\{CxiC_1, CxiC_2, CxiC_3, ..CxiC_z\}$. Here, x, y, z are the number of CxiC objects stored in the AC, DC and FC elements.

5.4 Rules for the Generation of Info-Chunk Objects

Behavior models with intelligent content involve specifications and knowledge for the design processes. The most appropriate forms of knowledge are formulas, rules, and checks. In the following sections, this work focuses on Info-Chunk object activities in the IC. Here, the MAAD structure is the driving factor for representing the behavior of the RFLP structure. Rules are the set of instructions that can be executed for generating and storing the Info-Chunk objects in the MAAD structure and IBCA structure. Rules are defined by using the pseudo-codes.

- In the case of the MAAD structure, the behavior objects $\{BiC_1, BiC_2, BiC_3, ...BiC_j\}$ are stored in the *behaviors* level and the context objects $\{CxiC_1, CxiC_2, CxiC_3, ..CxiC_h\}$ are stored in the *contexts* level. The Process plane of IC can elaborate on the BiC objects and CxiC objects for the behavior representation of the multidisciplinary product. After the analysis process, the analyzed objects are stored with the nomenclature of BiC_ab. If a human wants to evaluate the context of an analyzed object on the other analyzed object, the context object undergoes the effect process. The resultant objects are stored as $CxiC_ec$. Further, If a human wants to optimize the contextual object, it is stored as BiC_ob after the optimization process. It is also possible to optimize the behavior of an object without analysis. IC retrieve and store required objects at the *Engineering objectives* level to drive the behavior of RFLP structure.

- In the case of the IBCA structure, the behavior objects $\{BiC_1, BiC_2, BiC_3, ...BiC_n\}$ are stored in the behavior substructure and the context objects $\{CxiC_1, CxiC_2, CxiC_3, ..CxiC_x\}$, $\{CxiC_1, CxiC_2, CxiC_3, ... CxiC_y\}$, $\{CxiC_1, CxiC_2, CxiC_3, ..CxiC_z\}$ are stored in the contexts substructures. IP retrieve and store the objects to drive the behavior of the RFLP structure. The IP level and process plane of IP are not defined yet. The behavior representation for IP is the topic of future work.

Pseudo Codes for BiC & CxiC objects

- BEGIN LOOP
- **Initialize** a Process
- IF 'Process'is 'Analysis'
  - BEGIN LOOP
  - Store 'BiC_{ab}'in 'Behaviors level'where 1 $\leq$ ab$\leq$ j
  - IF 'Process'is 'Contextual'
    - BEGIN LOOP
    - Store '$CxiC_{ac}$'in 'Contexts level'where 1 $\leq$ ac$\leq$ h
    - IF 'Process'is 'Optimization'
      - BEGIN LOOP

        · Store 'BiC_{ob}'in 'Behaviors level'where 1 ≤ ob≤ j
        · END LOOP
      * END LOOP
  – END LOOP

- IF 'Process'is 'Optimization'
  – BEGIN LOOP
  – store 'BiC_{ob}'in 'Behaviors level'where 1 ≤ ob≤ j
  – END LOOP

5.5 Summary

The authors proposed a behaviors representation of multidisciplinary product using the Info-Chunk driven RFLP structure product model where Info-Chunks are used as objects. The LiCL and LiCF objects of the RFLP element are populated and retrieved by the BiC and CxiC objects of the MAAD structure. The proposed model describes the techniques to store information about the behavior of modeled objects in different circumstances. Automatic, reactive Info-Chunk based propagation of any change of the RFLP element at any stage of the modeling process makes the design consistent with intents, goals, and decisions. Info-Chunk objects provide necessary specifications and knowledge representations to simulate the behavior of the modeled objects. For practical approach, InfoChunkLib Application Programming Interface (API) will be discussed in the next chapter.
This chapter covers the Thesis Group 2.

Thesis Group 2

Thesis group 2: Info-Chunk driven Information Content for the Multidisciplinary Product Modeling

Thesis 2.2
I have introduced the Info-Chunk objects [Y4] in the Behaviors and Contexts layer of the Multilevel Abstraction based Self-Adaptive Definition (MAAD) structure for behavior representation of the multidisciplinary product. In the paper [45], the average behavior of the input-output signals of the switched reluctance generator has been reproduced required for system level analysis of the aircraft power distribution system. Considering it as a base, this research work proposes the representation of the behavior of a multidisciplinary product, which is based on the analysis, contextual connection, and optimization activities. The rules and logic is defined as per the Process plane of the Information Content.

Relevant own publications about this thesis group: [Y4].

Chapter 6

InfoChunkLib API and Content Web Server

This chapter focuses on the practical approach for the representation of the behavior of the multidisciplinary product. Therefore, An Application Programming Interface (API) called “InfoChunkLib” is proposed to handle by the Information Content (IC) to represent the modeled behavior information of a multidisciplinary product. Hence, a Content Web server is proposed, where IC web application is used to represent the modeled behavior data and zone information of the multidisciplinary product model. Then, the Content database is created to store the LiCL and LiCF entities information of the multidisciplinary product model. The chapter is structured as follows:
Section 6.1: Introduction
Section 6.2: Engineering Objects Using the OOP Concepts
Section 6.3: Practical Approach of the Community Zone and Process Plane
Section 6.4: InfoChunkLib API
Section 6.5: Content Web Server
Section 6.6: Summary the chapter.

6.1 Introduction

The InfoChunklib API, IC web application and Content web server are proposed in the Information Content (IC) to represent and store the modeled behaviors of the multidisciplinary product. The API is based on the communication between the MAAD structure and the RFLP structure. The API is coded by using the Java language as a JavaFX application. IC web application imports the InfoChunkLib API and coded it as a Web application. The retrieval of modeled behaviors information is according to the process plane of the IC. The zone and extracted modeled behavior data of a multidisciplinary product are displayed by the IC web application. The application is coded using the Java Servlets, Java Server Pages (JSP), Java Server Pages Standard Tag Library (JSTL), Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) & Java Database Connectivity (JDBC). The Apache HTTP Server is used as a Content Server and hosts the IC web application with a PostgreSQL database that is used to store the modeled behavior data from the Process plane. Here, RESTful web service to exchange the information between the Content and CAD system web server. This chapter starts with the class diagram of the Engineering Objects categorized in the previous chapter. Then practical approach of the Community zone and Process plane are described. Next, InfoChunkLib API is explained for the representation of the behavior of the multidisciplinary product. Then, the Content server is explained where the operations and Content database are emphasized. Finally, communication between the Content and CAD system web server is elaborated.

6.2 Engineering Objects Using the Object-Oriented Principles

As per the OOP concept, an EO in a product model is considered as a class and represented using the Class diagram as shown in Fig. 6.1.

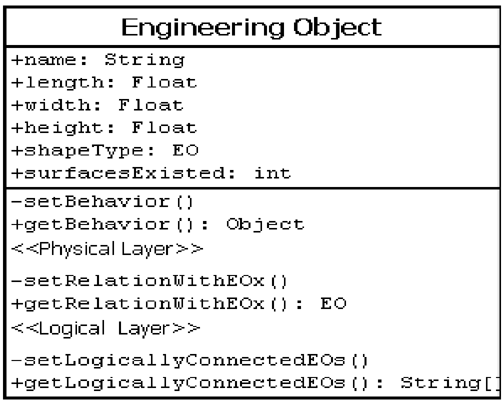


Figure 6.1: Class Diagram of a *EO*

Then, the Class diagram of a Single zone EO is shown in Fig. 6.2. Here, there are three associate connections and one dependency of the given EO EO_a.

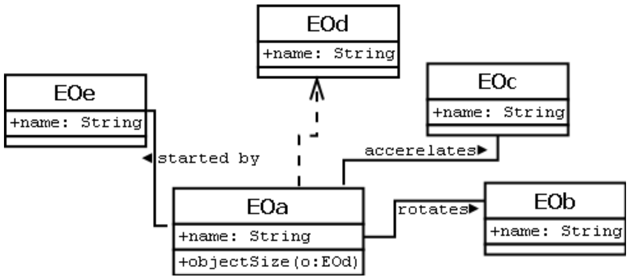


Figure 6.2: Class Diagram of Single Zone *EO*

Similarly, the Class diagram of a Multiple zone EO is shown in Fig. 6.3. Through the UML diagram, the Product model is aggregated by the Community zone, which is further aggregated by the EO as shown in Fig. 6.4.

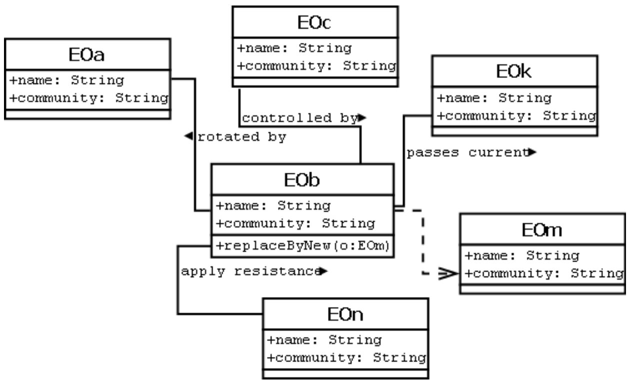


Figure 6.3: Class Diagram of Multiple Zone *EO*

Figure 6.4: UML Diagram for aggregation

6.3 Practical Approach of the Community Zone and Process Plane

The Community zones are initialized in the *Engineering Objectives* layer of the IC and the zones are represented by the Community diagrams in the *Representation plane* of the IC. As shown in the Fig. 6.5, Community zones organize the complex product model entities and their relationship based on the disciplines.

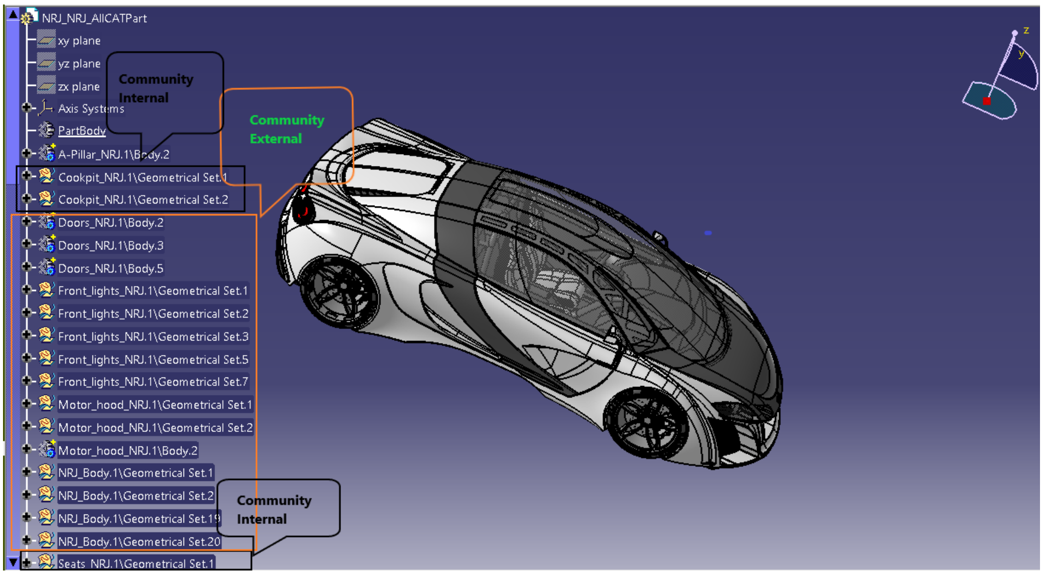


Figure 6.5: Community zone concept in a complex Product Model

Then, the Process plane is done by investigating the product behaviors of a multidisciplinary product model. The behaviors of modeled entities are analyzed in the IC by the process plane. The analysis is performed with the Multiphysics analysis. The multiphysics problem is inevitably nonlinear, but most of the solution is obtained by solving a series of linear sub-problems. The formulation of a multiphysics problem is highly dependent on the nature of the problem and how the component of different disciplines are coupled in space and time. The results of the analysis are categorized as *Analysis process* and stored in the *Engineering objectives* layer of the IC. It is also possible to evaluate the analysis of an engineering object or discipline in the context of others. The results are categorized as *Contextual process* and stored in the *contexts* layer of the IC. Further, analyzed data are optimized by nonlinear programming. It is done as per the requirement of product type and constraints that depend on the system performance, environmental conditions, contextual connection dependency, etc. Basic approaches used to optimize the nonlinear problem are to find local optima and global optima respectively. The results are categorized as *Optimization process* and stored in the *Decisions* layer of the Information Content. In the case of multidisciplinary product modeling, it is precisely achieved by finding global optima rather than discipline-specific optima [56]. It also depends on the required accuracy level. The level of accuracy depends on the engineering objectives. If absolute accuracy is required, then it is necessary to calculate global optima otherwise local optima is sufficient. In the case of global optima, absolute accuracy is calculated at the cost of efficiency. As the name indicates, the *Decisions layer* of the IC decides the optimized data of the product model and stores the best possible solutions in the form of the *Optimization process*.

6.4 InfoChunkLib Application Programming Interface (API)

The relationship between the abstraction levels of the MAAD structure is described by using the UML diagram as shown in Fig. 6.6. In the OOP, a UML diagram is used to define the relationship and model the behavior of the product.

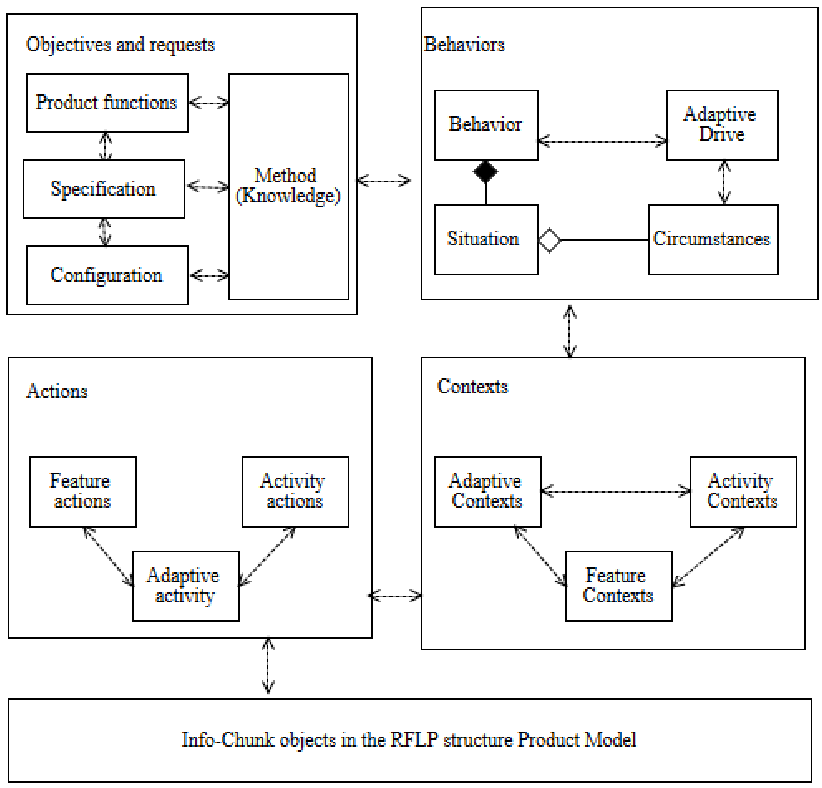
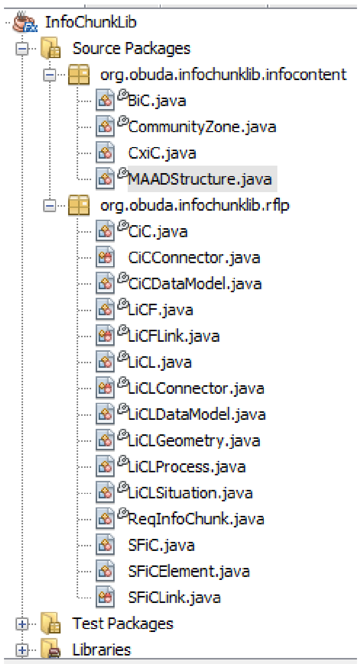


Figure 6.6: UML representation of MAAD structure with the Info-Chunk based RFLP Structure

Here, ER modeling of the MAAD structure is converted into OR modeling. As per the concepts, the relationship between objects is defined by the composition, aggregation, and association. Hence, there is a bi-directional association relationship between the *Objectives and Requests* level, *Behaviors* level, *Contexts* level, and *Actions* level. Inside the *Behaviors* level, the behavior object has a composite relationship with the situation object, which further has an aggregation relationship with the circumstances object. Also, the behavior object has a bi-directional association with the Adaptive Drive object. The *InfoChunkLib* API is coded in the JavaFX application as shown in Fig. 6.7. It consists of two Java packages. The *informationcontent* Package consists of all the classes related to the IC like *MAADStructure* class, *BiC* class, *CxiC* class, and *Community-Zone* class. The *rflp* Package consists of all the classes related to the RFLP structure like *LiCL* class, *LiCF* class, *CiC* class, and *SFiC* class.

Figure 6.7: InfoChunkLib API



6.4.1 Info-Chunk Objects in the RFLP Structure

To explain the proposed concepts in the system behavior, let us consider a car as an example. As per the community concepts, a car system is the combination of various communities where the *Electrical supply system* is one of the community. It consists of components like battery, starter, alternator, heater, fan, distributor, etc. Here, battery and alternator components are used for the Info-Chunk objects concept explanation. Then, the following scenario is considered as an example: The oil consumption of car depends on the engine behavior that must be modeled according to the situation such as the experience of the driver with the sets of circumstances like path traveled by car, the condition of the car, surrounding environment, etc. Here, the dynamic behavior of the engine strongly is influenced by the situation and weakly influenced by the circumstances. The parameters of the LiCL objects, CiC objects, LiCF objects and SFiC objects are described in the previous chapter. The descriptions of LiCF class and LiCL class are shown in the code below. The LiCF class is written to show the layer Info-Chunk object of the Functional layer in the RFLP structure. Here, the array of SFiC objects, ReqInfoChunk object, LiCFLink object and String parameters are used as an argument in the constructor. The constructor arguments of the ReqInfoChunk object are populated from the Requirement layer of the RFLP structure. The ReqInfoChunk class is written to show the customer requirements in the requirement layer of the RFLP structure. The SFiC class is written to show the sub-functional Info-Chunk object in the Functional layer of the RFLP structure. The LiCFLink is the enumeration class to store connector information. The concepts of constructor overloading are used so that LiCF can accept various sets of the argument depends on the initialization of the object. Further, the LiCL class is written to show the layer Info-Chunk object of the logical layer in the RFLP structure. Here, the array of CiC objects, LiCLGeometry object, LiCLSituation object, LiCLProcess object, LiCLDataModel object, LiCF object, LiCLConnector object, integer parameter, boolean parameter, string parameters, and the array of string parameters are used as an argument in the constructor. The constructor arguments of a LiCF object are populated from the Functional layer of the RFLP structure.

