Conceptual modelling for integrated decision-making in process systems

Canan Dombayci

ADVERTIMENT La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del repositori institucional UPCommons (http://upcommons.upc.edu/tesis) i el repositori cooperatiu TDX (http://www.tdx.cat/) ha estat autoritzada pels titulars dels drets de propietat intel·lectual **únicament per a usos privats** emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei UPCommons o TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a UPCommons (*framing*). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del repositorio institucional UPCommons (http://upcommons.upc.edu/tesis) y el repositorio cooperativo TDR (http://www.tdx.cat/?locale-attribute=es) ha sido autorizada por los titulares de los derechos de propiedad intelectual **únicamente para usos privados enmarcados** en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio UPCommons No se autoriza la presentación de su contenido en una ventana o marco ajeno a UPCommons (*framing)*. Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the institutional repository UPCommons (http://upcommons.upc.edu/tesis) and the cooperative repository TDX (http://www.tdx.cat/?locale-attribute=en) has been authorized by the titular of the intellectual property rights **only for private uses** placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading nor availability from a site foreign to the UPCommons service. Introducing its content in a window or frame foreign to the UPCommons service is not authorized (framing). These rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.

CONCEPTUAL MODELLING FOR INTEGRATED DECISION-MAKING IN PROCESS SYSTEMS

CONCEPTUAL MODELLING FOR INTEGRATED DECISION-MAKING IN PROCESS SYSTEMS

Canan Dombayci
Chemical Engineer, M.Sc

A Thesis presented for the degree of Doctor of Philosophy
Directed by Prof. Dr. Antonio Espuña

Center for Process Systems and Environmental Engineering - CEPIMA
Department d'Enginyeria Quimica - DEQ
Universitat Politecnica de Catalunya - UPC

Barcelona, January 2019

Aileme ...
A mi familia ...

Summary

This Thesis addresses the systematic construction of Decision Making Models (DMMs) from the conceptualization stage to its application in specific situations, with special emphasis on the treatment of scenarios where there is a hierarchy of decision levels, common in the Process Systems (PS). Although the methodologies developed are generic, the scope of this Thesis is limited to the perspective of Process Engineering.

The central component required to construct a DMM is the conceptual description of the reality, which supports the systematisation of management procedures. During this description, two different domains can be identified: the PS Domain, useful to describe the structure of the process as such (physical reality and the way in which its elements are related), and the Management Domain, identified in this Thesis as associated with the Conceptual Constraints (CC) that describe the restrictions associated with the management of the process. In this way, the PS Domain includes concepts and relationships that appear in the control standards of the process followed by the company: the description of the process to be developed, the description of the physical equipment in which it is developed, and that of its interactions, giving rise to the control of the execution of the procedures; this domain should allow managing the construction, design, operation and control of any manufacturing system. On the other hand, the CC Domain contains the information associated with the concepts and relationships that must be fulfilled to ensure a coherent set of decisions, with the purpose of identifying and representing the systematics to follow during the decision-making process, giving rise to the conceptual representation of this system and, finally, the construction of the corresponding DMM.

The first challenge addressed in this thesis is associated with the systematisation of conceptual modelling from semantic information, for the construction of ontologies from textual sources and a procedure to verify the internal coherence of these sources. The application of this methodology has been used for the identification of the essential concepts and relationships in the PS Domain, allowing creating a generic, common and shared model, unlike the existing models. In the next step, this PS Domain has been used to solve management problems in systems that comprise multi-level hierarchies. The resulting decision-making process allows integrating the decisions made at each level, ensuring their consistency

from an approach that simultaneously considers the management of all available information (data and knowledge).

On the other hand, the introduction of the necessary concepts and relationships to ensure the feasibility of the process management decisions, through the CC Domain, allows the development of systematic DMM creation procedures: this domain classifies the constrains (balances, sequence, etc.), adds abstract elements to them (e.g.: produced and consumed amounts) and allows to generalize the relation of its components with the information associated to the PS Domain.

The last part of this Thesis deals with the integration of the PS and CC Domains, and their application for the generation of new decision-making systems. For this, algorithms have been designed that, starting from the previously identified and classified restrictions, and patterns of DMMs also previously identified from existing cases, exploit the information available through the instances in the PS Domain, to generate new DMMs according to the user's specifications. Its use is illustrated through cases from different environments, demonstrating the generalisation capacity of the created systematics.

Resumen

Esta Tesis aborda la construcción sistemática de Modelos para la toma de Decisiones (DMMs) desde la etapa de conceptualización hasta su aplicación en situaciones concretas, con especial énfasis en el tratamiento de escenarios en los que existe una jerarquía de niveles de decisión, habitual en la Industria de Proceso (PS). Aunque las metodologías desarrolladas son genéricas, el alcance de esta Tesis se limita a la perspectiva de la Ingeniería de Procesos.

El componente central requerido para construir un DMMs es la descripción conceptual de la realidad a la que se orienta, que a su vez respalda la sistematización de los procedimientos de gestión. Durante esta descripción, se pueden identificar planteamientos asociados a dos dominios diferentes: el Dominio del Proceso (PS), útil para describir la estructura del proceso como tal (realidad física y forma en la que se relacionan sus elementos), y el Dominio de Gestión, asociado a las Restricciones Conceptuales (CC) que describen las restricciones asociadas a la gestión del proceso. El Dominio PS incluye conceptos y relaciones que aparecen en los estándares de control del proceso que sigue la empresa: la descripción del proceso a desarrollar, la descripción de los equipos físicos en los que se desarrolla, y la de sus interacciones, que dan lugar al control de ejecución de los procedimientos; este dominio debe permitir la construcción, el diseño, la operación y el control de cualquier sistema de fabricación. Por su parte, el Dominio CC contiene la información asociada a los conceptos y las relaciones que deben cumplirse para asegurar un conjunto coherente de decisiones, con el propósito de identificar y representar la sistemática a seguir durante el proceso de toma de decisiones, dando lugar a la representación conceptual de esta sistemática y, finalmente, a la construcción del correspondiente DMM.

El primer reto abordado en esta Tesis está asociado a la sistematización del modelado conceptual a partir de información semántica, para construcción de ontologías a partir de fuentes textuales y de un procedimiento para verificar la coherencia interna de dichas fuentes. La aplicación de esta metodología se ha utilizado para la identificación de los conceptos y las relaciones esenciales en el Dominio PS, permitiendo crear un modelo genérico, común y compartido, a diferencia de los modelos existentes.

En el siguiente paso, este Dominio PS se ha utilizado para la resolución de problemas de gestión en sistemas que comprenden múltiples niveles de jerarquías funcionales. El proceso de toma de decisiones resultante permite integrar las decisiones tomadas en cada nivel, asegurando su coherencia a partir de un enfoque que contempla simultáneamente la gestión de toda la información disponible (datos y conocimiento).

Por su parte, la introducción de los conceptos y relaciones necesarios para asegurar la factibilidad de las decisiones de gestión del proceso, a través del Dominio CC, permite el desarrollo de procedimientos sistemáticos de creación de DMMs: este Dominio clasifica las restricciones (balances, secuencia, etc.), agrega elementos abstractos a dichas restricciones (p.e.: cantidad producida y consumida) y permite generalizar la relación de sus componentes con la información asociada al Dominio PS.

En la última parte de esta Tesis se aborda la integración de los Dominios PS y CC, y su aplicación para la generación de nuevos sistemas de toma de decisiones. Para ello, se han diseñado algoritmos que, partiendo de las restricciones anteriormente identificadas y clasificadas, y patrones de DMMs también previamente identificados a partir de casos ya existentes, explotan la información disponible a través de las instancias del Dominio PS, para generar de nuevos modelos de toma de decisión de acuerdo con las especificaciones del usuario. Su utilización se ilustra a través de casos procedentes de diferentes entornos, demostrando la capacidad de generalización de la sistemática creada.

Acknowledgement

The completion of this thesis has required an enormous amount of work and effort on my part; however, none of this would have been possible without guidance from Dr Antonio Espuña. It has been a pleasure and honour to work with someone so dedicated and committed to his work. I am truly grateful to him for giving me the opportunity to pursue this path and prove myself.

I would also like to thank those whose help has directly contributed to the development of my work. Thank you, Dr Moisès Graells for giving me food for thought; Dr Edrisi Muñoz and Dr Elisabet Capòn-Garcìa, for letting me grow; and Dr Javier Farreres and Dr Horacio Rodríjuez for supporting the growth of the conceptual understanding in my research.

I want to thank to CEPIMA members for being such great inspirations: Dr Luis Puigjaner; Dr Montserrat Perez; Dr Javier Silvente; and last but not least, Dr Marto Benito-Morena.

Thank you, my team-mates, for sharing this journey with me: Dr Sergio Medina being at my back against the waves; Francesca Audino for cleaning clouds by being good; Gicela Lupera; Dr. Hamed Ardakani; Ahmed Shokry; Shabnam Morakabatchiankar; Ana Somoza; Xiangwei Yu; and many others, thank you.

During this period, I received a great deal of support from my friends, thanks for being with me during this journey: S. Merve; Victoria; Dilara; Dr Tuba Ayhan; Dr Seda Erbas Nowag; Servet; Dr Javier Corea Araujo; Pervin; Karen; Gloria; and many others, thank you.

Of course, I could not do much without my family. Please forgive me again for letting this thesis and I steal so much time. My parents, Ikbal and Kamil, thank you for your endless support and admiration. Thank you, my sisters, Sema and Ilgin, for being another part of me. And thank you my little Ataberk for being the hope for the future.

I am grateful to my partner Dr Gerardo Guerra. His support has constantly helped me succeed. I cannot express my gratitude to him for believing in me. And I can only hope to be there for him as he has been there for me all the time.