The concepts of constructor overloading are used so that LiCL accepts various sets of the argument depends on the initialization of the object. The CiC class is written to show the component Info-Chunk object in the logical layer of the RFLP structure. The LiCLGeometry class is written to show the geometry of the multidisciplinary product. Here, it could be possible for data retrieval of the product model and part model STEP files in the LiCL class. In that case, LiCLGeometry constructor arguments $part_info$ and $assembly_info$ are converted from string types into the STEP file format. Here, JSDAI API [61] could be the possible approach to read and write the STEP file format. Then, LiCLGeometry object, affect zone and array of circumstances are used as a constructor argument for the LiCLSituation object. The LiCLSituation class is written to show the situation with a set of circumstances applicable to a LiCL object. The LiCLProcess class is written to show the process plane of the IC. It accepts String and Boolean values of processes as a constructor argument. The string values store the name of the processes whereas Boolean value stores the status of a process. The LiCLConnector is the enumeration class to store connector information. The get method returns the value of objects required to the main application. It is used in the next subsection. The code of LiCF and LiCL classes are demonstrated below:

```java
// LiCF.java
/**This class is written to show the layer Info-Chunk object of
   the functional layer in the \acrshort{rflp} structure
*@param func_name This parameter stores the name of a function
```

```java
*@param func_descrip This parameter stores the description of a function
*@param comm_name This parameter stores the community name of a function
*@param func_input This parameter stores the inputs to a LiCF object
*@param func_output This parameter stores the outputs from a LiCF object
*@param arrySFiC This parameter stores the array of the \acrshort{sfic} objects
*@param req This parameter initializes the \acrfull{rfic} object
*@param funct_link This parameter stores links between the two functions of LiCFLink type. It could be Data flow or Control flow
*/
package org.obuda.infochunklib.rflp;
public class LiCF {
 String func_name, func_descrip, comm_name, func_input, func_output; SFiC[] arrySFiC = null; ReqInfoChunk req = null; LiCFLink func_link;
/**
*This constructor is used to initialize the \acrshort{licf} objects without information from the requirement layer
*@param name_func This parameter defines the name of a function
*@param descrip_func This parameter defines the description of a function
*@param name_comm This parameter stores the community name of a function
*@param link This parameter stores links between the two functions of LiCFLink type. It could be Data flow or Control flow
*@param input_func This parameter stores the inputs to a \acrshort{licf} object
*@param output_func This parameter stores the outputs from a \acrshort{licf} object
*@param subArry This parameter initializes the array of the \acrshort{sfic} objects
*/
     public LiCF (String name_func, String descrip_func, String name_comm, LiCFLink link, String input_func, String output_func, SFiC[] subArry) {
func_name = name_func; func_descrip = descrip_func; comm_name = name_comm; func_link = link; func_input = input_func; func_output = output_func; arrySFiC = subArry;}
/**
*This constructor is used to initialize the \acrshort{licf} objects with the information from the requirement layer
*@param spec_LiC This parameter stores the specification of a \acrshort{licf} object
```

```
*@param design_LiC This parameter stores the design of a \
    acrshort{licf} object
*/
     public LiCF (String name_func, String descrip_func, String
        name_comm, LiCFLink link, String input_func, String
        output_func, SFiC[] subArry, String spec_LiC, String
        design_LiC) {func_name = name_func; func_descrip =
        descrip_func; comm_name = name_comm; func_link = link;
        func_input = input_func; output_func = func_output;
        arrySFiC = subArry; req = new ReqInfoChunk (spec_LiC,
        design_LiC);
     }}

// LiCL.java
/**This class is written to show the layer Info-Chunk object of
    the logical layer in the \acrshort{rflp} structure
*@param comp_name This parameter stores the name of a component
*@param community_name This parameter stores the community name
    of a component
*@param descrp_CiC This parameter stores the description of a
    component
*@param contrib_product This parameter stores the contribution
    of the component in the product modeling
*@param type_output This parameter stores the outputs from the \
    acrshort{licl} object
*@param type_input This parameter stores the inputs to the \
    acrshort{licl} object
*@param comp_connected This parameter stores the number of
    connected components in a \acrshort{licl} object
*@param connect This parameter stores information of connector
    type. It could be Inner connector or Extended connector
*@param components This parameter stores the array of the CiC(
    Component InfoChunk) objects
*@param functionality This parameter stores the feature of a \
    acrshort{licl} object with \acrshort{licf} type
*@param gmtry This parameter stores the geometry of the
    components in a \acrshort{licl} object
*@param situation This parameter stores the information of
    influenced components and geometry of the components in a \
    acrshort{licl} object at the given situation
*@param process This parameter stores the process involved in a
    \acrshort{licl} object
*@param data_model This parameter stores the detail description
    of a \acrshort{licl} object in the context of the physical
    object such as process, geometry, and situation
*/
package org.obuda.infochunklib.rflp;
public class LiCL {
private String comp_name, community_name, descrp_CiC,
```

```
    contrib_product , type_input , type_output ; int comp_connected ;
     LiCLConnector connect ; CiC [] components = null ; LiCF
    functionality = null ; LiCLGeometry gmtry = null ;
    LiCLSituation situation = null ; LiCLProcess process = null ;
    LiCLDataModel data_model = null ;
/**
*This constructor is used to initialize the LiCL object without
    the information to the physical layer
*@param name_comp This parameter stores the name of a component
*@param name_community This parameter stores the community name
    of a component
*@param connected_comp This parameter stores the number of
    connected components in a LiCL object
*@param product_contib This parameter stores the contribution of
     the component in the product modeling
*@param input_type This parameter stores the inputs to the LiCL
    object
*@param output_type This parameter stores the outputs from the
    LiCL object
*@param affect_zone This parameter stores the influenced
    components during the analysis in a LiCL object
*@param part_info This parameter stores part information in the
    geometry of a LiCL object
*@param assembly_info This parameter stores assembly information
     in the geometry of a LiCL object
*@param form_features This parameter stores form feature
    information in the geometry of a LiCL object
*@param circum This parameter stores the array of the
    circumstance of a situation in a LiCL object
*@param arryCiC This parameter stores the array of the CiC(
    Component InfoChunk) objects
*@param function This parameter stores the feature of a LiCL
    object with LiCF type
*/
     public LiCL(String name_comp, String name_community, int
        connected_comp , String product_contib , String input_type ,
         String output_type , String affect_zone , String part_info
        , String assembly_info , String form_features , String []
        circum , CiC [] arryCiC , LiCF function ) {
comp_name = name_comp ; comp_connected = connected_comp ;
    community_name = name_community ; contrib_product =
    product_contib ; type_input = input_type ; type_output =
    output_type ; functionality = function ; connect =connection ;
    components = arryCiC ; gmtry = new LiCLGeometry( part_info ,
    assembly_info , form_features ) ; situation = new LiCLSituation(
    affect_zone , circum , gmtry ) ;
     }
/**
*This constructor is used to initialize the LiCL object with the
```

```
    information to the physical layer
*@param process_analysis This parameter stores the status of the
    analysis process in a LiCL object
*@param process_effect This parameter stores the status of the
   effect/contextual process in a LiCL object
*@param process_optimization This parameter stores the status of
    the optimization process in a LiCL object
*@param value_analysis This parameter stores the array of
   analysis process values in a LiCL object
*@param value_effect This parameter stores the array of
   contextual process values in a LiCL object
*@param value_optimization This parameter stores the array of
   optimization process values in a LiCL object
*@param connection This parameter stores information of
   connector type. It could be Inner connector or Extended
   connector
*@param contexual_PO This parameter stores knowledge of
   contextual Physical object/s in a LiCL object
*@param connected_PO This parameter stores knowledge of
   connected Physical object/s in a LiCL object
*/
    public LiCL(String name_comp, String name_community, int
       connected_comp, String product_contib, String input_type,
        String output_type, String affect_zone, String part_info
       , String assembly_info, String form_features, String[]
       circum, Boolean process_analysis, Boolean process_effect,
        Boolean process_optimization, String[] value_analysis,
       String[] value_effect, String[] value_optimization,
       LiCLConnector connection, String contexual_PO, String
       connected_PO, CiC[] arryCiC, LiCF function) {
comp_name=name_comp; community_name=name_community;
   comp_connected = connected_comp; contrib_product =
   product_contib; type_input = input_type; type_output =
   output_type; functionality = function; connect =connection;
   components = arryCiC;
gmtry = new LiCLGeometry(part_info, assembly_info, form_features
   );
situation = new LiCLSituation(affect_zone, circum, gmtry);
   process = new LiCLProcess(process_analysis, process_effect,
   process_optimization, value_analysis, value_effect,
   value_optimization); data_model = new LiCLDataModel(
   contexual_PO,process,situation,connected_PO, type_input,
   type_output);
    }

    private String getSituation() {
    return situation.affect_zone;}
    private String [] getCircumstances() {
    return situation.circumtnces ; }
```

```
private CiC[] array_Components(){
return components;}
private LiCF getLiCF(){
return functionality;}
private LiCLProcess getProcessInfo(){
return process;}}
```

6.4.2 Info-Chunk Objects in the MAAD Structure

The BiC and CxiC class are the application classes for the behavior representation of the multidisciplinary product model as shown in the code below. The output is the graph between the components of various disciplines. It is the outcome of the process plane of the IC. The BiC class accepts the LiCL and LiCF objects as a constructor argument. Using the LiCL object, the LiCLProcess object can check the status of the analysis and optimization process. If the value is true, then it can generate the graph related to the process. The outcome of the Analysis process is shown in Fig. 6.8.

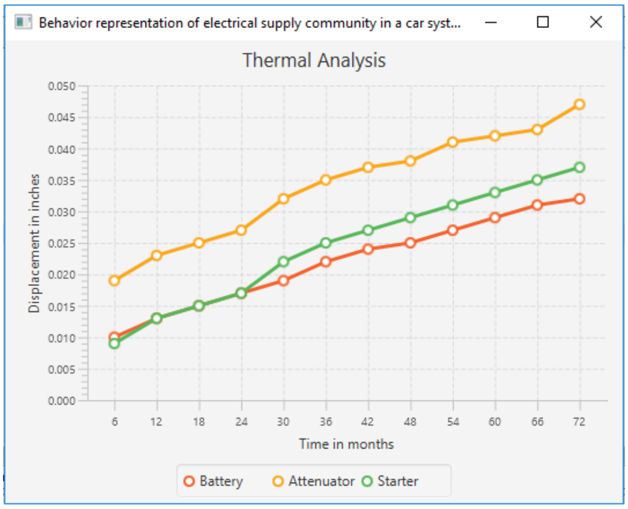


Figure 6.8: The graph of components after thermal analysis

The graph explains the displacement of the battery, starter and attenuator components w.r.t to time after the Thermal Analysis process.

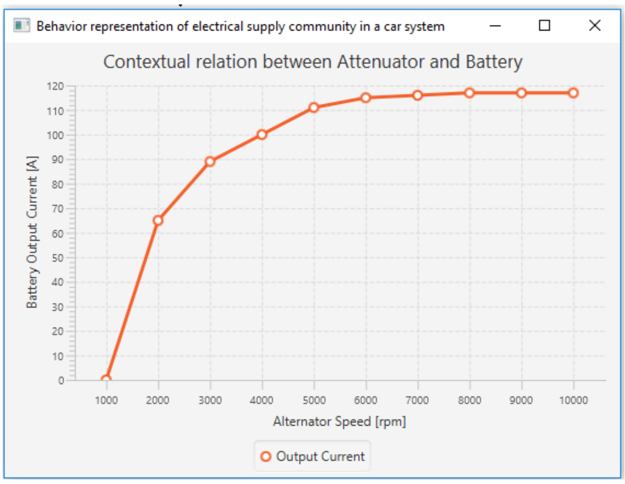


Figure 6.9: The graph between battery and attenuator component after contextual process

The CxiC class accepts LiCL object as a constructor argument. Using LiCL object, the LiCLProcess object can check the status of the contextual process. If the value is true, then it can generate the graph based on the contextual relationship between engineering objects. The outcome of Effect process is shown in Fig. 6.9. where contextual relation between attenuator and battery is explained by varying the battery output current with the attenuator speed.

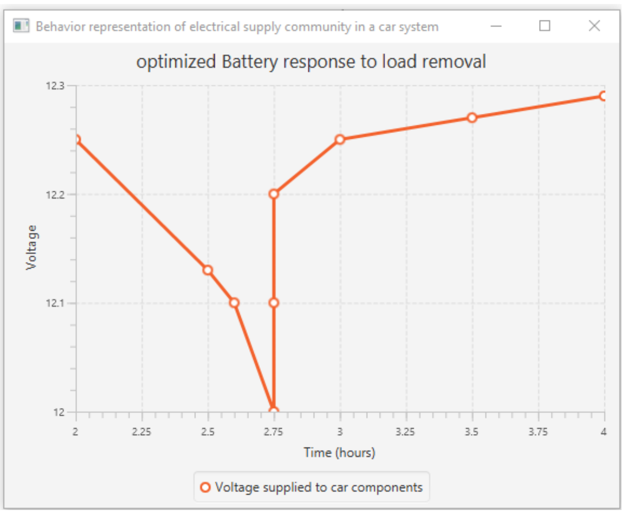


Figure 6.10: The graph of the battery component after the optimization process

Here, XYChart class is used for generating the Line Chart graph. The outcome of the optimization process is shown in Fig. 6.10. The graph explains the voltage required w.r.t time for the optimized battery response. The MAADStructure class is the main method class that launches the application by calling the objects of BiC and CxiC objects. The code of BiC, CxiC and MAADStructure classes are demonstrated below:

```
//BiC Class
/**
This class is written to show the behavior Info-Chunk object of
    the Behaviors layer in the \acrshort{maad} structure
*@param bfunc This parameter initializes the LiCF (Layer
    InfoChunk in the functional layer) object.
*@param blogic This parameter initializes the LiCL (Layer
    InfoChunk in the logical layer) object.*/
public class BiC extends Application {
   LiCF bfunc = null;
   LiCL blogic = null;
/**
*This is the only constructor used to initialize the BiC (
    Behavior InfoChunk) object with the information of LiCF and
    LiCL object*/
    public BiC(LiCF funct, LiCL logic) {
    funct = bfunc;  logic = blogic; }

/*This method is used to demonstrate the graph obtained from the
     analysis and optimization process. The parameters could be
    the components, parts or expected changes in the assembly. */
  @Override
 public void start(Stage stage) {
if(blogic.getProcessInfo().isAnalysisProcess()){
//generate graph
}
if(blogic.getProcessInfo().isAnalysisProcess()){
//generate graph
}
if(blogic.getProcessInfo().isOptimizationProcess()){
//generate graph
}}

//CxiC Class
/**
This class is written to show the Contexts Info-Chunk object of
    the Behaviors layer in the \acrshort{maad} structure
*@param blogic This parameter initializes the \acrshort{licl}
    object */
public class CxiC extends Application {
   LiCL blogic = null;
```

```
/**
*This is the only constructor used to initialize the CxiC (
   Contextual InfoChunk) object with the LiCL object*/
  public CxiC(LiCL logic) {logic = blogic;}

/**This method is used to generate the graph obtained from the
   contextual process. The parameters could be the components,
   parts or expected changes in the assembly. */
    @Override
    public void start (Stage stage) {
if(blogic.getProcessInfo().isEffectProcess()){
//generate graph}
}}

//MAADStructure Class
/** This class is written to show the main application of the
   InfoChunkLib API. */
public class MAADStructure {

 /** This method is used to start the main application by
    calling the BiC and CxiC objects and generate the graphs*/
    public static void main(String args[]) throws Exception {
    Application.launch(BiC.class, args);
    new Thread(){
    Application.launch(CxiC.class, args);
    }}   }
```

6.4.3 Implementation of InfoChunkLib API in the Information Content

InformationContent class is the application which imports the InfoChunkLib API and handles the multidisciplinary product model. It could be a Java application or Web application. The output is stored in the database. As shown in the code below, SFiC objects and CiC objects are initialized first. Then, LiCF objects are initialized from SFiC objects and LiCL objects are initialized from CiC objects and LiCF objects. Then, BiC objects are initialized by LiCF objects and LiCL objects. CxiC objects are initialized by LiCL objects. Finally, the MAADStructure class is called by InformationContent arguments and graphs are generated for the behavior representation of multidisciplinary product model. The code of InformationContent class is demonstrated below:

```
//InformationContent.java
```

```
/**This class is written to use InfoChunkLib API to drive the
    multidisciplinary product model*/
import org.obuda.infochunklib.rflp.SFiC;
import org.obuda.infochunklib.rflp.SFiCLink;
import org.obuda.infochunklib.rflp.CiC;
import org.obuda.infochunklib.rflp. ConnectorCiC;
import org.obuda.infochunklib.rflp.LiCF;
import org.obuda.infochunklib.rflp. FunctionLink;
import org.obuda.infochunklib.rflp.LiCL;
import org.obuda.infochunklib.rflp.ConnectorLiC;
public class InformationContent{
public static void main(String args[]) throws Exception{
//Extract SFiC object arguments information from the functional
    layer and physical layer (.step file)
SFiC subfunc1 = new SFiC("To recharge the battery", "Energize a
    field current that turns a rotor inside a set of stators that
     can produce high current in alternating directions",
    SubfunctionLink.DataFlow, "Mechanical energy", "Electrical
    energy", "The electrical system of a car is a closed circuit
    with an independent power source the battery");

//Extract CiC object arguments information from the logical
    layer and physical layer (.step file)}}
CiC comp1 = new CiC("Alternator", "Electrical supply", "large
    BATT terminal connected to battery positive, Relay Terminal
    connected to the connect to the dash warning light, Sense
    Terminal connect the pigtail directly to the BATT terminal",
    "provide power to the car electrical system", subfunc1, "
    Magnet movement", "Energy", "Battery", "Engine and Starter",
    ConnectorCiC.Inner);
//Extract LiCF object arguments information from the functional
    layer and physical layer (.step file)
LiCF funct = new LiCF("To power the car system", "The battery
    provides juice to the starter. Then, the alternator gives
    that battery the energy required to power the car system", "
    Electrical Group", FunctionLink.DataFlow, null, "Power",
    arrySFiC);
//Extract LiCL object arguments information from the logical
    layer
LiCL logic = new LiCL("Electrical supply", "Electrical Group",
    3, "To supply power to Car system", "Mechanical Energy", "
    Power", "Experienced_driver","Alternator, Starter and Battery
    ", assembly_info, null, true, false, circumstnce, true, false
    , true, value_analysis_thermal, null,
    value_optimization_global, ConnectorLiC.Extended, "Lighting
    and signaling system", "Ignition electronic system", arryCiC,
     func); }}

//Initialize BiC object and CxiC object from LiCL object and
```

```
    LiCF object
BiC behav = new BiC(funct, logic);
CxiC context = new CxiC(logic);

//Call MAADStructure class for behavior representation of
    multidisciplinary product model
String[] args = new String[0];
MAADStructure.main(args);}
```

6.4.4 Testing Phase of the InfoChunkLib API

It is necessary to check the stored information in the Info-Chunk objects. In the OOP based language like Java, JUnit testing is a popular tool to check the behavior of an object. The behavior of a multidisciplinary product can be tested by varying the attributes and methods of the BiC and CxiC objects in the virtual environment. These values are compared with the values obtained from the physical environment. Further, formulas can be derived from the consistent values obtained from the virtual and physical environment.

6.5 Content Web Server

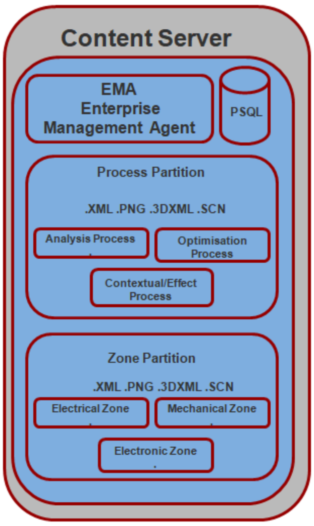


Figure 6.11: Content Server

Content Web server is the Apache Http Server that used to store and display the data of the Information Content (IC) as shown in Fig. 6.11. Tomcat Servlet is used for the IC web application. Enterprise Management Agent (EMA) is the integral software component responsible for managing and maintaining the IC based Web application. It also allows monitoring the CAD Product database, through management plug-ins and connectors. The Process partition consists of the outcome of the Process plane of the IC, the product model after a certain set of the process applied and the files that explain the location of outcomes of the Process plane and product model. Similarly, the Zone partition consists of the outcome of the community zone of the IC, the product model after divided into the zones and the files that explain the location of outcomes of the Community zone and product model. The outcomes are the graphs obtained from the Process plane and Community zones. The authors store the graphs in the PNG format, product model application in the Dassault Systém's 3DEXPERIENCE file format (3DXML), the XML file format for the data interchange and SCN file format for the 3D product model (Assembly model or part model) management.
The Content database is created by using PostgreSQL. It stores the data of the IC application while handling the behavior modeled data and zone information of the multidisciplinary product application. ER [62] diagram is used for the physical data modeling as shown in the Fig. 6.12. It is required for the schema level for creating a database. There are nine tables created based on the concept of LiC entities of the RFLP structure. During the product modeling using the RFLP structure, there is a set of information transferred from the Requirement layer to the Physical layer. Behaviors of a product model are represented in the Functional and Logical layer of the RFLP structure. LiCF table is used to store the attributes of the Functional layer and the LiCL table is used to store the attributes of the Logical layer of the LiC entity of the RFLP structure. In these tables, some of the data types are built-in while others are user-defined. In the case of the LiCL table,

- LiCLConnector is the Enumerated data type that stores the inner and connector values of the LiCL entity.
- LiCF, CiC, and LiCLDataModel are the composite data type, whose attributes and data types are specified in the Content ER diagram.
- In the CiC table, CiCConnector is the Enumerated data type that stores the inner and stream values of the CiC entity
- In the CiC table, SFiC and CiCDataModel are the composite data type, whose attributes and data types are specified in the Content ER diagram.

- In the LiCLDataModel table, LiCLSituation and LiCLProcess are the user-defined composite data types.

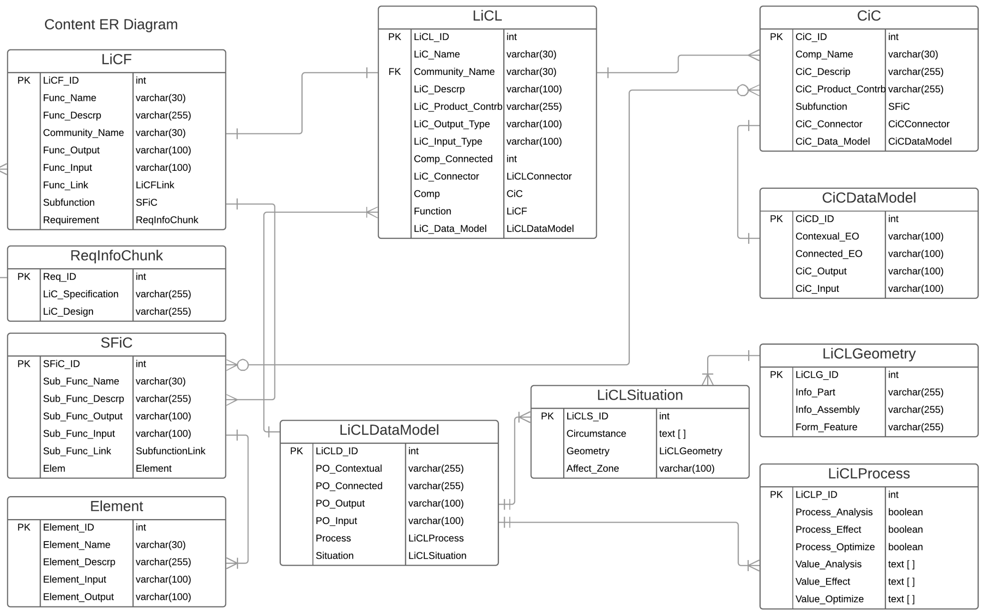


Figure 6.12: Content Entity Relationship Diagram

The CiC table is used to store the attributes of the CiC entity present in a LiCL entity. LiCLDataModel table is used to store the attributes of the detailed description of the Physical layer of the RFLP structure. LiCLProcess table is used to store the attributes of the Process plane of the IC. LiCLSituation table is used to store the attributes of a situation in the logical layer of the RFLP structure. Here, LiCLGeometry composite data type is used to store the information of a part model or assembly model in a situation. In the case of the LiCF table,

- LiCFLink is the Enumerated data type that stores the inner and connector values of the LiCF entity.

- SFiC and ReqInfoChunk are the composite data types, whose attributes and data types are specified in the Content ER diagram.

- In the SFiC table, SubFunctionLink is the Enumerated data type that stores the inner and stream values of the SFiC entity

- In the SFiC table, Element is the composite data type, whose attributes and data types are specified in the Content ER diagram.
- In the ReqInfochunk table, attributes and data types are specified in the Content ER diagram

For reference, LiCLConnector, LiCLDataModel, and LiCLProcess commands are demonstrated using the SQL statements of PostgreSQL as shown below. Here, new tables and data type is created using the CREATE statement.

```
CREATE TYPE LiCLConnector AS ENUM
('inner', 'Stream');

CREATE TYPE LiCLDataModel AS (
    LiCLD_ID INT,
    PO_Contextual VARCHAR(255),
    PO_Connected VARCHAR(255),
    PO_Output VARCHAR(100),
    PO_Input VARCHAR(100),
    Process LiCLProcess,
    Situation LiCLSituation);

CREATE TYPE LiCLProcess AS (
    LiCLP_ID INT,
    Process_Analysis BOOLEAN,
    Process_Effect BOOLEAN,
    Process_Optimize BOOLEAN,
    Value_Analysis TEXT[],
    Value_Effect TEXT[],
    Value_Optimize TEXT[]);
```

The modeled behavior data of a multidisciplinary product is stored in the entities based on the entities relationship. The entities are populated by the IC application.

- In the context of the Functional layer, One LiCF entity may have many SFiC entities and one or many LiCF entities may have one ReqInfoChunk entity. Further, one SFiC entity may have one or many Element entities.
- In the context of the Logical layer, One LiCL entity may have one LiCF entity and many CiC entities. Also, one or many LiCL entities may have one LiCLDataModel entity. Further, one and only one LiCLDataModel entity may have one or many LiCLProcess and LiCLSituation entities. Here, one or many LiCLSituation entities may

have one LiCLGeometry entity. Also, One CiC entity may have one CiCDataModel.

6.5.1 Operations

A human expert handles the multidisciplinary application through the IC application. To model the behavior data, the process plane from the *Engineering objectives* layer of the IC interacts with the Info-Chunk objects of the *Product Behaviors* level of the MAAD structure, which further, drives the Info-Chunk objects of the *Functional* and *Logical* layer of the RFLP structure. Here, the Process plane of the IC communicates with LiC entities of the RFLP structure using the Info-Chunk objects to retrieve the modeled behavior data of a multidisciplinary product plane. The data is stored in the Process partition. Also, the Product model is divided into community zone based on the discipline. The outcome is stored in the Zone partition. Then, the human can interact with the results stored in the partition through the representation plane of the Interactive IC application. The outcome could be static or dynamic and represented as graphs, images or animation.

6.5.2 Communication between Content Server and CAD Product Server

The CAD Server pulls process partition and zone partition from the Content server when replaying through the IC interface in the Multidisciplinary web application as shown in Fig. 6.13. Content server partitions information is saved in CAD server cache and auto-deleted almost immediately after the replay. EM-EMA Link handles the publishing of configuration between framework and Content server. RESTful Web API Link handles passing of modeled behavior data and zone partition details from Postgres job queue to Local Contact DB which then gets moved on to Central Contact DB by ETL SQL process of the CAD server. The advantage of this API there is no need to install additional software or libraries and provide a great deal of flexibility. Content Server handles the retrieval of.XML,PNG, .3DXML and .SCN content from the Content Server to the IC Webtop application interface of the Multidisciplinary application for replay. The process and zone partition details are taken from Central Contact DB and converted to .3DXML format for the multidisciplinary application and then it is deleted. Further, a web application is proposed on the webserver. The web application is created based on the concepts used by the author in the paper [Y9].

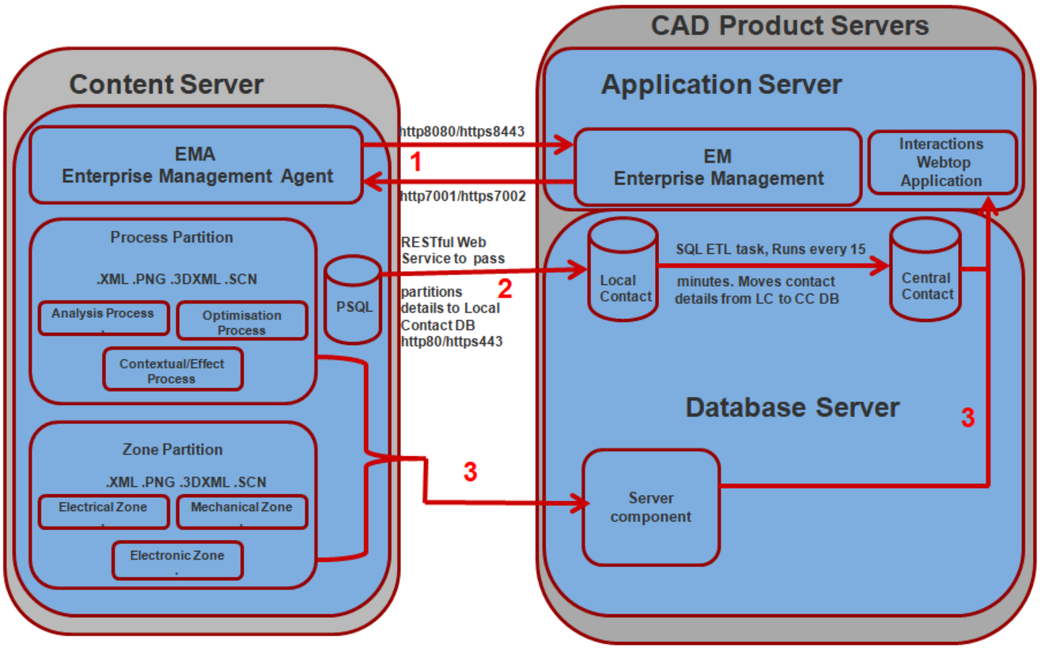


Figure 6.13: Communication between Content server and CAD server

6.6 Summary

This research work proposes the Content server to store zone and modeled behavior information of a multidisciplinary product model. This work starts with the HCI of a multidisciplinary product model where the model is handled directly by the IC web application or through an interface in the multidisciplinary product application. The operation and process of IC web applications are stored in the Content server. Then, the server is explained in brief, where data is stored in the Zone partition and Process partition based on the communication between the IC and RFLP structure. It is done by the BiC objects, CxiC objects and stored in the Content database. Finally, communication between the Content Server and CAD product server is explained where information of zone partition and process partition pushed temporarily to the CAD product server so that IC webtop application in the main application could handle the multidisciplinary product model. As Modelica and Info-Chunk objects are based on the OOP concepts, the RFLP structure and IC could be compatible with each other and exchange information easily. This research work is an effort to provide efficient user interaction of a multidisciplinary product model through the Information Content. This chapter covers the Thesis Group 3.

Thesis Group 3

Thesis group 3: Relationship between Info-Chunk and Virtual World

Thesis 3.1
I have introduced InfoChunkLib Application Programming Interface (API) [Y4], IC web application [Y9] and Content web server [Y10] for the representation of the behaviors of the multidisciplinary product. It is the practical approach to represent the community zone information and extracted modeled behavior data of a multidisciplinary product. As compared to the research work [45], where average behaviors of a product model have been implemented in a virtual test bench and its response by the real system, this research work focuses on the representation of the behaviors of a multidisciplinary product through the graphs. The proposed concepts can be used for collaborative engineering in terms of handling the multidisciplinary product indirectly by the Information Content. The Apache HTTP Server hosts the IC web application with a PostgreSQL database that is used to store the modeled behavior data from the process plane of the IC. The API is code by using the Java language as a JavaFX application. It is imported to the IC application and generates the graphs related to the behaviors of the multidisciplinary product. The author make an effort to visualize the outcome of the research work through the web technologies

Relevant own publications about this thesis group: [Y9], [Y4], [Y10].

Chapter 7

Relationship between the Info-Chunk and Virtual world

This work focuses on the practical approach for the behavioral modeling of the multidisciplinary product. This research work proposes an active knowledge representation of IC & IP using the Info-Chunk objects to store and represent the RFLP structure layers information effectively. Further, an active approach to behavior modeling of a multidisciplinary product model in the IC is outlined. Here, active models act as agents after exchange them with other modeling systems at applications of models as explained in the paper [63]. Finally, the practical feasibility of IC web application with the multidisciplinary product application of CAD system under various scenarios is explained. The chapter is structured as follows:
Section 6.1: Introduction.
Section 6.2: Role of Info-Chunk objects in the Active Knowledge Model.
Section 6.3: Behavior Modeling of Multidisciplinary Product.
Section 6.4: Application of Info-Chunk objects driven Information Content.

7.1 Introduction

This chapter proposes active knowledge representation in the multidisciplinary product modeling using the IC & IP. For the practical approach, behavioral modeling of a multidisciplinary product in the IC and IP is explained by using the sequence diagram. These diagrams are understood by engineers with significantly different backgrounds. Then, there are various scenarios considered based on the interaction between the IC application and the Multidisciplinary product application of the CAD system. This chapter starts with the format supported by the product model. Then, associative entities for active knowledge representation are explained. Following this, behavior modeling of the multidisciplinary product using Info-Chunk object with intelligent content is emphasized. It involves knowledge from the element of the design processes. Then, an active approach of the multidisciplinary product using Info-Chunk based modeling of engineering are outlined by the sequence diagram. Finally, the various scenarios of IC application with the multidisciplinary product application. Then, this research work is concluded.

7.1.1 Format Supported by the Product Model

Every CAD based company has its file format to store the product model. Some of the company based file formats are mentioned in the Table 7.1. It is stored as one of the formats depending on the CAD systems.

Table 7.1: Company specific formats to store product model

Company specific File Format	**File Extensions**
CATIA V5, V6	CATPART, CATPRODUCT
3DEXPERIENCE	3DXML
Inventer	IPT, IAM
Pro/ENGINEER/Creo	ASM, NEU, PRT, XAS, XPR
SolidWorks	SLDASM, SLDPRT

Table 7.2: Neutral formats to store product model

Neutral Formats	**File Extensions**
STEP	STP, STEP
IGES	IGS, IGES
VRML	WRL, VRML
JT	JT
Universal 3D	U3D

Further, companies also tend to support neutral file formats that are designed for inter-operating ability. The neutral format is supported by almost all of the CAD systems. Therefore, company specific file format is translated into the neutral format so that it is accessed by different vendors of CAD system. Some of the neutral file formats are mentioned in the Table 7.2.

7.2 Info-Chunk Objects in the Active Knowledge Model

An active and knowledge-based version of associations and constraint-driven multidisciplinary product modeling is proposed by using the Info-Chunk objects as shown in Fig. 7.1.