Finally, I hope that reading this thesis will be (or was) an inspiring experience and that I managed to open new ways of thinking for designing a better future.

Canan Dombayci
Reading, January 2018

Disclaimer

This thesis is submitted as partial fulfilment of the requirements for the Doctor of Philisopy (PhD) degree at the Universitat Politectica de Catalunya (UPC), BarcelonaTech. The PhD project was carried out at the Centre for Process Systems and Environmental Engineering (CEPIMA) at the Department of Chemical Engineering from June 2014 to August 2018 under the guidance of Professor Antonio Espuña as the main supervisor.

Financial support from the Spanish Ministry of Economy and Competitiveness and ERDF (projects SIGERA:DPI2012-37154-C02-01, AIMS: DPI2017-87435-R and ECOCIS: DPI 2013-48243-C2-1-R), and AGAUR (2014-SGR-1092-CEPEiMA, 2017-SGR-950, and grant FI) is fully appreciated.

Contents

III. Decision-Making Models and Conceptualization 117

List of Figures

List of Tables

Part I.

Overview

Chapter 1.

Introduction

This first chapter connects the perspectives from Process Systems Engineering (PSE) and thesis topics on systematic Decision-Making Models (DMMs) building from conceptualisation to the application of methods and considering the integrated multi-level hierarchies found in Process Systems (PS). The core component required to build DMMs is a conceptual description of the reality that supports the systematisation of building procedures. Therefore, domain conceptualisation of the multi-level structure and DMMs are focuses of this thesis. Overall, the methodologies that were developed are designed to achieve systematic coordination and operation of activities in PS, considering knowledge management throughout the multi-level hierarchies.

1.1. Process Systems Engineering Perspective

The term “Process Systems” refers to chemical, physical or biological systems that are designed, built and run to manufacture products (inspired by Marquardt et al. (2010)). PSE may be described as the art of decision-making for PS. PSE deals with the Chemical Engineering disciplines (e.g. design, operation, and control of chemical, physical, and biological processes) in addition to the use of systematic computer-based methods as resources (i.e. conflicting/multi objectives, support tools) (Grossmann and Westerberg, 2000; Klatt and Marquardt, 2009; Glavič, 2012). PS-related questions (i.e. “how to plan?”, “how to design?”, “how to operate?”, and “how to control?” PS) are probably the most critical (Takamatsu, 1983) that engineers use to state PSE problems.

PS can be represented by a hierarchy that is a structure of sets and subsets such that every subset of a set is of lower rank than the set (Williams, 1994). This hierarchy can be any structure consisting of units and subunits in which the subunits are of lower rank than the units involved. In the system view, every hierarchy has vertical interactions of subsystems that contain the entire system in which the term (sub)-system is a transformation of inputs into outputs (Mesarovic et al., 1970). Multi-level hierarchical systems represent this categorised system of PS including interactions among subsystems. Each system may occur as part of the process model, physical model or procedural control model. Hereafter, (unless it is stated otherwise) multi-level hierarchies refer to the representation of PS. This representation is described in Section 2.2.1 in which the topic is extended to decisions and standardisation.

In terms of study topics, Chemical Engineering and PSE have grown in parallel for many decades in some subjects (see in Figure 1.1). These PSE-oriented subjects can be listed as follows:

Process and Product Design: pharmaceutical product design, process synthesis (Cameron and Gani, 2011; Bertran et al., 2017)

Enterprise-wide optimization: supply chain management, risk management, integration of activities in enterprise (Grossmann, 2005; Zhang and Grossmann, 2016)

Production Scheduling: real time scheduling, utility scheduling, recipe management, reactive scheduling(Harjunkoski et al., 2014; Maravelias and Sung, 2009)

Process Control: real time process control, model based process control, process supervision, fault diagnosis (Luyben et al., 1997; Rawlings, 2000; Mayne et al., 2000; Daoutidis et al., 2018)

In order to address these subjects, PSE has been the developer and user of many support tools:

Mathematical Modelling: Mathematical programming, dynamic programming (Williams, 1978; Edgar et al., 1988)

Algorithms: Solvers, decomposition algorithms, genetic and evolutionary algorithms, machine learning Guillén et al. (2006); Benders (1962); Herbrich and Graepel (2012)

Software Applications: AIMMS (AIMMS, 2018), GAMS (GAMS, 2016), Aspen (Aspen, 2018), SuperPro (Intelligen, 2018), Matlab (Matlab, 2018), Python (open source language) (Python, 2018), OntoCep, MOSAIC Kraus et al. (2014), ICAS

The prospective progress of PSE is supported by advances in conceptual understanding. PSE trends include ways to make specific advances, conceptual ways to understand and support the advances and aspects of problem-solving. Examples are the development of efficient or comprehensive mathematical models, the determination of better ways to manufacture products, the most acceptable design of a supply chain, the fastest ways to build models and the safest way to produce products. Thus, these conceptual modelling trends