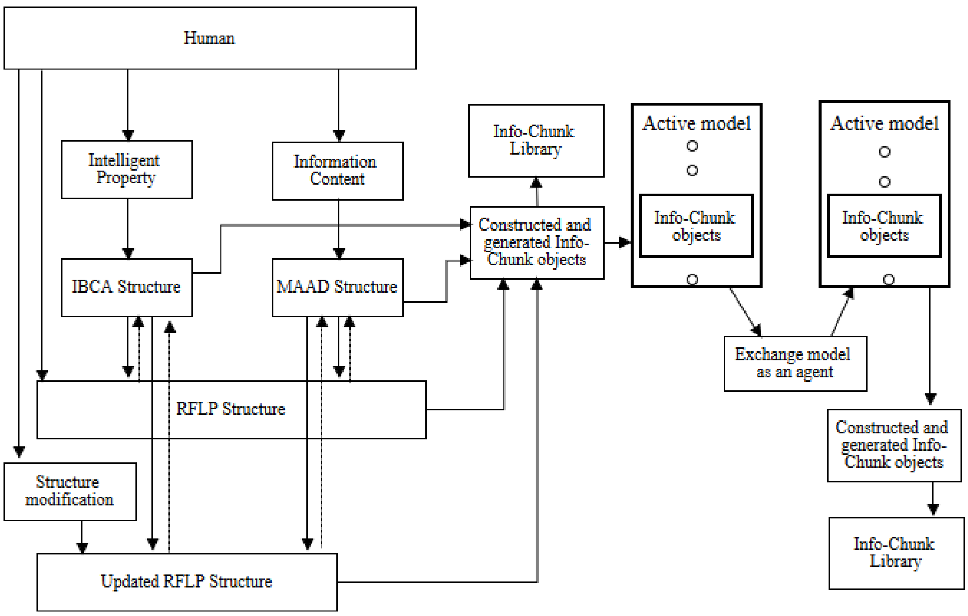


Figure 7.1: Active model of Multidisciplinary product using Info-Chunk Objects

The solid lines represent the operation performed for handling the information of the RFLP structure. The dashed lines represent the response of the RFLP structure in the form of Info-Chunk objects. Humans interact with the IC & IP to handle the information of the multidisciplinary product model. Through Info-Chunks objects of MAAD structure & IBCA structure, IC & IP communication with the LiC entities of RFLP structure. As explained

in the previous chapter, Info-Chunk objects refer to the BiC & CxiC objects. To model the behavior data, the Process plane of the IC interacts with the Info-Chunk objects of the *Product Behaviors* level of the MAAD structure, which further, drives the Info-Chunk objects of the Functional level and Logical level of the RFLP structure. The IC and IP are used for the active knowledge representation of the RFLP structure. For every operation, there is a set of Info-Chunks objects stored in the MAAD structure, IBCA structure & RFLP structure. Based on the information exchange, the LiC entities of the RFLP structure is updated accordingly. In the case of the IC, the information stored in the Info-Chunk objects of the MAAD structure is used to store the modeled data of a multidisciplinary product model.
The information is displayed in the representation plane of the IC in the form of the web application. The IC application could be a static or dynamic web application and coded by HTML, CSS, PHP, JavaScript or ASP. The Info-Chunk objects are stored in an Info-Chunk library or database. The database could be a Hierarchical, Relational, and Object-oriented or ER model database. It is stored by MySQL, PostgreSQL or MongoDB. It is updated and shared with the remote humans according to the requirement and permission. The above-outlined approach provides the solution for the construction of exchanged multidisciplinary product models in the remote CAD systems where data is transferred as the Info-Chunk objects. This model is accessed by the remote human to contribute and provide suggestions accordingly. As the knowledge model of a multidisciplinary system is handled by the IC web application. Therefore, it is easily handled remotely to contribute and provide suggestions accordingly. The most appropriate forms of knowledge are graphs, updated assembly models, updated part models, representations, formulas, rules, and checks. The IBCA structure organizes and supports the processing of active driving knowledge content in the product model. Driving active knowledge relies upon company expertise and experience in a contextual generic product model. IBCA and MAAD structures are the driving factor for representing the knowledge of the RFLP structure.

7.2.1 Associative Entities in Knowledge based Models

The generic part related knowledge is included in the product model for the active application. The associative relationship between model entities is defined for integration product-related partial models [64]. Knowledge is related to associations, creation and modification of model entities rely on maintaining associations and is defined as constraints. Knowledge-based modeling solves the nonlinear mathematical optimization problems of mechanical parts by using numerical algorithms. The related knowledge is represented in the form of rules and checks. As Info-Chunk is an entity, it should be converted into the object first. After the conversion, Info-Chunk objects retrieves the modeled behavior data from the LiC entities stored in the Functional and Logical layer of the RFLP structure. It can communicate with the Info-Chunk objects of the MAAD structure and IBCA structure to transfer the layer information for behavior representation of the multidisciplinary product model. IC and IP store and represent the information of the Info-Chunk objects for active knowledge representation of the multidisciplinary product model. In the case of IC, it is represented by the IC web application, which displays the information of the representation plane of the IC. The information is stored in the Content database. The objects store the information of the design goal that is represented by the optimization intent such as cost, volume, time, mass, stress, and displacement. The objects must have satisfied specific design limits such as material strength or allowable displacement which are constraints that are represented by the associations between the entities. For example, the sensitivity analysis and adaptive analysis information are stored as analyzed Info-Chunk objects in the MAAD structure and IBCA structure. This knowledge is used in the behavior-based reactive geometric model which enables a feature of intelligent modeling called the automatic contextual change of model.

7.3 Behavior Modeling of Multidisciplinary Product

Behavior-based models with intelligent content involve specifications and knowledge for the design processes. The knowledge-based model is used for the behavior modeling of a multidisciplinary product model and can access remotely by the cloud.

7.3.1 Data-driven modeling of Multidisciplinary Product

The above-mentioned concepts represent the dynamic behavior of a multidisciplinary product model. Further, one of the stimuli types is data-driven modeling [65]. Here, The sequence diagram is used to show the interactions between the human and IC as explained in Fig. 7.2. Human defines intent in the IC to handle the RFLP structure. IC communicates with the RFLP structure via the MAAD structure. Info-Chunk objects from the Functional and Logical layer of the RFLP structure are used to store the design and

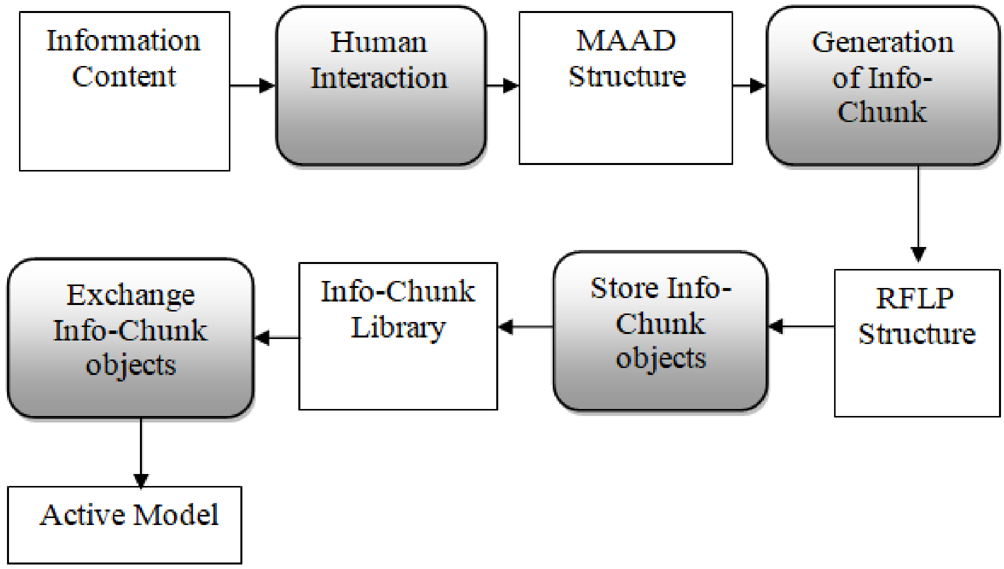


Figure 7.2: Sequence model by using Information content.

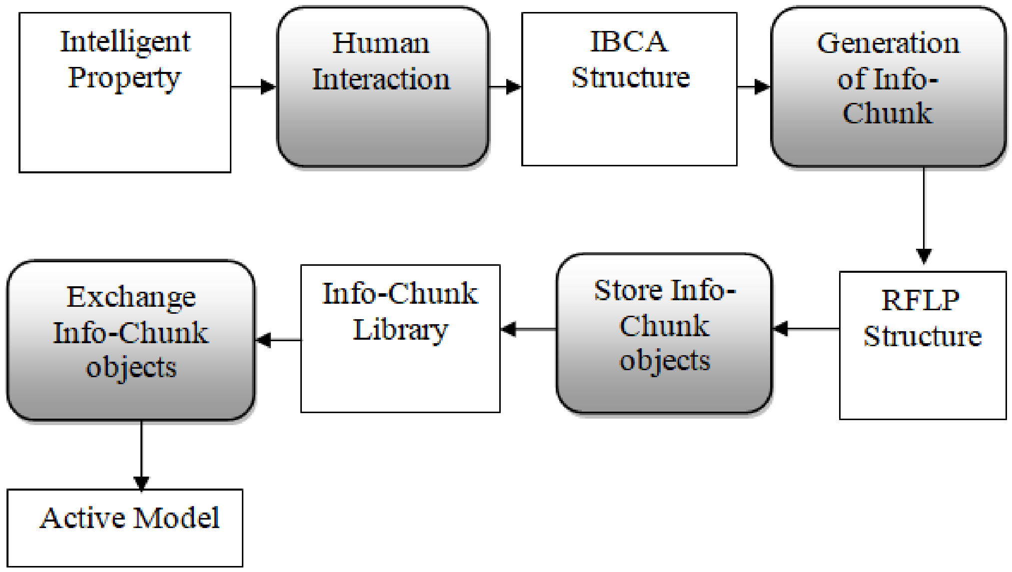


Figure 7.3: Sequence model by using Intelligent property.

geometry information of the multidisciplinary product model. Based on certain situations and circumstances, Info-Chunk objects of the MAAD structure store the modeled behavior data from the Info-Chunk objects of the RFLP structure. Further, these objects are analyzed and optimized in the process plane of the IC. The Info-Chunk objects are stored in the Info-Chunk library or database, which is exchanged by using the active model. In the case of IBCA structure, human interaction with the IP to handle the RFLP structure as shown in Fig. 7.3. IBCA Structure is used for active knowledge content and generates Info-Chunk objects based on the information. These objects are obtained from the Info-Chunk objects of the RFLP structure. The Info-Chunk objects are stored in the Info-Chunk library or database, which is exchanged by using the active model.

7.4 Application of Info-Chunk objects based Information Content

In the last section, active knowledge inside the content is represented by Info-Chunk objects. Now, the question is the practical feasibility of active knowledge content. There are four possible proposed scenarios for the interaction between the IC application and Multidisciplinary application.

- **Scenario A**: Multidisciplinary product application and CAD software are on the local machine and IC application is on the webserver
- **Scenario B**: Multidisciplinary product application and CAD software is on the one web server and IC application is on the other web server
- **Scenario C**: Multidisciplinary product application and IC application are on the same server. Here, multidisciplinary product software could not be required
- **Scenario D**: IC application is accessible through an interface from the multidisciplinary product application

Scenario A is the case when multidisciplinary CAD software and generated multidisciplinary product model application is on the local machine of the user and IC application is on the web server as shown in Fig. 7.4. This is a very common case as most of the CAD software like SolidWorks, AutoCAD, CATIA V5, etc. is installed on the local machine. There is one interface required between the IC and CAD software. It is difficult for the IC application to interact with the multidisciplinary application as CAD software has its format. Therefore, it should be converted into a neutral format. The coding part of the IC could be complex. Further, IC handles the information of a multidisciplinary product model via the web. The generated output is stored on the Content web server via Info-Chunk objects. The advantage is it requires only one machine to take the license of CAD software. The other users are accessing the software via the IC application. It saves money and memory of the systems. The drawback is the user cannot interact with the multidisciplinary product.

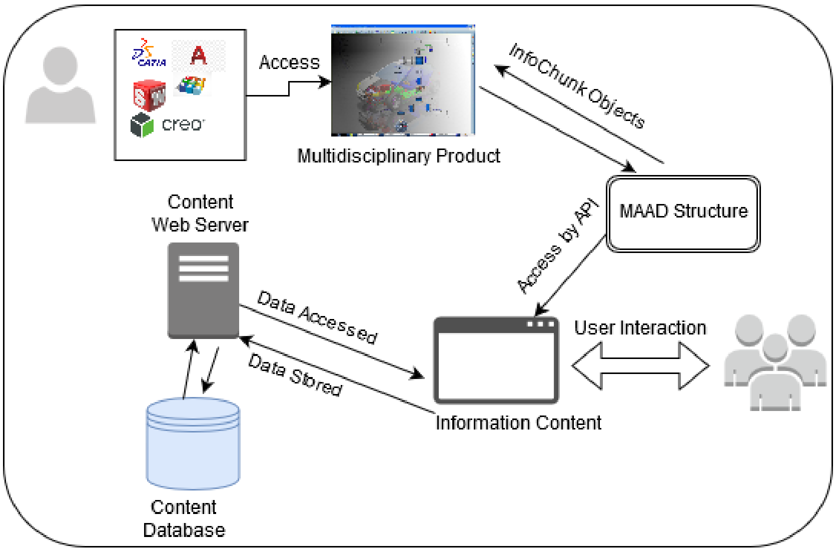


Figure 7.4: Multidisciplinary software, application on the local machine and information content application on the server

Scenario B is the scenario when multidisciplinary CAD software and generated product model applications are on the one web server and IC application is on the other web server as shown in Fig. 7.5. This is the case, for example, when cloud-based multidisciplinary CAD software like Dassault Systems CATIA 3DEXPERIENCE interacts with the IC application. Both applications have their web server for resource management and database to locate the information. There is one interface required between the IC and cloud-based CAD software. It is comparatively easy for IC application to interact with the multidisciplinary application as there is only one format i.e.3DXML. Therefore, the coding part of the IC application could be easier. The advantage is the user can interact with the multidisciplinary product model as well as IC via the web. The drawback is every user/organization could take the license of CAD software.
Scenario C is the scenario when the multidisciplinary application and IC application are on the same web server as shown in Fig. 7.6. It requires CAD

software only once for the generation of the multidisciplinary product application. As there is plenty of CAD software, it is recommended to convert the company-specific format to the neutral format. This format is uploaded to the web server, where the IC application handles the multidisciplinary application. Through Info-Chunk objects, IC modifies the information in the multidisciplinary application, so that it can provide simplified and easy human interaction of the product model.

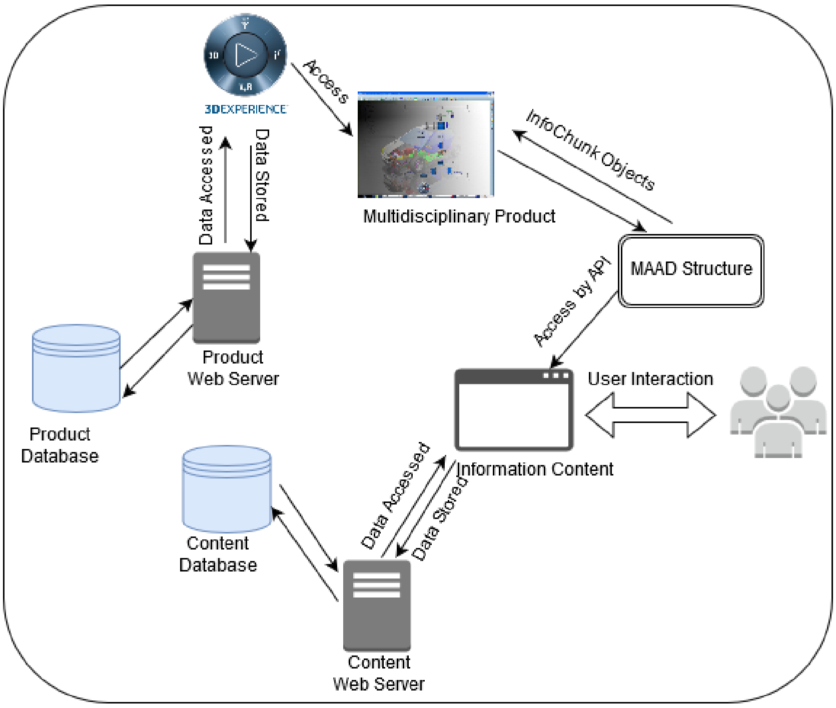


Figure 7.5: Multidisciplinary application on one server and IC application on another server

The IC application is accessible by the humans within an organization by the intranet or internet. The generated output from the IC application is stored on the same web server and output information in the content database. The advantage is there is no need for a CAD software license dedicated to the IC application. The drawback is the passive multidisciplinary application, complex database, and overloaded web server. Scenario D is the situation when Information Content application is accessible by an interface from the multidisciplinary product application. The user interacts with the Multidisciplinary product application to access the IC application through a separate plane as shown the Fig. 7.7. There is an interface between the two

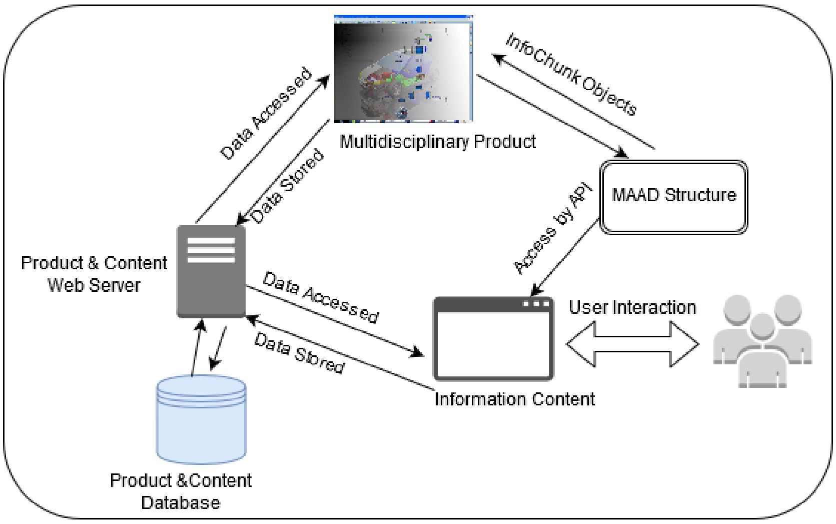


Figure 7.6: Multidisciplinary application and IC application are on the server

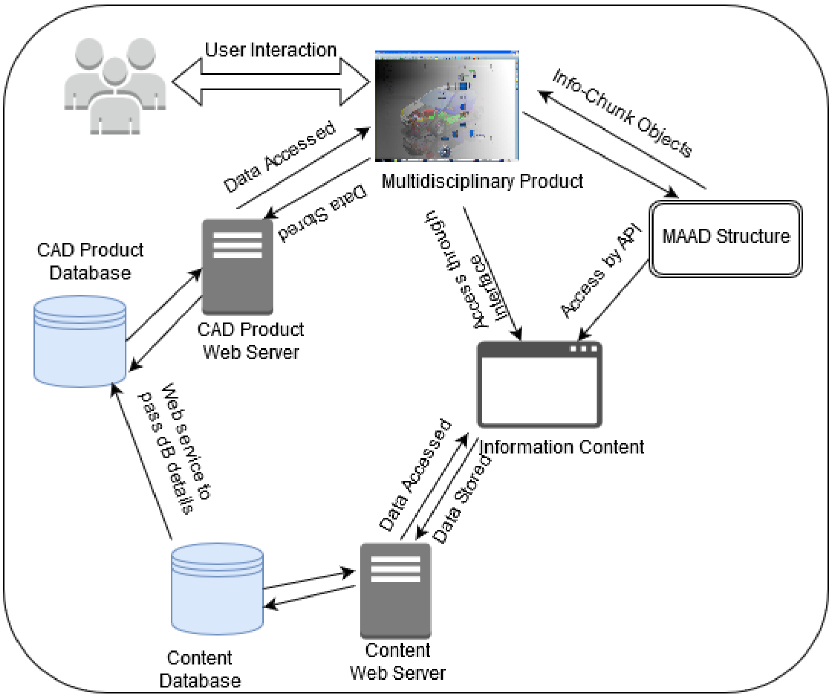


Figure 7.7: IC application is accessible from the Multidisciplinary Product application

applications. The database of the CAD product server retrieves the process and zone partition information by using the web services from the database of the Content web server. The web service API could be REST, SOAP or RESTful. The advantage of this case is the user can only interact with the multidisciplinary application with the benefits of IC web application. The drawback is wastage of resources.

7.4.1 Result Analysis and Discussion

The Windows operating system is installed on the virtual machines. Based on the proposed scenarios, CAD Software, CAD server, CAD application, and IC server are placed on the virtual machines. As we found that, Scenario D is the most efficient case as it needs one server for all the operations. Scenario A & Scenario C is the most inefficient case as it needs to convert the CAD file format into a neutral format. Accuracy and complexity is the issue. Scenario B is the feasible case as CATIA 3DEXPERIENCE is available in the market and it is possible to handle the CAD application through the IC application.

7.5 Summary

This research work proposes the active knowledge representation of the multidisciplinary product by using Info-Chunk objects based on IC and IP. Here, the knowledge-based model of the multidisciplinary product is described using the Info-Chunk objects. The objects of the MAAD structure and IBCA structure are stored in the Info-Chunk library. This library is accessed by using IC and IP respectively. Through the knowledge model, behavior modeling of IC and IP are proposed. Here, the sequence diagram is used to represent the sequence of actions involved during the interaction between humans and IC and IP in processing the Info-Chunk objects. The automatic, reactive Info-Chunk objects based propagation and any change in the multidisciplinary product at any stage of the modeling process makes the design consistent with intents, goals, and decisions. Info-Chunk objects provide the necessary specification and knowledge representations to represent the knowledge of the multidisciplinary product. Finally, the interaction between humans, IC and multidisciplinary application are explained by various scenarios. This research work provides the simplified interaction between the humans and multidisciplinary product through Info-Chunk objects based Information Content. This chapter covers the Thesis Group 3.

Thesis Group 3

Thesis group 3: Relationship between Info-Chunk and Virtual World

Thesis 3.2
I have introduced a practical approach to the Behavior modeling of a multidisciplinary product model using the Information Content (IC) and Intelligent Property (IP) [Y5]. In the paper [38], Feature definition in case of agent based active model is outlined. In this research work, Info-Chunk objects are used for the active models as it stores the detailed information of the modeled behaviors data from the functional and logical layer of the Requirement Functional Logical Physical (RFLP) structure based on the disciplines. Further, the active knowledge based model is used in the Information Content (IC) for the behavior modeling of a multidisciplinary product and can be accessed and updated remotely by the cloud. The proposed method can be considered as an extension of collaborative engineering in terms of the product design to take the collaborative actions by the various engineering disciplines. For the flexibility of the Information Content (IC), there are various scenarios considered for interaction between the Information Content (IC) application and the multidisciplinary product application.

Relevant own publications about this thesis group: [Y5].

Chapter 8

Behaviors Representation for the Cyber Physical System

The Cyber Physical System (CPS) forms the basis of emerging and future smart services as it collaborates computational entities with the surrounding physical world. It demands a system-level product model where it requires advanced engineering modeling and the possible solution is multidisciplinary CPS. Therefore, this chapter works on the representation of the behavior of the multidisciplinary CPS. It is structured as follows:
Section 8.1: Introduction
Section 8.2: Extended Engineering Model System (EMS) using Info-Chunk driven RFLP Structure
Section 8.3: Active Information Content (AIC) using Info-Chunk driven RFLP Structure
Section 8.4: Case Study
Section 8.5: Summary the chapter

8.1 Introduction

The CPS is the origination of smart products in embedded intelligent information and communications technology. Modeling of the CPS is the advanced stage of modeling where a system can be smarter with the power of computing. Hence, it is more complex than the product modeling. To handle the modeling of multidisciplinary product, concepts of CPS enable IC [66] for EMS and AIC structure [31] was proposed. They are used for driving entities in the multidisciplinary representation of the CPS system. IC was introduced as an active means to organize and relate knowledge applied to drive the decision on product information that serves as an integrated and very complex description and representation of an Engineering structure (ES). AIC was conceptualized to establish CPS enabled IC and establish IC level communication between RFLP structured and CPS environments. The decision on changes in the product model is done according to the result of behavior analysis. In the RFLP structure, behavior analysis is done in the functional and logical level. Here, the ES model system is prepared for the representation of cooperating systems and manufacturing, and its connections with cyber units of CPS. After its production and installation, ES is operated as CPS. In the previous work, the author introduced the concepts of Info-chunk in the Functional and Logical layers of the RFLP structure. This chapter is structured as follows: Info-Chunk driven RFLP structure is applied to the extended EMS for the CPS. Further, communication between IC and Info-chunk in the extended EMS is discussed where IC is the mode of communication between Info-Chunk driven RFLP structure and cyber units CPS structure. Then, the role of Info-Chunk in AIC is discussed. Further, communication between AIC and Info-chunk is explained. Finally, the Summary of the chapter is discussed.

8.2 Extended Engineering Model System (EMS) using Info-Chunk driven RFLP Structure

Info-Chunk driven RFLP structure is applied in the extended EMS [25] for CPS modeling as shown in Fig. 8.1. It transfers the information of Info-chunk to the cyber units by the information content that includes knowledge and information required to drive object parameter generation procedures on other levels of extended EMS [25]. These objects parameters are retrieved from the LiC entities of the Functional and Logical level of Info-Chunk driven RFLP structure model. Requirements by CPS system connection track the real situations and provide feedback according to the changed situation. It is done by mutual communication of content and information to this list. Here, details are given about Info-Chunk driven RFLP structure and main driving contexts of IC in the extended EMS. As explained in the [67], the structure of IC includes four levels for intent status, systems, behaviors and actions to record, establish structure, organize behaviors, situations, simulations, organize physical level actions on physical level Engineering Structure model and cyber units of CPS. Communication between Info-Chunk driven RFLP structure and IC are explained in Fig. 8.2 [67]. Systems level organizes system related IC in substructures where Substructure Systems operate the ES, Substructure Functions operate functions in the context of systems and Substructure Contexts operate the context of a logical component of the system. In the Fig. 8.2 functional and logical sub-levels of system level model depend on behavior representations. *Behaviors level* of IC supports the behavior representation definition.

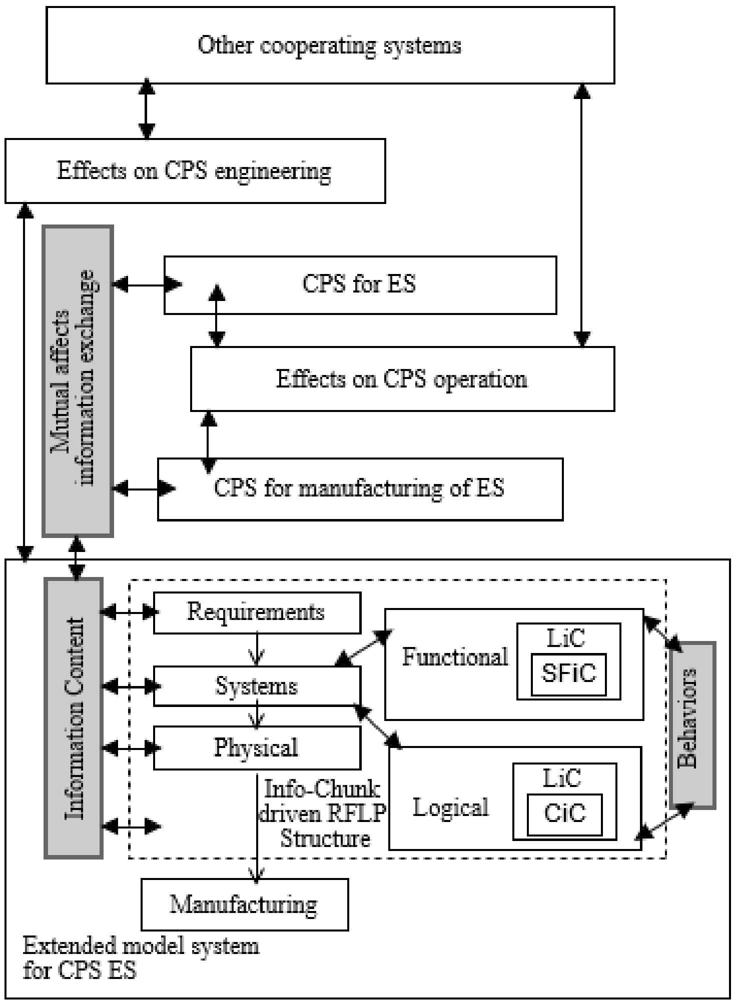


Figure 8.1: Extended Engineering Model System (EMS) with Info-Chunk driven RFLP structure

The behavior information from Info-Chunk driven RFLP structure is retrieved from the LiC of the functional and logical layer. This information is stored by the Substructure Systems of the IC and communicates with the Substructure Behaviors. *Levels of Actions* are defined in the context of behaviors obtained from Substructure Behaviors. Based on the LiC information obtained from the Substructure Behaviors, Substructure features drive the physical level features in the RFLP structure and send information to the substructure controllers which is a contextual structure of the content in the background of controller definitions. The substructure controller handles the cyber units by substructure procedure and feedback is retrieved by substructure features.

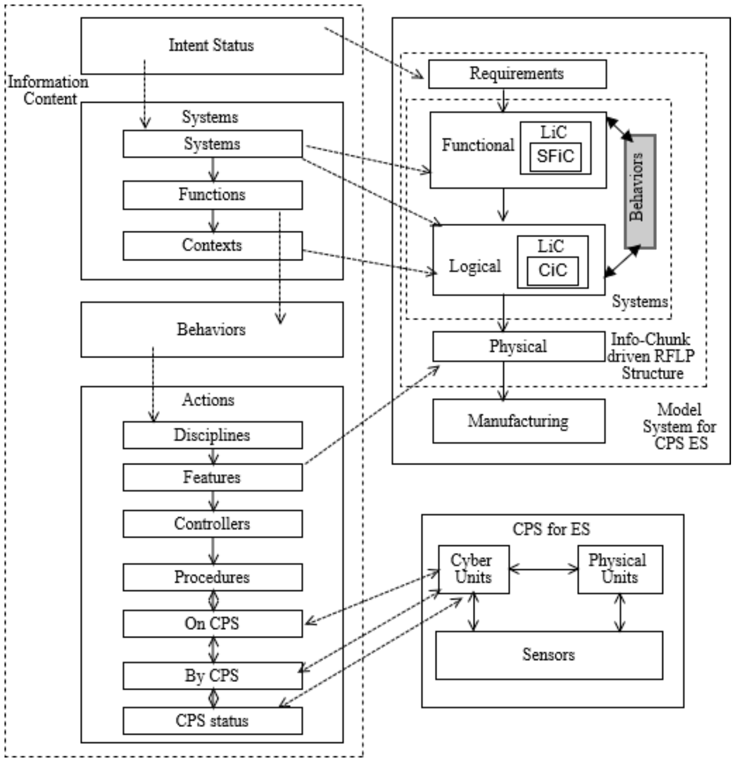


Figure 8.2: Extended EMS for CPS ES with LiC entities

Elements of cyber units depend on the *On CPS*, *By CPS*, and *CPS status* substructures on the *Actions level* that provide or receive suggested, experienced, and monitored types of CPS operation parameters. Element in substructure On CPS carries information about suggested CPS operating parameters, Element in substructure By CPS carries information about experienced CPS operating parameters and Element in substructure CPS status receives monitoring information for the status of physical units in CPS. The status information together with the CPS situation by specific simulation suggested parameter and situation information for cyber units of CPS. Based on the suggested parameters and situation, parameters of LiC in the functional and logical levels of the RFLP structure change its value. Like many other Industry 4.0 related concepts, these concepts are part of developing CPS technology. Actual (decided), predicted, and experienced situations are defined. Based on the experienced situations, values of Info-Chunk can be changed so that the system can be operated accurately according to the situation.

8.3 Active Information Content (AIC) using Info-Chunk driven Multidisciplinary Product Model

AIC [31] structure was introduced to extend the capabilities of information content structure to support active driving entities in RFLP and CPS cyber structures. Driving connection of AIC,RFLP structure, organized IP transfer structure, production system model (SPR) and CPS was discussed in the paper [31] as shown in the Fig. 8.3.

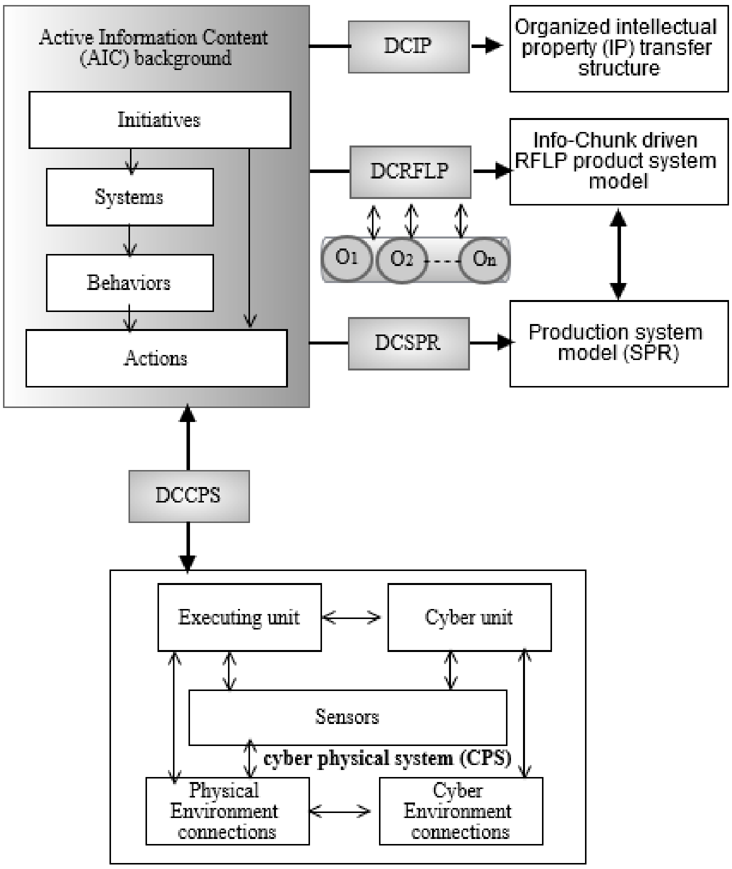


Figure 8.3: Main driving contexts of Info-Chunk driven RFLP structure from AIC

This research work focuses on the driving connection between AIC and Info-Chunk driven RFLP structure. Outside driving connections are between AIC element and Info-Chunk driven RFLP structured production system is the model component (DCRFLP). The author proposed the objects in the DCRFLP component that can transfer the Info-Chunk related information between AIC and RFLP structure. The objects can carry LiC information on the functional and logical levels of the RFLP structure. The objects are based on the OOP and contain attributes and methods. Behaviors are applied as feature independent and discipline-specific P level features in the RFLP structure. The virtual execution of the product concept model needs the representation of dynamic and state logic behaviors in F and L level components. Therefore, LiCL and CiC store the knowledge of Logical level and LiCF and SFiC store the knowledge of the Functional level. The *behavior level* of AIC structure retrieves the knowledge of Functional and Logical level entities by the objects to establish full behavior driving of product definition. AIC levels, substructures, and main contexts were explained in the paper [31]. The author focuses on the communication between AIC and Info-Chunk driven RFLP structure in terms of dynamic and state logic behaviors as shown in Fig. 8.4. The rest of the contexts are the same as explained in the paper [31]. Behavior substructure elements drive Functional and Logical elements (contexts d and e, respectively).
Behavior substructure elements are function-driven (context a). Functions substructure elements on the systems level of AIC structure drive Functional level components in the RFLP structure (context b), retrieves the objects from LiCF and corresponding SFiC entities for driving the content of system representation in the RFLP structure. AIC structure drives system in RFLP structure through objects in the substructure functions using LiC related parameters. Elements in functions driven context substructure to assure system centered driving of LiC entities of Logical level components by objects in the RFLP structure (context c). Features substructure elements drive procedures elements through the related controllers which are included in CPS to control executing units. It is simulated in the product model and findings of simulations are applied at the control of executing CPS units. Controller related entities are represented on the Physical level of RFLP structure and are driven from controllers substructure (context k) and higher levels of RFLP structure. CPS control units are connected with elements in the procedures substructure (context h). Controllers substructure elements drive procedures substructure elements that are driving the generation of Physical level components in the RFLP structure (context g). Element in simulations substructure drives elements in the controller substructure (context f). CPS specific substructures are physical actions and physical status and are in connection

with the decision (context i)and monitor (context j) functions in cyber units of the CPS, respectively. Physical actions substructure element helps decisions in CPS and values in LiC entities of the RFLP structure are modified through objects.

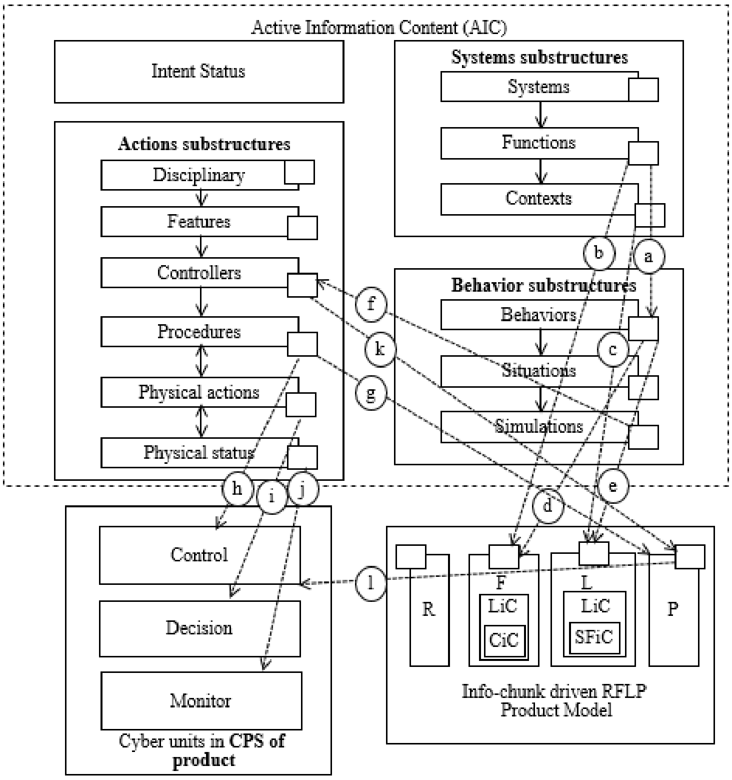


Figure 8.4: AIC levels communication with Info-Chunk driven Multidisciplinary Product model

8.4 Case Study

To explain the above mentioned research concepts, let as consider the Cyber Physical System (CPS) in the Smart Manufacturing as per the Industry 4.0. There is a set of interacting systems, where highly skilled engineers will be provided with operational insights directly from coordinated intelligent machines controlled by a central entity. Here, physical machines in an automotive plant are fitted with massive sensors that send status data to a virtual reproduction constantly. Engineers can work on potential problems and possible solutions directly in this cyber replica. In the case of Info-Chunk entities extended EMS and AIC for CPS modeling, the Info-Chunk objects transfers the information of Info-chunk entities to the cyber units by the information content that includes knowledge and information required to drive object parameter generation procedures.