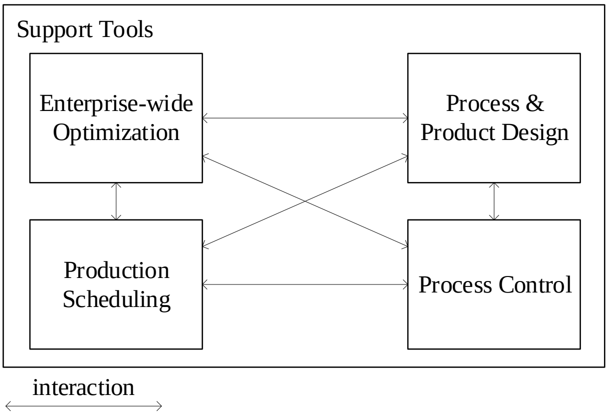


Figure 1.1.: Main subjects of PSE and support tools (symbolic)

in PSE are central to this thesis. One way to check the current state and trends in PSE is to follow up recent meetings on the topic. The Computer-Aided Chemical Engineering[1] (CACE, n.d.) and 2040 Vision of PSE[2] events introduced some current topics. For instance:

- *Data exchange:* data exchange models should be represented "meaningfully" with ontologies to domain's shared conceptualisation. Ontologies are required to support the solution to modelling (e.g. data exchange in the multi-scale modelling) (Henning, 2017*a*).

- *Generic optimisation algorithms and models:* even though part of a system is used separately or changed in the entire system, every component should continue to work without the need for adjustment. For instance, in the case of changes in a process (e.g. a new column or membrane), the code containing the main core should not need to be updated (Ydstie, 2017).

- *"A model for modelling":* mathematical modelling is a language with a syntax and a grammar (an ontological model for mathematical knowledge representation has been presented in (Muñoz et al., 2014)). This mathematical language can be the basis of "a model for modelling" for machine-generated models. The idea of using intelligent systems to capture process knowledge and transform it into equations is crucial. This improvement is expected to help model-building experts by reducing model-revising cycles and allowing model-builders to think more about the process

---

[1] http://www.wcce10.org/index.php/jointevents/escape27

[2] http://stephanopoulos-symposium.mit.edu/

itself instead of spending time on creating mathematical models from scratch (Linninger, 2017).

- *Code generation:* one topic in current research in PSE is automated code generation, either from captured knowledge (Elve and Preisig, 2017) or process models written in generic languages (e.g. Latex and MathML) (Tolksdorf et al., 2017) leading to standard executable instructions that can be processed by an optimiser or a flowsheet environment (e.g. CAPE-OPEN).

Beyond current perspectives, one of the first PSE issues was the importance of "coordinated decisions in the chemical processing units that affect networks globally and usage of the artificial intelligence field" (*The International symposium on process systems engineering*, 1982). Suggestions were to use computer technologies, software and hardware for Computer-Aided Chemical Engineering, and expert systems to overcome computational issues in PSE. Nevertheless, these suggestions should also incorporate the primary challenge for chemical engineers: transformation of chemical process knowledge into a form that is suitable for decision-making (Motard, 1983).

Douglas (1988) presented a hierarchical conceptual modelling approach for building a process model, considering five hierarchical design decisions. These hierarchical decisions are systematised as: (i) batch versus continuous, (ii) input-output structure of the flow-sheet, (iii) recycle structure of the flow-sheet, (iv) the general structure of the separation system (vapour and liquid recovery system), and (v) energy integration. In this line, Sargent (2005) mentioned that "the idea of automatic generation of a dynamic mathematical model from a purely qualitative description of the process opens new perspectives and deserves a little bit more discussion". In summary, in the context of "no existing perfect model", the set of fundamental or empirical equations can be provided or selected from a previously stored library, considering the purpose of the model that is to be built. Additionally, model construction should be considered by controlling the complexity of the models and can be achieved by selecting suggested models from several alternatives. In this line, Stephanopoulos et al. (1990*a*,*b*) introduced a language, MODEL.LA, to build mathematical systems automatically for process modelling. This modelling language has been improved with a modelling logic and a modelling environment to support teaching activities in courses (Bieszczad, 2000). Another modelling language, BatchDesign-Kit, supports the design of batch processes by synthesising process flowsheets and performing a systematic analysis to identify economic objectives (Linninger, 1995). In addition, TechTool can be used to construct process models using a language based on chemical and physical phenomenon (Linninger et al., 2000).

Previously, supply chains were identified as highly complex information sources. They include the plant's operational decision hierarchy with its enterprise data. Enterprise data is comprised of commercial and economic information and process knowledge itself (National Research Council, 1999). Supply chain management requires innovations that go beyond developments in information and advances in technical computing technology. However, the vision regarding computational needs in chemical industry is restricted to information storage and sharing and computing tool technologies under transactional information. Essentially, model-building methods and tools were enabled by advances in computing technologies related to the conversion of data into knowledge, support tools for process and business decisions and training methodologies for non-experts. Decision Support Systems (DSSs) and tools for process modelling, including modelling methodologies, are crucial for experts in process model building and tool use (Edgar et al., 1999).