- The transferred information from the functional layer of the RFLP structure is Requirement class attribute of the LiCF entity and corresponding Elements description attributes of the SFiC entity.

- The transferred information from the logical layer of the RFLP structure is situation attribute and data model attribute from the LiCL entity and corresponding data model attribute and behavior attribute of their CiC entity. The information retrieved is the actual situation, circumstances for the situation, the adaptive drive to drive context definitions, concept behavior, activity, adaptive and product feature contexts, connection behavior definitions, model definition activities, contexts for an adaptive drive, and context for the physical level product.

In the case of extended EMS, the experienced situations change the values of the Info-Chunk entities so that the system can be operated accurately according to the situation. Whereas, In the case of the AIC, the Physical actions substructure helps the decisions in the CPS by modifying the values of the LiC entities.

8.5 Summary

This chapter proposed Info-Chunk driven RFLP structure in the CPS modeling. Extended Info-Chunk driven RFLP structure is applied in the IC based EMS for CPS modeling. Here, communication between Info-chunk and IC is explained. Then concepts of OOP are proposed in the AIC where communication between AIC, Info-chunk and cyber units is done by the objects. The main purpose of this research work is to simplify the communication between the EMS, AIC and RFLP structure by Info-Chunk entities and concepts of OOP in the CPS modeling. This chapter covers the Thesis Group 2.

Thesis Group 2

Thesis group 2: Info-Chunk driven Information Content for the Multidisciplinary Product Modeling

Thesis 2.3

I have introduced the Info-Chunk entities concepts in the extended Engineering Model System (EMS) and Info-Chunk objects concepts in the Active Information Content (AIC) for the behavior representation of the multidisciplinary Cyber Physical System (CPS) [Y7]. As mentioned in the paper [66], there is a need for a definition of model entities, object parameters, and contextual connections with the demand of implementation in the multidisciplinary system based industrial model by using configuration, structuring, object definition and programming capabilities for CPS systems [31]. Therefore, this research work focus on Information Content based EMS and AIC. Here, communication between the Information Content, proposed LiC entities and CPS entities are explained first. Based on the experienced situations, values of Info-Chunk can be changed so that the system can be operated accurately according to the situation. Also, communication between AIC, Info-Chunk objects and cyber units is done through dynamic and state logic behavior. As compare to the existing methodologies, the proposed entities and objects are a novel approach for the contextual connections between the EMS, AIC and multidisciplinary CPS. The author made an effort to represent behaviors efficiently in CPS modeling through Info-Chunk entities and objects.

Relevant own publications about this thesis group: [Y7].

Chapter 9

Behaviors Evaluation of Multidisciplinary Product using Fuzzy Logic

This chapter describes the evaluation of the behavior of a multidisciplinary product by proposing the Requirement, Function and Logical Block corresponds to the RFLP structure product model. After that, soft computing is used to improve the behavior of a multidisciplinary product in real-time. The chapter is structured as follows:
Section 9.1: Overview of the concepts.
Section 9.2: Introduction of the Requirement Functional Logical (RFL) Block.
Section 9.3: Fuzzy Logic concepts for Multidisciplinary product modeling.
Section 9.4: Comparison between Fuzzy Inference Systems.
Section 9.5: Summary the chapter.

9.1 Overview

Recently increasingly come into view soft computing methods applying Analytical Network Processes (ANP) and Analytic Hierarchy Process (AHP) approach where the learning of the decision or management process is supported by the soft computing methods [68]. There is three essential product behavior represented in the RFLP structure namely dynamic logical behavior, global dynamic behavior and discrete behavior as explained in the paper [6]. Discrete behavior is evaluated and represented in Functional and Logical level elements of the RFLP structure. There are numerous ways to analyses the discrete behavior of the product in the RFLP structure. The discrete behavior of a multidisciplinary product is determined by using the concepts of soft computing. Mamdani's FIS and Adaptive Neuro FIS are considered in the context of this research. Here, the FIS is represented by fuzzy sets, fuzzy operators and knowledge base [69]. In this chapter, the author proposes the concept of discrete behavior monitoring of a product using soft computing because it allows modeling a complex dynamic system more intuitively as well as can deal with uncertainty and non-linearity. Here, the author uses fuzzy logic in the combination of neuro computing. This chapter is structured as follows. It introduces Requirement Block, Function Block and Logic Block for the RFLP structure. Later, the scope of soft computing in product modeling is discussed. In this context, Mamdani's FIS and Adaptive Neuro FIS are discussed. In Mamdani's FIS, member function and rules are defined then system behavior is evaluated. Similarly, for Adaptive Neuro FIS, member function and rules are defined then system behavior is evaluated. Next, a Comparison between Mamdani's FIS and Adaptive Neuro FIS is discussed. Finally, the Conclusion and Results are discussed.

9.2 RFL Blocks

To explain the concepts of soft computing in product modeling, Different blocks are proposed corresponding to the Requirements, Functional and Logical levels of RFLP structure. Here, a camera drone is considered as a multidisciplinary product for the explanation of proposed work as shown in Fig. 9.1. Here,h is the altitude of the drone to the ground and d is the distance between car and drone respectively.

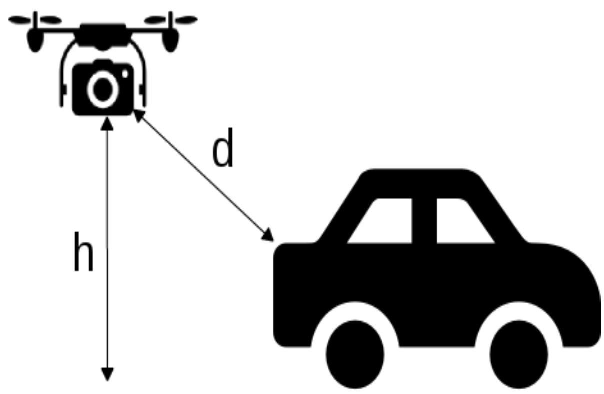


Figure 9.1: Drone Camera as a Multidisciplinary Product

9.2.1 Requirement Block

Requirements of a product are defined by end-user or customers according to their needs. In other words, end-user is expecting certain types of behavior from a product. The Requirement level is represented by Requirement block which is the collection of different kinds of behavior expecting by the end user as shown in Fig. 9.2. Here, every behavior has a different level of priority according to the needs of end user i.e. some behaviors are more important than others.

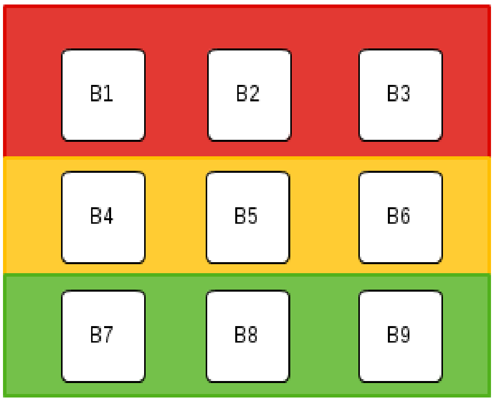


Figure 9.2: Requirement Block of a Product System

From the above mentioned product, Expected behaviors can be as follow: recognize the target, capture high quality pictures and videos, maintain threshold distance and so on.

9.2.2 Functional Block

To fulfill the requirements of a product, engineering objects should attain a certain set of functionalities. Therefore, Functional Block is proposed which is the collection of sub-functionality of single or multiple EOs as shown in Fig. 9.3. It is interesting to note that every Functional block belongs to the particular engineering object i.e. all functionality of a EO placed in a Functional block.From the above mentioned product, engineering objects are drones and cameras respectively. Functionality may be as follow: Establish the connection between target sensor and product sensor, focus the camera on target with appropriate angle, check the battery and memory of camera in a given interval of time and so on.

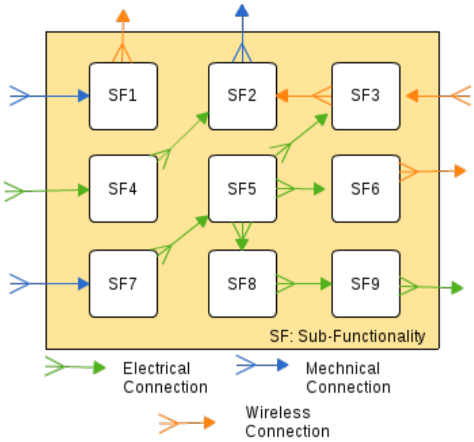


Figure 9.3: Function Block of a Product System

9.2.3 Logical Block

The Logical level can play an important role as it is a mandatory level to define product behavior. It means a product cannot behave properly if there is some error in logical connections. Here, Logical block is proposed corresponds to the Logical level and logical connections between different engineering connections can be defined as shown in Fig. 9.4. There is a different kind of logical connection that exists in different sub-functionalities of the function block.

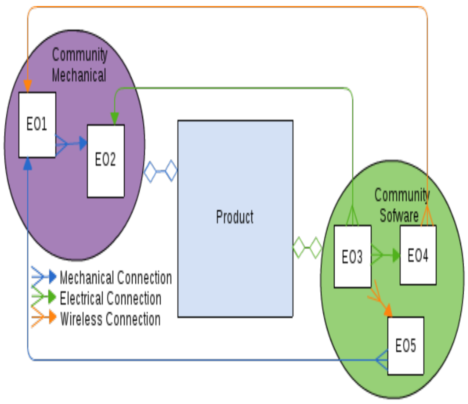


Figure 9.4: Logical Block of a Product System

For above mentioned product, Drone can be placed in the mechanical community and a Camera can be placed in the software community. The logical connection can be an electrical, mechanical, wireless connection. Finally, the Physical level is the abstraction of a product without taking care of its complexity.

9.3 Fuzzy Logic concepts for the Multidisciplinary product modeling

Fuzzy Logic is tolerant of imprecision, uncertainty, partial truth, and approximation. Therefore, it is one of the possible approaches to analyze the product system behavior using the RFLP structure. There are numerous ways to analyze system product behavior. In this chapter, the author analyzed the dynamic behavior of system product behavior of multidisciplinary product by using principal constituents of Fuzzy Logic as follow:

- Mamdani's FIS
- Adaptive Neuro FIS

9.4 Mamdani's Fuzzy Inference System

Fuzzy logic is tolerant of imprecise data. Here, Output FIS variable is the expected behavior that can be evaluated using Mamdani FIS from the different combination of input variables, which is sub-function values of different or same function blocks as shown in Fig. 9.5. Further, In the context of this

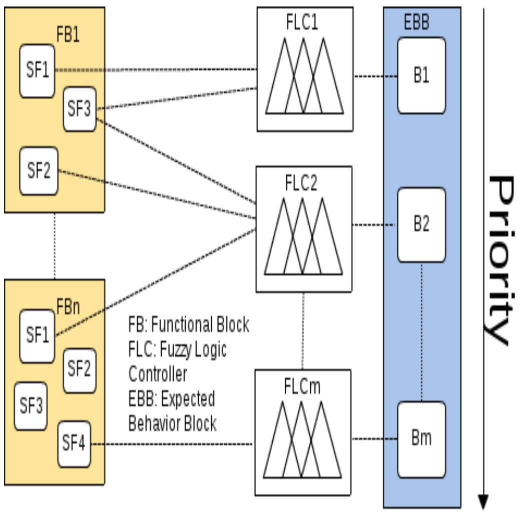


Figure 9.5: Evaluating particular Behavior using Sub-functionality

work, Input variables are categorized into two parts:

- Static variable: sub function values, which cannot change with time for example resolution of camera.
- Dynamic variable: sub function values, which can change with time for example distance between drone and car.

9.4.1 Defining Membership Functions

In case of input FIS variables, membership function may be the worst scenario, acceptable scenario and best scenario while for ouput FIS variable (behavior), membership function can be fail, average and excellent. It is important to note that author used Trapezoidal-shaped membership function as input and output FIS variable.

9.4.2 Defining Rules

Rules for evaluating the expected behavior can be explained using the table below. The terms BS, AS and WS refers to Best Scenario, Acceptable Scenario and Worst Scenario respectively.

Case	Behavior	Majority	Minority
1	Excellent	BS	AS
2	Average	AS	WS, BS
3	Fail	WS	AS,BS

Table 9.1: General table to define Fuzzy rule

9.4.3 Evaluating System Discrete Behavior

To understand the situation, consider an example for tracking the car as an expected behavior/output FIS variable from the above mentioned product. To obtain this behavior, input variables can be the resolution of the camera and distance between car and drone respectively. In this case, the membership function for the input variable has equivalent values and can be explained using the table below: From the above-mentioned configuration, Mamdani's FIS can be constructed of input and output FIS variable. Similarly, other Mamdani's FIS are constructed equivalent to the other expected behaviors by using the same procedure. The next step is to give priority to the obtained behaviors according to the requirements of the customer.

Input	BS	AS	WS
resolution of camera	high	medium	low
distance	within range	close	far
altitude	within range	high	low

Table 9.2: Membership Function considering car system

Finally, Product system behavior has been evaluated by constructing Mamdani's FIS and taking behavior's Mamdani FIS and corresponding priorities as an input FIS variable as shown in Fig. 9.6.

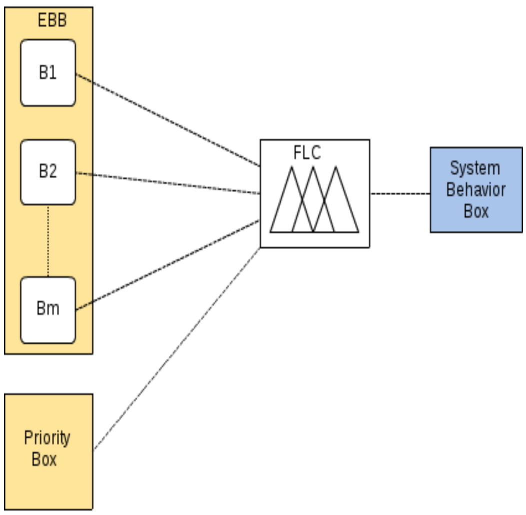


Figure 9.6: Evaluating Product System Behavior using Mamdani FIS

Here, the Priority box defines the budget of the customer corresponding to the system and assigns the priorities to the behaviors of the system according to the budget of the customer.

Priority Value	Majority	Minority
Low Budget	MB	-
Medium Budget	MB	IB
High Budget	MB, IB, OB	-

Table 9.3: Rules used by the Priority box

In the virtual environment using the RFLP structure, any type of behavior can be implemented in product modeling. But, in the Physical Environment, the product has been constrained in terms of budget. Therefore, the Priority box is proposed. In terms of fuzzy logic, It is an input to the controller and membership function can be defined in terms of low budget, medium budget, and high budget. Rules can be defined using the table above. The terms MB, IB and OB refer to the Mandatory Behavior, Important Behavior, and Optional Behavior respectively. Rules for evaluating the car system discrete behavior can be made using the table below.

System Behavior	Majority(Expected Behavior, Priority)	Minority(Expected Behavior, Priority)
Excellent	(Excellent,High)	(Average,Medium) and/or (Average,High) and/or (Excellent,Medium)
Average	(Average,High) and/or (Excellent,Medium)	(Average,medium) and/or (Excellent,low) and/or (Average,low) and/or (Worst,low)
Fail	(Worst,High)	Any combination

Table 9.4: Rules for evaluating the car system discrete behavior

It is important to notice that product system behavior is time-variant. Some inputs are static while others are dynamic for example camera resolution is static input while the distance between car and drone is dynamic input. In terms of executing the above-mentioned concepts, create a Simulink model and import all the specific behavior Mamdani's FIS and their inputs (static variables and dynamic variables) in its work space. Finally, import system behavior Mamdani's FIS and connect it to different behavior Mamdani FIS and priority box as its input FIS variable and display or scope as output FIS variable to check the results as shown in the Fig. 9.7. The main benefit of the hierarchically organized decision structure is the evaluation of the approximate performance of the system based on the behaviors of each component of the product.

9.5 Adaptive Neuro-Fuzzy Inference System

This is the second approach for discrete behavior monitoring of a product system using fuzzy logic. Here, the Advantage of Adaptive Neuro Fuzzy Inference System (ANFIS) is that it has the potential to improve system behavior by providing precise values.

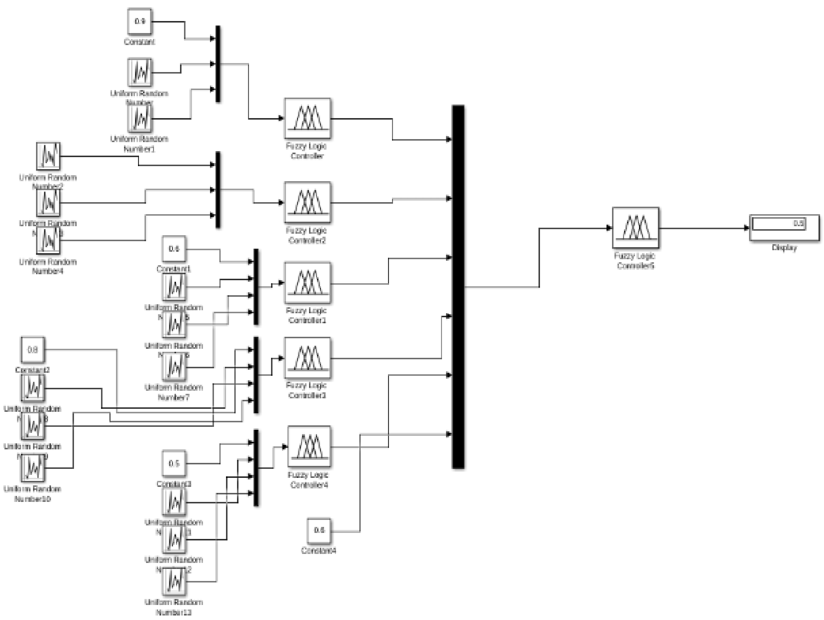


Figure 9.7: Discrete System Behavior using Mamdani FIS

To start with the Adaptive Neuro FIS, it is mandatory to convert Mamdani's FIS into Sugeno FIS. In the context of this work, the author used constant type output FIS variables and triangular-shaped membership function type input FIS variable. Also, the number of rules must be equal to the number of output ANFIS variables and defined by using the following table:

Input 1	Input 2	Input 3	Output
WS	WS	WS	Fail
AS	AS	AS	Average
BS	BS	BS	Excellent

Table 9.5: General table to define rules using soft computing

Above mentioned configuration(FIS variables and rules) are recommended otherwise, the system can generate errors. The next step is to obtain the input and output values from the product in the physical environment and store it into a file. Here, output values are assigned based on the input values. In the context of this work, the author used a different combination of input values(depends on functionality) to obtain corresponding behavior as output value and drone with the camera as a product in the physical environment. In the training period, a different combination of input values from the physical environment can be modified until the best scenario input values can be reached. As already mentioned, some variables are static while others are dynamic, ANFIS trains only the data set of a dynamic variable. Therefore, the main data sets for the ANFIS training are the dynamic variables as static variables have been chosen by the customers according to their requirements or budget. The parameters associated with dynamic variables membership function changes throughout the learning process until it can reach to best scenario output in the physical environment. Error measure = (data set of Best scenario)^2 − (data set of average scenario/worst scenario)^2. The model error for the checking data set tends to decrease as the training takes place up to the point that over-fitting begins. Here, model error is inversely proportional to system performance.

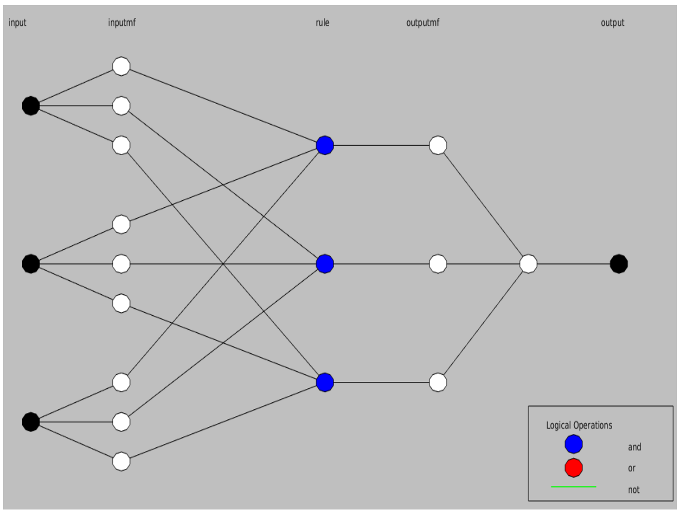


Figure 9.8: Model Structure for particular behavior

The final step is to open neuro-fuzzy designer GUI from the Sugeno FIS and load the file of physical environment values. As shown in Fig. 9.8, The model Structure of a particular behavior is generated in neuro-fuzzy GUI. Here, black nodes are dynamic and static input FIS is variables, connected white nodes are their member functions. These member functions are the

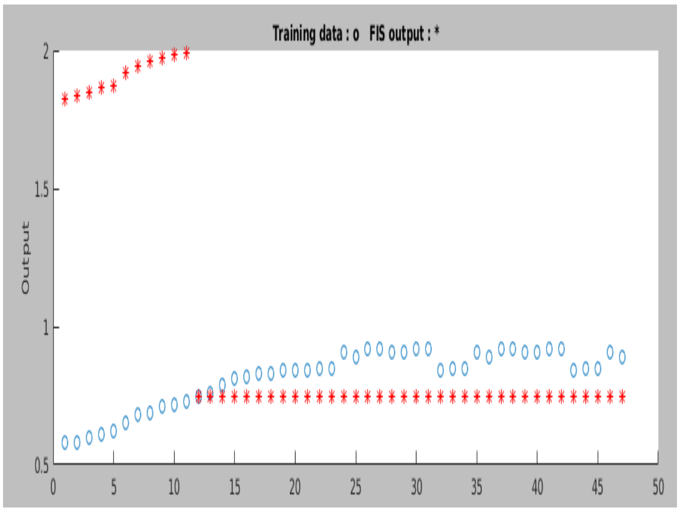


Figure 9.9: Test the FIS for particular behavior using anfis

Worst Scenario (WS), Acceptable Scenario (AS), and Best Scenario (BS). Blue nodes are the rules. White nodes connected to rules are the output membership function and finally output FIS variable is the black node, which is equivalent to expected behavior. Then, Train the FIS either by using the Back Propagation algorithm or Hybrid Algorithm (a combination of Back Propagation and Least-Squares Algorithm) depends on the efficiency. Finally, Test the FIS against the training data as shown in Fig. 9.9. As seen in the graph, some values of training data merged with FIS output variables. These values replace the values of FIS variables, which are far from training data. Now, repeat the experiment with the new FIS variables and check the graph until all FIS variables merged with trained data. Similarly, all the Mamdani's FIS, equivalent to the certain behavior, are converted into Sugeno FIS and corresponding ANFIS and their values can be evaluated and obtained.

9.5.1 Evaluating Discrete System Behavior

The final step to evaluate the system behavior using the behavior information obtained from the different ANFIS. It is important to note that each ANFIS provides the performance of behavior corresponds to the functionality of the product system. After optimizing, testing and obtaining the required values of behavior from a given ANFIS, they are selected as training data for evaluating the system behavior and stored in the file. Similarly, values obtained from the different behaviors of the product system can be considered as training data and stored in the same file. The values obtained from evaluating the different behavior can be considered as input ANFIS variables while output ANFIS variables can be assigned equivalent to the input ANFIS variables. Follow the same steps from the previous section, convert Mamdani's FIS of system behavior into Seguno FIS. Then, open the Neuro-Fuzzy designer GUI from the Sugeno FIS. As shown in Fig. 9.10, black nodes are behavior obtained from different ANFIS, connected white nodes are their member functions.

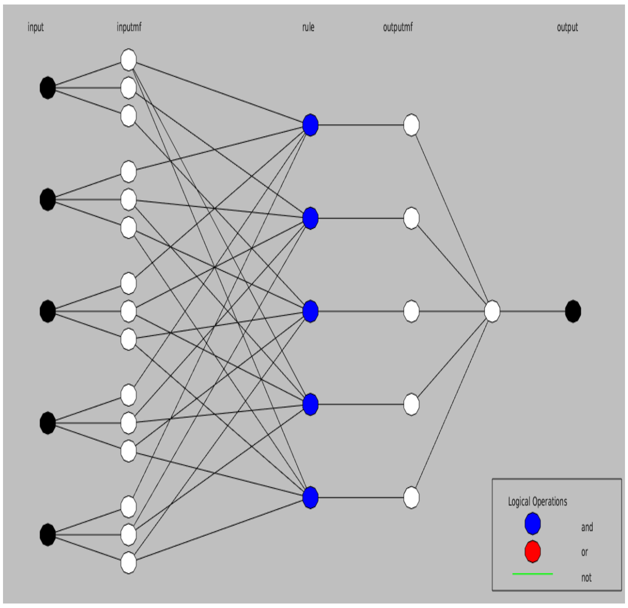


Figure 9.10: Model Structure for system discrete behavior using ANFIS

Finally output ANFIS variable is the black node, which is equivalent to system behavior. Test the FIS against the training data until the expected values of system behavior are obtained.In Fig. 9.11, output FIS variable has similar values of trained data. It is clear that at these points, the system behaves efficiently.

9.6 Comparison Between the Fuzzy Inference Systems

Both FIS has its merits and demerits. It depends on the type of production system and situation. Mamdani's FIS can be applicable in the broader scenario while Adaptive Neuro FIS can be the application for the narrower scenario of a product system. This is because rules constructed in Mandani

FIS can consider all the possible situations of functionality to obtain corresponding behavior while Adaptive Neuro FIS can consider the only specific situation. Also, Mamdani's FIS can be applicable when some behaviors are more important than others to define the product system behavior while in the case of Adaptive Neuro FIS, all behavior has equal weight. On the other hand, Adaptive Neuro FIS can be applicable for the product system when plenty of data obtained from the physical environment needs to be optimized such that the product can behave efficiently in the successive iteration. It can optimize the data by back-propagation or hybrid algorithm whereas Mamdani's FIS can be applicable for limited data input and there are no ways to optimize the data.

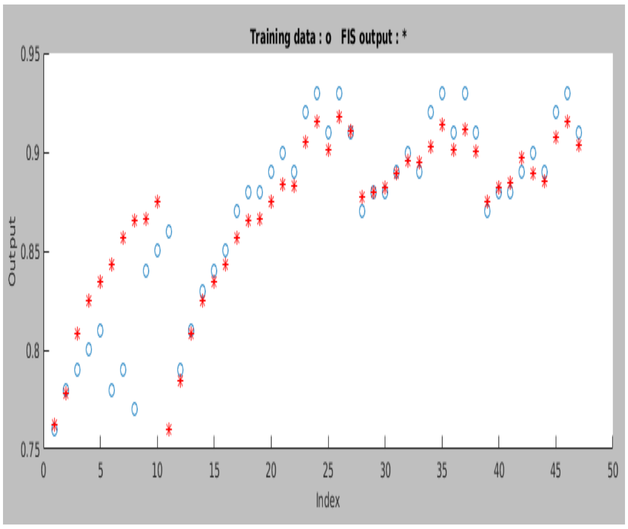


Figure 9.11: Test the FIS for the system discrete behavior

9.7 Conclusion

I have proposed a discrete behavior evaluation of a multidisciplinary product in the simulated environment by using the fuzzy logic & soft computing. Here, behavior models of IC are used to prioritize the behaviors of a multidisciplinary product. Then, discrete system behaviors of the RFLP structure are analyzed by the Matlab toolbox like Mamdani FIS and Adaptive Neuro FIS. Adaptive Neuro FIS can be considered a good approach for critical systems as feedback is obtained from the real-time physical modeled product. In the successive iteration, physical product behaves as per the virtual product if all the functionalities are perfect. Also, a product system can behave efficiently by using back-propagation or hybrid algorithms while Mamdani

FIS can be considered as a good approach for the general system where some functions are more important than others and define the system behavior. This is done by a different combination of rules. It is true that soft computing is not the best approach for analyzing all types of behavior in a product system, but it is a promising approach in the area of product modeling. This chapter covers the Thesis Group 4.

Thesis Group 4

Thesis group 4: Discrete System Behavior Evaluation of the Multidisciplinary Product Model using Fuzzy Logic

Thesis 4
I have introduced Requirement, Functional and Logical Blocks in the Requirement Functional Logical Physical (RFLP) structure for discrete behavior evaluation of a multidisciplinary product by using the fuzzy logic [Y6]. Indeed, the application of the fuzzy set theory is applied to the evaluation performance system [70]. In terms of product modeling, the Fuzzy logic strategy was used for solving multidisciplinary design optimization problems [71]. In this research work, the system discrete behaviors evaluation of the multidisciplinary product is done by using fuzzy logic. The discrete system behaviors of the multidisciplinary product are analyzed by the Matlab toolbox, Mamdani FIS, and Adaptive Neuro FIS. Adaptive Neuro FIS can be considered a good approach for critical systems as feedback is obtained from the real-time physical modeled product while Mamdani's FIS can be considered as a good approach for a general system where some functions are more important than others and define the system behavior. The proposed blocks based on the fuzzy logic can be used in the development of intelligent systems for decision making during the behavioral modeling. It could be applicable in the initial phase of the product modeling to check the feasibility as accuracy is the drawback. As, fuzzy logic is not the best approach for analyzing all types of behavior in a product system, but it is a promising approach in the area of product modeling.

Relevant own publications pertaining to this thesis group: [Y6].

Chapter 10

Conclusion

This dissertation presented Behavioral modeling and Behaviors representation of the Multidisciplinary product using the RFLP structure. The thesis is divided into four groups which are strengthened by case studies.
The First thesis group discusses the structured representation and organization of the process activities of the Multidisciplinary Product model by organizing the Engineering Object (EO)s in the Information Content (IC). As the structure of a multidisciplinary product model is formal so that the causes and characteristics of connections are hard to reveal at the development or revision of an existing structure [1] and management of the high number of changes of modeled engineering objects and representation of background of modeled information in product models [9]. Therefore, I have introduced the concepts of Community zone and Community diagram in the IC for the structured organization of the multidisciplinary product model. Hence, an engineer can evaluate the multidisciplinary product model relevant to its discipline. Also, there are critical issues occur for the effective assistance of decision making in product modeling as mentioned in the paper [10]. Therefore, I have introduced the Process plane in the Information Content (IC) to organize the process activities for effective decision making during the multidisciplinary product modeling. This plane provides an effective decision methodology for representing the behaviors of the multidisciplinary product. It is the building block of this research work. The outcomes of this subthesis are used for the behavioral modeling and behaviors representation of the multidisciplinary product. The author makes an effort for simplified representation as well as effectively analyzes the aspects of a multidisciplinary product using the proposed research work.
The Second thesis group proposes the Behavioral modeling and Behaviors representation of the multidisciplinary product and multidisciplinary Cyber Physical System (CPS). In the paper [38], Behavior based models with intel-

ligent content were emphasized, where feature models for specification and knowledge representations were conceptualized. It is based on the Classical Product Model (CPM) and has limited knowledge to simulate the behavior of the modeled objects. Whereas, this research work proposes the behavior models for the multidisciplinary product with Behavior content and Discipline content based on the Info-Chunk Entities. I have proposed entities called Info-Chunk in the Functional and Logical of the Requirement Functional Logical Physical (RFLP) structure for the behavioral modeling of the multidisciplinary product. They are able to simulate the behavior of the modeled objects. The conceptual models are used to guide the engineer for the simplified representation of the multidisciplinary product to take the correlating decisions. In the paper [45], the Average behavior of the input-output signals of the switched reluctance generator has been reproduced required for system level analysis of the aircraft power distribution system. Considering it as a base, this research work proposes the representation of the behavior of a multidisciplinary product, which is based on the analysis, contextual connection, and optimization activities. Therefore, I have proposed Info-Chunk objects in the Behaviors and Contexts layer of the Multilevel Abstraction based Self-Adaptive Definition (MAAD) structure. The rules and logic are defined as per the Process plane of the Information Content. As mentioned in the paper [66], there is a need for a definition of model entities and contextual connections with the demand of implementation in the RFLP structured system based industrial model using configuration, structuring, object definition [31] and programming capabilities for Cyber Physical System (CPS) systems. Therefore, I have proposed Info-Chunk entities concepts in the extended Engineering Model System (EMS) and Info-Chunk objects concepts in the Active Information Content (AIC) for the behavior representation of the multidisciplinary CPS. This research work focus on the Information Content based EMS and AIC. Here, communication between the Information Content, proposed LiC entities and CPS entities are explained. Based on the experienced situations, values of Info-Chunk can be changed so that the system can be operated accurately according to the situation. Also, communication between AIC, Info-Chunk objects and cyber units is done dynamic and state logic behavior. The author made an effort to represent behaviors efficiently in CPS modeling through Info-Chunk entities and objects.

The Third thesis group focus on the practical approach for the representation of the behavior of the multidisciplinary product using the InfoChunkLib Application Programming Interface (API), Information Content (IC) web application and Content web server. Also, an active approach for the Behavioral modeling of the multidisciplinary product using the Information Content (IC) and Intelligent Property (IP). As compared to the research work [45],

where average behaviors of a product model have been implemented in a virtual test bench and its response by the real system. In this research work, behaviors of a multidisciplinary product are represented through web technologies. It is the practical approach to represent the zone information and extracted modeled behavior data of a multidisciplinary product. The IC web application is based on "My 3D Software". Also, In the paper [38], Feature definition in case of agent based active model is outlined. In this research work, Info-Chunk objects are used for the active models as it stores the detailed information of the modeled behaviors data from the functional and logical layer of the RFLP structure based on the disciplines. Further, the active knowledge based model is used in the IC for the behavior modeling of a multidisciplinary product and can be accessed and updated remotely by the cloud. The proposed method can be considered as an extension of collaborative engineering. For the flexibility of the IC, there are various scenarios considered for interaction between the IC application and the multidisciplinary product application.
The Fourth thesis group focuses on the Fuzzy logic concepts in the system behaviors evaluation of a Multidisciplinary product model. Indeed, the application of the fuzzy set theory is applied to the evaluation performance system [70]. In terms of product modeling, the Fuzzy logic strategy was used for solving multidisciplinary design optimization problems [71]. In this research work, system behaviors evaluation of the multidisciplinary product is done by using fuzzy logic. I have proposed Requirement, Functional and Logical Blocks in the RFLP structure for discrete behavior evaluation of a multidisciplinary product by using the fuzzy logic. The discrete system behaviors of the RFLP structure are analyzed by the Matlab toolbox, Mamdani FIS, and Adaptive Neuro FIS. Adaptive Neuro FIS can be considered a good approach for critical systems as feedback is obtained from the real-time physical modeled product while Mamdani's FIS can be considered as a good approach for a general system where some functions are more important than others and define the system behavior. Indeed, fuzzy logic is not the best approach for analyzing all types of behavior in a product system, but it is a promising approach in the area of product modeling.
During the Ph.D. studies, the author made an effort in other areas as well as proposes an algorithm for future internet considering the Product Lifecycle Management (PLM) [Z1], Predicting the future Using Web Knowledge [Z2] and Gender Prediction of European Schools Principal Using Machine Learning [Z3].

Chapter 11

Application Possibilities and Implementation Issues

The proposed concepts of Community zones, Community diagram, Process plane, Info-Chunk entities, Info-Chunk objects, Discipline-based content, Behavior-based content and InfoChunkLib API in the Information Content (IC) for the Multidisciplinary product modeling could be implemented in the form of the Web application. It is important to note that the behaviors representation and behavioral modeling of the multidisciplinary product proposed in this research work are on the conceptual level. The author proposed the practical approach based on the white papers of the RFLP structure of Engineering systems, CATIA V6 [72] and 3DEXPERIENCE [4]. These systems are using the Modelica [57] component in the Logical level of the RFLP structure. The Modelica component is coded by using the Modelica programming language. There is a possibility to code IC based web applications with similar principles. Therefore, there could be some implementation issues. There could be various scenarios to the handle the multidisciplinary product model through the IC based web application like Multidisciplinary product application and Engineering software is on the local machine and IC application is on the web server, Multidisciplinary product application and engineering software is on the one web server and IC application is on the other web server, Multidisciplinary product application and IC application are on the same server, IC application is accessible through an interface from the multidisciplinary product application. Further, the IP level and Process plane of IP are not defined yet. The behavior representation for IP is the topic of future work. In the case of CPS modeling, the definition of objects and code an API of CPS modeling based on the in the library of Modelica is the topic of future work.