Another milestone was reached by the Committee on Challenges for Chemical Sciences in the 21st Century (Hopf, 2004): the development of new and powerful computational methods, applicable from atomic level to chemical process and enterprise levels as a crucial factor to enable multi-scale optimisation. This challenge broadened the scope of one of the objectives attained by the PSE approach: the systematisation of decision-making through modelling and optimisation, to a new generalised paradigm.

Attention to knowledge management in the PSE field has been increasing in recent years. A simple Scopus[3] search (TITLE-ABS-KEY (ontology) OR TITLE-ABS-KEY (knowledge AND management) AND (LIMIT-TO (DOCTYPE, “ar”)) AND (LIMIT-TO (SUBJAREA, “CENG”))) demonstrates the use of knowledge management and/or ontology concepts in the Chemical Engineering area. Figure 1.2 shows the number of articles published on these topics from 1980 to 2016 (years 2017 and 2018 are not considered since Scopus needs to update the latest publications). Clearly, the mid-1990s represent a turning point for Chemical Engineering and knowledge management, as more papers began to be published from this time. Even graduate courses on integrated process operations have introduced knowledge representation in their syllabus (Doyle et al., 1997). The number of publications more than doubled from 2011 to 2016. So far, the peak is in 2015 when 252 articles were published. A simple forecast analysis conducted for 2017 to 2020 using data retrieved from Scopus (from 1980 to 2016) and published article level remains at a similar level as in previous years.

Finally, it can be concluded that systematic decision-making and the use of State-of-the-Art methods and tools to model and optimize systems are important pillars of PSE. The essential point is not only to improve on currently available, familiar subjects, but also to introduce interdisciplinary advancements to collective work on PSE.

---

[3] www.scopus.com

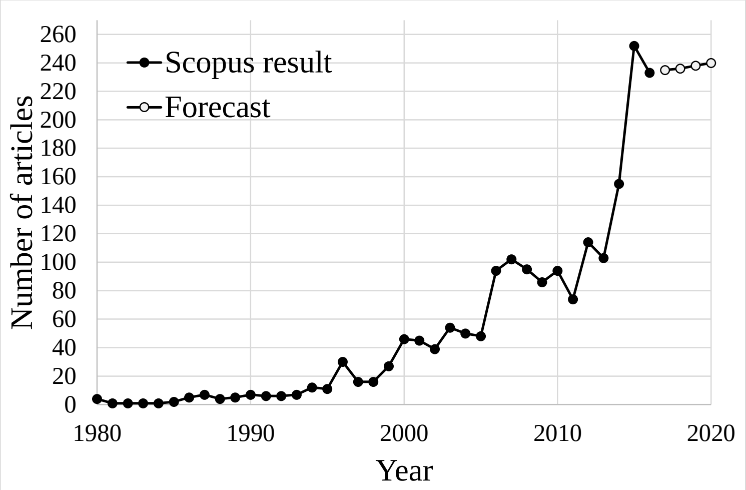


Figure 1.2.: Scopus results on knowledge management (or ontology) in Chemical Engineering between year 1980 and 2018

1.2. Thesis Outline

This thesis presents cutting-edge contributions to methodologies for conceptualising PS domain modelling and DMMs, and algorithms to build DMMs automatically, considering conceptual modelling steps from a structural and functional perspective. The thesis has been divided into five parts as shown in Figure 1.3.

Part I gives an overview of the thesis, covering PSE perspectives and the thesis outline. Thesis structure is conceptualised in Figure 1.3. Part I also contains a background chapter (see Chapter 2) presenting the context and the latest developments in the thesis topics. The background section also identifies current challenges and background connections as well as relationships between the background and the solution addressed in the thesis (see Chapter 3).

Part II introduces building and using the conceptual model of PS. Chapter 4 presents a domain ontology construction methodology, which has many outcomes: the approach itself, a consistency check for semantic models and a case study that produces a general PS Domain that is used throughout the thesis. Chapter 5 presents a new implementation of this domain, in which semantic models are used to communicate between DMM and problem instances to be solved, leading to integrated PS management.

Afterwards, Part III addresses DMM and conceptualization starting with an introduction to constraint conceptualization in Chapter 6. The ideas behind this introduction lead to the

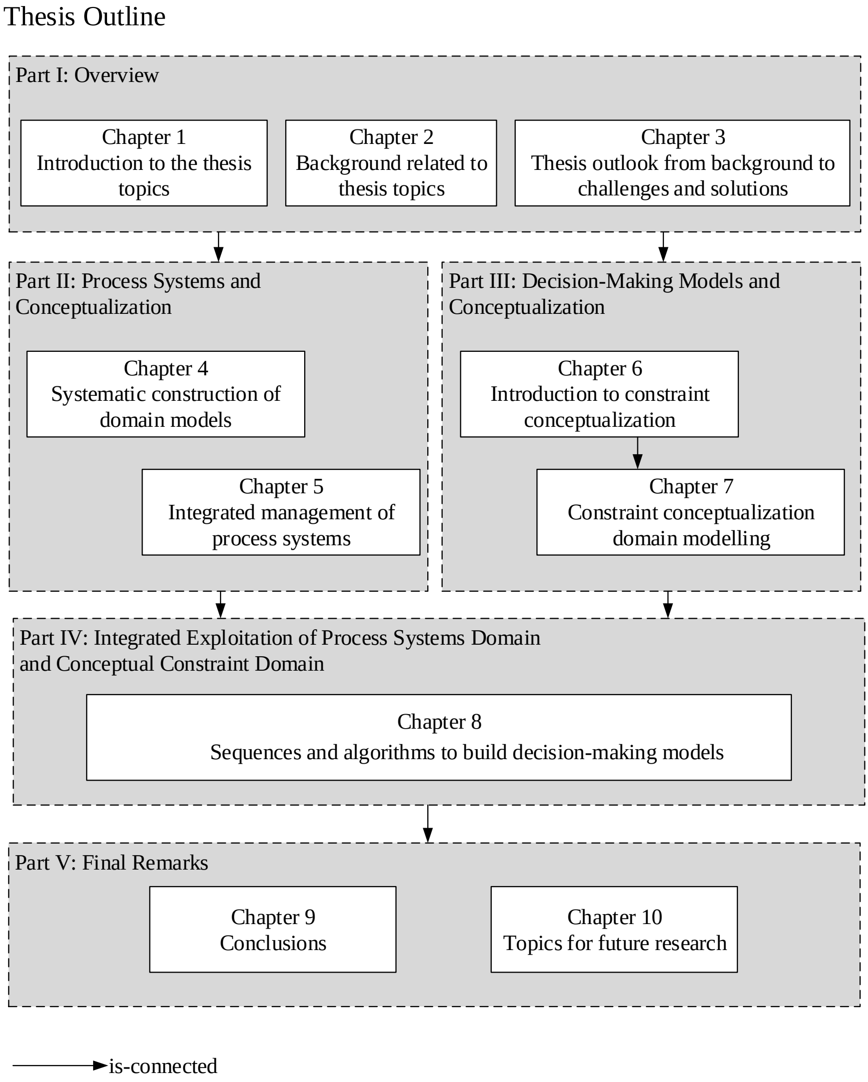


Figure 1.3.: Thesis outline

construction of a domain that has the comprehensive constraint conceptualisation scope in Chapter 7; this chapter includes concepts and relations that support the conceptual DMMs for further developments of functionalities.

Furthermore, Part IV uses both PS and constraint conceptualization to introduce functionalities with an aim of building DMMs. Thus, sequences and algorithms to build DMMs are developed in Chapter 8.

Finally, Part V restates in Chapter 9 the conclusions derived from the research developed in this thesis with the contributions; Chapter 10 introduces topics for future research.

Chapter 2.

Background

The aim of the thesis to explore conceptual modelling for integrated decision-making in Process Systems (PS). Topics are addressed in multidisciplinary fields such as Operations Research (OR), mathematical optimisation, Chemical Engineering, automation, modelling and unified approaches, which are thoroughly reviewed in this background chapter. Decision-making is in Section 2.1, with a focus on mathematical models built to support decisions and associated elements, structures and mathematical programming solution strategies used to solve the models. Section 2.2 introduces PS representation and management. Section 2.3 introduces conceptual modelling and describes the fundamentals of conceptualisation and recent developments in Process Systems Engineering (PSE) and conceptualisation. Finally, 2.4 includes a summary of state of the art in DMM integration and the use of conceptual models in PSE.

2.1. Decision-Making

Decision-making in PS uses models and systematic methods, including process, physical, and procedural control models as well as appropriate solution strategies. In general, it is accepted that any decision-making process consists of 4 phases (Simon, 1960): set-up (intelligence), design, choice, and decision implementation (see Figure 2.1). The set-up phase is associated with the need to characterise the system's situation. The need for a decision-making process and objectives are established, resulting in a problem statement. The design phase entails selecting the solution approach, identifying and designing alternatives and building the DMMs. In the choice phase experimentation is undertaken with the model(s), which results in a decision: a suitable alternative is selected, a sensitivity analysis is undertaken, and a solution is adopted.