Own Publications Pertaining to Theses

[Y1] Yatish Bathla, "Structured organization of engineering objects in the information content of plm system," in *International Symposium on Applied Computational Intelligence and Informatics (SACI)*, vol. 11. IEEE, May 2016, pp. 473 – 478.

[Y2] Y. Bathla, "Different types of process involved in the information content product model," in *International Symposium on Intelligent Systems and Informatics (SISY)*, vol. 14. IEEE, August 2016, pp. 99–104.

[Y3] Yatish. Bathla, "Conceptual models of information contents for product modeling," *Acta Polytechnica Hungarica*, vol. 15, no. 2, pp. 169–188, 2018.

[Y4] Yatish Bathla, "Info-chunk objects as new behavior representation for system based model of product," *Acta Polytechnica Hungarica*, vol. 16, no. 4, pp. 87–112, 2019.

[Y5] Yatish Bathla and Szenasi Sandor, "Active model of the multidisciplinary product in the cad system using the oop concepts," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 1, pp. 292–298, 2019.

[Y6] Yatish Bathla, Marta Takacs, "Evaluating product system behavior using soft computing in product structure modeling," in *International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, vol. 15. IEEE, Jan 2017, pp. 000 307 – 000 312.

[Y7] Yatish Bathla, "Info-chunk driven rflp structure based product model for multidisciplinary cyber physical systems," in *International Symposium on Intelligent Systems and Informatics (SISY)*, vol. 16. IEEE, Sep. 2018, pp. 327–332.

[Y8] Y. Bathla, "Era of behavioral modeling in product lifecycle management system: State of the art survey," in *International Symposium on Applied Computational Intelligence and Informatics (SACI)*, vol. 14. IEEE, August 2018, pp. 229–234.

[Y9] Yatish Bathla, Mahmod Al-Bkree, "Web application for 3d software comparison considering plm system," *HADMÉRNOK*, vol. 13, no. 1, pp. 302–313, 2018.

[Y10] Y. Bathla and Szenasi Sandor, "A web server to store the modeled behavior data and zone information of the multidisciplinary product model in the cad systems," *Journal of Informatica*, Accepted.

Own Publications Not Pertaining to Theses

[Z1] Yatish Bathla, "Community based routing scheme for future internet considering plm systems," in *2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Jan 2016, pp. 205–210.

[Z2] A. Mosavi, Y. Bathla, and A. Varkonyi-Koczy, "Predicting the future using web knowledge: State of the art survey," in *Recent Advances in Technology Research and Education*, D. Luca, L. Sirghi, and C. Costin, Eds. Cham: Springer International Publishing, 2018, pp. 341–349.

[Z3] Y. Bathla, C. Verma, and N. Kumar, "Smart approach for real-time gender prediction of european school's principal using machine learning," in *Proceedings of ICRIC 2019*, P. K. Singh, A. K. Kar, Y. Singh, M. H. Kolekar, and S. Tanwar, Eds. Cham: Springer International Publishing, 2020, pp. 159–175.

Bibliography

[1] Aniko Szakal, "Application of product model for engineering process definition," in *International Conference System Science and Engineering(ICSSE)*, vol. 4. IEEE, Nov 2011, pp. 66–71.

[2] Ulf Sellgren and Raphafil Drogou, "Behavior modeling in mechanical engineering - a modular approach," in *Engineering with Computers*, vol. 14, no. 3. IEEE/ACM International Workshop, September 1998, pp. 185–196.

[3] Szykman, S., R.D. Sriram, C. Bochenek, and J. Senfaute, "Design repositories:next-generation engineering design databases," in *IEEE Intelligent Systems and Their Applications*. IEEE, Feb 2000, pp. 1 – 18.

[4] Frederic Chauvin, Gauthier Fanmuy, "Systems engineering on 3dexperience platform - uas use case," in *Dassault systèmes*. Dassault systèmes, 2017, pp. 1 – 14.

[5] Horvath Laszlo, "Supporting lifecycle management of product data by organized descriptions and behavior definitions of engineering objects," in *Journal of Advanced Computational and Intelligent Informatics*, vol. 11, no. 9, 2007, pp. 1107–1113.

[6] Imre J. Rudas, Horvath L., "Systems engineering methods for multidisciplinary product definition," in *International Symposium on Intelligent Systems and Informatics (SISY)*, vol. 12. IEEE, Nov 2014, pp. 293 – 298.

[7] F.C.T.van der Helm, "Analysis of the kinematic and dynamic behavior of the shoulder mechanism," *Journal of Biomechanics*, vol. 27, no. 5, pp. 527 – 550, 1994.

[8] John Stark,, *Product Lifecycle Management: 21st Century Paradigm for Product Realisation.* 3, illusztrált: Springer, 2015.

[9] Imre. J. Rudas, L. Horvath, "Towards the information content driven product model," in *System of Systems Engineering.* IEEE, June 2008, pp. 1–6.

[10] I. J. Rudas, Horvath, L., "Bringing up product model to thinking of engineer," in *International Conference Systems, Man and Cybernetics.* IEEE, Oct 2008, pp. 1355 – 1360.

[11] Laszlo Horvath, Imre J. Rudas, Marta Takacs, "Content representation structure for product system modeling," in *Intelligent Engineering Systems (INES)*, vol. 19. IEEE, Sep 2015, pp. 85 – 90.

[12] L. Horváth, "New methods on the way to intelligent modeling in computer integrated engineering," in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society.* IEEE, Nov 2010, pp. 1359–1364.

[13] Ray Y. Zhong, Xun Xu, Eberhard Klotz, Stephen T. Newman, "Intelligent manufacturing in the context of industry 4.0: A review," *The Engineering*, vol. 3, no. 5, pp. 616–630, 2017.

[14] Laszlo Horvath, Imre J. Rudas, "Systems engineering in product definition," in *Applied Machine Intelligence and Informatics (SAMI)*, vol. 13. IEEE, Jan 2015, pp. 181 – 186.

[15] I. J. Rudas, Horvath, L, "Multilevel abstraction based self control method for industrial plm model," in *International Conference on Industrial Technology (ICIT).* IEEE, June 2014, p. 695 – 700.

[16] Lazlo Horvath, "New method for definition of organized driving chains in industrial product model," in *International Conference on Industrial Technology (ICIT 2017).* IEEE, 2017, pp. 1183–1188.

[17] Mocko, G. and S. Fenves, "A survey of design-analysis integration issues," *National Institute of Standards and Technology*, pp. 1 – 47, 2003.

[18] I. J. Rudas, Horvath, L, "Product definition by requested behavior," in *International Conference on Intelligent Engineering Systems (INES),.* IEEE, June 2013, pp. 249 – 254.

[19] I. R. Horvath, Laszló, "Content driven generation of rflp structured engineering system representation," in *World Automation Congress (WAC).* IEEE, July 2016, pp. 1–6.

[20] L. Horváth, "A new method for enhanced information content in product model," *WSEAS Transactions on Information Science and Applications*, vol. 5, no. 3, pp. 277 – 285, 2008.

[21] Horvath, L., "New design objective and human intent based management of changes for product modeling," *Acta Polytechnica Hungarica*, vol. 4, no. 1, pp. 17 – 30, 2007.

[22] R. Sudarsan, S.J. Fenves, R.D. Sriram and F. Wang, "A product information modeling framework for product lifecycle management," in *Computer-Aided Design*, vol. 37, no. 13. Elsevier, June 2011, pp. 1399 – 1411.

[23] R. Jardim-Goncalves, N. Figay and A. Steiger-Garcao, "Enabling interoperability of step application protocols at meta-data and knowledge level," in *International Journal of Technology Management*, vol. 36, no. 4, June 2006, pp. 402–421.

[24] Michael J. Prat, "Introduction to iso 10303 - the step standard for product data exchange," in *Journal of Computing and Information Science in Engineering*, vol. 1, no. 1, June 2001, pp. 102–103.

[25] I. J. Rudas, Horvath, L., "Control of product object definition by background of human influence,," in *International Conference Intelligent Engineering Systems (INES)*, vol. 15. IEEE, June 2011, pp. 45 – 50.

[26] Imre J. Rudas, Horvath, L., "New product model representation for decisions in engineering systems," in *International Conference System Science and Engineering(ICSSE)*. IEEE, June 2011, pp. 546 – 551.

[27] L. Horvath, I. J. Rudas, "Active driving content in rflp structured product model," in *Recent Advances on Mechanics, Materials, Mechanical Engineering and Chemical Engineering*, pp. 123–131.

[28] Laszlo Horvath, Imre J. Rudas, "Content driven generation of rflp structured engineering system representation," in *World Automation Congress (WAC)*. IEEE, July 2016, pp. 1 – 6.

[29] L. Horvath, Imre J. Rudas, "Product definition using a new knowledge representation method," in *Computational Intelligence in Engineering*. Springer, August 2010, pp. 273 – 285.

[30] Laszlo Horvath, Imre Rudas, "Multilevel abstraction based self control method for industrial plm model," in *Industrial Technology(ICIT)*. IEEE, Feb 2014, pp. 695 – 700.

[31] Horvath, L., "Intelligent property support for cyber-physical product system modeling," in *World Automation Congress (WAC)*. IEEE, Oct. 2017, pp. 3474–3479.

[32] PTC, "White paper behavioral modeling the next generation of mechanical design automation." PTC, 2000, pp. 1–14. [Online]. Available: https://learn.lboro.ac.uk/ludata/cd/cad/behavioural_mod.pdf

[33] Rajarishi Sinha and Christiaan.J.J. Paredis and Pradeep.K Khosla, "Integration of mechanical cad and behavioral modeling," in *Behavioral Modeling and Simulation*. IEEE/ACM International Workshop, Oct. 2000, pp. 31–36.

[34] U.Roy and B.Bharadwaj, "Design with part behaviors: behavior model, representation and applications," in *Computer-Aided Design*, vol. 34, no. 9. Elsevier, June 2002, pp. 613–636.

[35] Horvath Laszlo, Imre J. Rudas, "Intelligent computer methods in behavior based engineering modeling," in *Systems, Man and Cybernetics*. IEEE, October 2002, pp. 31–36.

[36] Laszlo Horvath, Imre J. Rudas, Aniko Szakal, "Behavior and petri net based techniques for intelligent engineering modeling," in *1st Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence*, vol. 1. IEEE Hungary Section, Feb 2003, pp. 25–40.

[37] Laszlo Horvath, Imre J. Rudas, Gerhard Hancke, "Behavior model assisted decisions in active modeling supported engineering," in *1st Romanian-Hungarian Joint Symposium on Applied Computational Intelligence*, vol. 1. SACI 2004, June 2004, pp. 291–301.

[38] Laszlo Horvath, Imre J. Rudas, "Behavior and Design Intent Based Product Modeling," in *Acta Polytechnica Hungarica*, vol. 1, no. 2, Nov 2004, pp. 17–34.

[39] Horvath Laszlo, Imre J. Rudas, "New approach to knowledge intensive product modeling in plm systems," in *Systems, Man and Cybernetics, ISIC*. IEEE, October 2007, pp. 668–673.

[40] Huichao Sun, Rémy Houssin, Mickael Gardoni, François de Bauvrond, "Integration of user behaviour and product behaviour during the design phase:software for behavioural design approach," in *International Journal of Industrial Ergonomics*, vol. 43, no. 1. Elsevier, Jan 2013, pp. 100–114.

[41] Zhijia Xu and Jie Zhang and Yuan Li and Shoushan Jiang and Yuanliang Sun, "Product modeling framework based on interaction feature pair," in *Computer-Aided Design*, vol. 45, no. 12. Elsevier, March 2013, pp. 1591–1603.

[42] L. Horvath, I.J. Rudas, "Situation based feature definition on product behavior level," in *Applied Computational Intelligence and Informatics (SACI)*, vol. 13. IEEE, May 2013, pp. 245–250.

[43] Laszlo Horvath, Imre Rudas, "Elevated level design intent and behavior driven feature definition for product modeling," in *Industrial Electronics Society, IECON 2013*. IEEE, Nov 2013, pp. 4374–4379.

[44] L. Horvath, I.J. Rudas, "Requirement based behavior driven product feature generation," in *Recent Advances in Intelligent Control, Modelling and Simulation: Proceedings of the 2nd International Conference on Intelligent Control, Modelling and Systems Engineering*, vol. 2. WSEAS Press, Jan 2014, pp. 139–144.

[45] Virgilio Valdivia, Rebecca Todd, Frank J. Bryan, Andrés Barrado, Antonio Lázaro, Andrew J. Forsyth, "Behavioral modeling of a switched reluctance generator for aircraft power systems," in *Computer-Aided Design*, vol. 61, no. 6. IEEE Industrial Electronics Society, June 2014, pp. 2690–2699.

[46] Lazlo. Horvath, Imre. Rudas, "New method for behavior driven product concept definition," in *Intelligent Engineering Systems (INES)*, vol. 18. IEEE, September 2014, pp. 139–144.

[47] L.Horvath, J.Fodor, I.J.Rudas, "Manufacturing aspect of the ibca structure for active knowledge content representation in product model," in *International Federation of Automatic Control*, vol. 48, no. 3. Elsevier, 2015, pp. 1616–1621.

[48] Horváth, László and Rudas, Imre J, "Intelligent content for product definition in rflp structure," in *Intelligent Software Methodologies, Tools and Techniques*. Springer International Publishing, 2015, pp. 55–70.

[49] I. J. Rudas, Horvath L., "Modeling and Analysis of Requests, Behaviors, and Actions for RFLP Structure," in *IEEE International Conference Systems, Man, and Cybernetics (SMC)*. IEEE, 2015, pp. 725 – 730.

[50] Horvath, László, "Representation of intellectual property for multidisciplinary industrial engineering model system," in *Industrial Electronics (ISIE).* IEEE, June 2016, pp. 328–333.

[51] L. Horváth, "New method for enhanced driving of entity generation in rflp structured product model," in *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, June 2017, pp. 541–546.

[52] Patricia Derler, Edward A. Lee, Alberto Sangiovanni Vincentelli, "Modeling cyber–physical systems," in *Proceedings of the IEEE*, vol. 100, no. 1. IEEE, Jan 2012, pp. 13–28.

[53] Qiang Wang, Gang Yang, Xingshe Zhou, Yalei Yang, "Behavior modeling of cyber-physical system based on discrete hybrid automata," in *Computational Science and Engineering (CSE)*, vol. 16. IEEE, Dec 2013, pp. 680–684.

[54] Kamaldeep Kaur, L.T. JayPrakash, "Modeling and simulation of cyber physical systems: A case of an adaptive water management system," in *Confluence The Next Generation Information Technology Summit*, vol. 5. IEEE, September 2014, pp. 267–273.

[55] M. Thorup and U. Zwick, "Compact routing schemes," in *SPAA 01 Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures*, vol. 13. ACM Press, June 2001, p. 1 – 10.

[56] Christof Ledermann, Claus Hanske, Jrg Wenzel, Paolo Ermanni, Roland Kelm, "Associative parametric cae methods in the aircraft pre-design," *Elsevier Aerospace Science and Technology*, vol. 9, no. 7, pp. 641 – 651, 2005.

[57] P. Fritzson, *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*, 2nd ed., ser. IEEE Press. Hoboken, NJ: Wiley, 2015.

[58] S. Kleiner and C. Kramer, "Model based design with systems engineering based on rflp using v6," in *Smart Product Engineering*, M. Abramovici and R. Stark, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 93–102.

[59] Y. Ou, "On mapping between uml and entity-relationship model," in *The Unified Modeling Language*, M. Schader and A. Korthaus, Eds. Heidelberg: Physica-Verlag HD, 1998, pp. 45–57.

[60] B. Thalheim, *Entity-Relationship Modeling: Foundations of Database Technology*, 1st ed. Berlin, Heidelberg: Springer-Verlag, 2000.

[61] Y. Li, M. Hedlind, and T. Kjellberg, "Implementation of kinematic mechanism data exchangebased on step," 2011.

[62] Dimitris Bertsimas, John N. Tsitsiklis, "Introduction to linear optimization," *Athena Scientific, Belmont, Massachusetts*, vol. 1, pp. 81 – 137, 1997.

[63] L. Horváth, I. J. Rudas, and G. Hancke, "Behavior model assisted decisions in active modeling supported engineering."

[64] L. Horvath, I. J. Rudas, and G. P. Hancke, "Associative modeling of machining processes using feature based solid part models," in *2000 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies*, vol. 2, 2000, pp. 1267–1273 vol.2.

[65] I. Sommerville, *Software Engineering*, 9th ed. USA: Addison-Wesley Publishing Company, 2010.

[66] Laszlo Horvath, Imre J. Rudas, "Engineering modeling for cyber physical systems," in *World Symposium on Applied Machine Intelligence and Informatics SAMI*, vol. 16, no. 2. IEEE, Feb 2018, pp. 000 207–000 212.

[67] L. Horvath, Andrea Serester, "Content driven engineering model system for cyber physical systems," in *International Symposium on Applied Computational Intelligence and Informatics*, vol. 12, no. 2. IEEE, May 2018, pp. 367–372.

[68] Zivkovic Zivan, Nikolic Djordje, Djordjevic Predrag, Mihajlovic Ivan, Savic Marija, "Analytical network process in the framework of swot analysis for strategic decision making (case study: Technical faculty in bor, university of belgrade, serbia)," *Elsevier Aerospace Science and Technology*, vol. 12, no. 7, pp. 199 – 2016, 2015.

[69] P. J. Nikumbh, S.K. Mukhopadhyay, Bijon Sarkar and Ajoy Kumar Datta, "Product-mix optimization through soft computing," in *CIE International Conference*. IEEE, July 2009, pp. 443 – 448.

[70] A. Shaout and M. Al-Shammari, "Fuzzy logic modeling for performance appraisal systems: A framework for empirical evaluation," *Expert Systems*

with Applications, vol. 14, no. 3, pp. 323 – 328, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417497000857

[71] A. Benaouali and S. Kachel, “Multidisciplinary design optimization of aircraft wing using commercial software integration,” *Aerospace Science and Technology*, vol. 92, pp. 766 – 776, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963818324131

[72] D. S. AB, *CATIA V6 Introduction Buggy Case Study*, 2007.