Clearly, the design phase is the core of the decision-making process, since DMMs sustain the main structure of model-based methods. Techniques exist that are not just based on

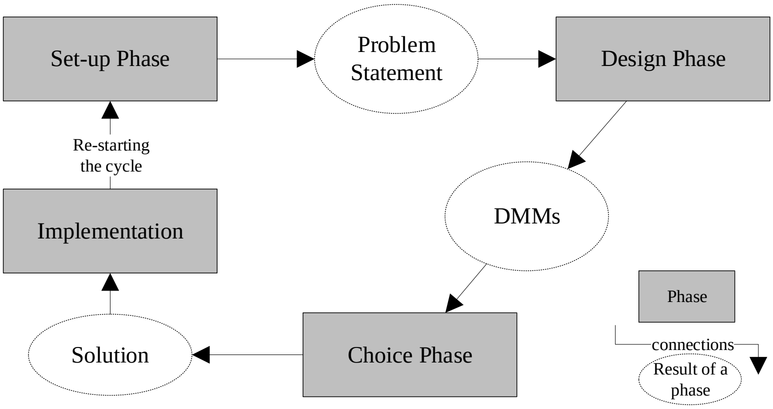


Figure 2.1.: Phases of the decision-making process (Simon, 1960)

models, but models supported by methods, algorithms and software tools that help in the design phase of decision-making process such as (Zhang et al., 2015):

- Mathematical programming (i.e. including mathematical models and their optimization over a subject),
- Case-based reasoning (i.e. a type of knowledge-based decision making which requires background knowledge of similar cases in order to solve the new case),
- Data warehouse and data mining,
- Decision tree,
- Fuzzy set and models, and
- Constraint programming.

However, apart from mathematical programming, these techniques are outside the scope of this thesis. Among the techniques used to formulate a DMM, mathematical programming is designed to find optimum decision(s). It is a crucial technique/tool for decision-making and PSE studies.

Constraint programming should also be mentioned, although it is not discussed further in this thesis. Constraint programming uses the same elements as mathematical programming, including a set of variables or constraints. However, constraint programming supports logical constraints (e.g. if, else if), arithmetic expressions (e.g. modulo, integer division, minimum, maximum) and special constraints such as “all-different” (IBM, 2018*b*). Several constraint types of constraint programming are introduced in Hooker (2012).

In the next section, mathematical programming is discussed as an advanced tool for mathematical modelling and optimisation in many study areas to build DMMs. The section covers modelling elements, structure, solution strategies, and tools to introduce mathematical programming, from a general perspective.

2.1.1. Mathematical Programming

Mathematical programming is a way to model and optimise PS and many other systems. With the help of its elements (see Section 2.1.1.1) and structures, models are built according to complexity and the details of the problem (see Section 2.1.1.2). As is the nature of mathematical programming, decision(s) are established by applying solution techniques (see Section 2.1.1.3) and special tools for modelling and optimisations are developed (see Section 2.1.1.4).

One of the exceptional applications of mathematical programming is classification. Üney and Türkay (2006) developed a mathematical programming model for classification in PSE. More importantly, Xu and Papageorgiou (2009) adopted a hyper-boxes mathematical programming model to solve multi-class data classification. The original mathematical programming model was developed to find an optimum plant layout in which hyper-boxes are concepts of units that appear in production facilities. These two implementations show that mathematical programming is a powerful tool that can serve different purposes. Notably, the second implementation (from (Xu and Papageorgiou, 2009)) shows that the concepts behind mathematical programming elements can be changed to solve problems other than those comparing to the original construction of DMM.

2.1.1.1. Mathematical Programming Elements

Generally, f is an objective that is a function of x (variables) to be maximised or minimised. An optimisation problem is named according to the nature of the objective function(s) and constraints. In addition, the nature of the variables (e.g.continuous, discrete and integer variables) is considered, to select the possible solution algorithms. Penalisation terms can be added to the objective function to find a feasible solution to the problem if the solver that is used does not depend on derivatives (e.g. genetic algorithms). Depending on the optimisation procedure, the optimum value can be called global or local optimisation (Nocedal and Wright, 1999). The formal representation of mathematical programming (variables, x, the objective function, $f(x)$, equality constraints, $h(x)$, and inequality constraints, $g(x)$) can be illustrated as follows:

$$\begin{aligned} \min_{x} \quad & f(x) && && (2.1) \\ \textbf{s.t.} \quad & h(x) && = 0, && (2.2) \\ & g(x) && \leq 0. && (2.3) \end{aligned}$$

DMMs expressed using mathematical programming consist of five items (Williams, 1978): (i) *objective:* a mathematical expression to be minimised or maximised during the decision-making procedure, (ii) *variables:* adjustable elements of models or unknowns (including decision elements), (iii) *parameters:* coefficients of the model (scalars or matrices that do not change during the calculations), (iv) *constraints:* relations between parameters and variables, that have to be considered to ensure the feasibility of the proposed decision, and (v) *sets/subsets:* collections of objects or items gathered together indicating the size/complexity of the mathematical representation to be solved.

Traditionally, DMMs are constructed manually according to existing data and problem features (e.g. time horizon, parameters, decision variables, applicable/significant constraints, and objective functions). Models are improved by revising cycles according to the problem and solution features.

One of the crucial elements in DMMs is an objective function(s). There are many discussions about the objective function(s) in mathematical programming relating to practicability and conceptual construction. Performance indicators are used to find the optimum answer to the problem and model that exists in PS (see Section 2.2.1). Regarding conceptual understanding, three sustainability elements (i.e. economic, social, and environment) are generally discussed and used. However, it is hard to implement these three pillars of sustainability in real organisations since there is no objective function(s) of organisations that is better than profitability (Williams, 1978).

Solving a mathematical programming model of a situation with an objective function provides a feasible solution if it exists (i.e. a solution that satisfies the constraints). Some constraint types can be introduced as follows (Williams, 1978):

- Production or labour capacity constraints,
- Raw material availabilities (limited supply),
- Marketing demands and limitations (additionally limitations should be added if we can produce more than the market upper limit),
- Material balance (continuity) constraints, and
- Quality obligations.

Or constraints can be sorted as their functionalities as follows:

- Hard and soft constraints (i.e. hard constraints are the constraints which cannot be violated, whereas soft constraints can be violated in a range with a cost; for example, the availability of the raw materials can be extended with capacity, but it will bring extra cost.),
- Chance constraints (i.e. if a constraint should be held in a range of probability),
- Conflicting constraints (i.e. constraints that cannot be satisfied simultaneously),
- Inactive constraints (i.e. redundant for the current model and data (or problem input) but can become active if the data changes),
- Simple and generalised lower and upper bounds, and
- Unusual constraints (i.e. including some situations or logical expressions such as “product 1 can be produced only if product 2 is produced, but neither of products 3 or 4 are produced.”).

Depending on the problem, this classification can be reduced to a collection of different types of constraints. For supply chain design models, constraints can be grouped under material balance or balances, design constraints, capacity constraints and market-suppliers constraints (Laínez et al., 2009). For a short-term scheduling problem, in addition to material balances/balances, other types of constraints are added to the collection (e.g. allocation, sequence-dependent changeover, operational, assignment and utility constraints) depending on the detail and range of the problem and the required modelling features (i.e. discrete, continuous time) (Méndez et al., 2006).

2.1.1.2. Structural Types of Mathematical Programming Models

Mathematical programming models can be classified depending on their elements (e.g. variables, constraints, number of objective functions, and time dependent variables). Generally, models built using mathematical programming have linear constraints, single or multiple objectives with a linear or non-linear structure, and (mixed) discrete and continuous variables. Thus, mathematical programming models can be classified according to non-linearity or discrete and/or continuous variables as follows:

Linear programming (LP): LP is the based structure with linear constraints and objective function(s) with continuous variables.

Mixed-integer linear programming (MILP): MILP introduces integer decision variables to the LP structure.

Non-linear programming (NLP): NLP introduces non-linear constraint(s) and/or objective function(s) to the LP structure.

Mixed-integer non-linear programming (MINLP): MINLP introduces integer decision variables to the NLP structure.

Optimisation problems can also be classified as dynamic optimisation (or dynamic programming) if time-dependency appears in the models. Furthermore, models can be stochastic (stochastic programs) or deterministic, depending on their completeness. Finally, the number of objectives introduce the single objective and multi-objective optimisation types (Mitsos, 2012). However, the essential feature of the mathematical programming structure may be convexity, which affects the preference for solution algorithms (Nocedal and Wright, 1999). Next, solution algorithms for the optimisation of mathematical programming models are explained.

2.1.1.3. Solution Algorithms

Solution algorithms are developed according to the structures (see in Section 2.1.1.2), problem types (e.g. convex and non-convex) and scales. There may be some restrictions on algorithms depending on the type of problem. For instance, a simplex method, which can solve LP problems, cannot solve a MILP problem unless another algorithm such as branch and bound supports the solution. However, solution algorithms can perform differently on a different class of problems. For instance, the "no free lunch" theorem (Whitley and Rowe, 2008) states that algorithms may perform differently on a different class of optimisation problems and the algorithm of parameters should be adjusted according to problem types, structures, and scales. The following descriptions introduce examples that are used to solve mathematical programming problems (Graells, 2016):

Exact algorithmic methods: in some cases, specific algorithms can be applied that ensure the optimum solution is achieved with limited computational effort, for instance, the simplex method for LP. The simplex method follows the edges and explores all the vertices that appear in the problem (Dantzig et al., 1953).

Enumerative methods: another type of solution algorithms can be grouped under enumerative methods that create a complete, ordered listing of all the items in a collection. For instance, branch and bound is one method that is used to solve MIP problems. The main idea of the branch and bound method is to systematically search branches that appear in the MIP and calculate upper and lower bounds of the objective function. Solution branches are discarded in the case of infeasible solutions for the objective function and the best integer solution is reached (Wolsey, 1998; Pochet and Wolsey, 2006).

Decomposition algorithms: the main idea for solving MINLP problems is two-phase strategy that divides the main problem into two parts Kocis and Grossmann (1988). The first is the sub-problem with NLP problem, which is produced by fixing integers into certain values to optimise continuous variables. The solution of this problem is the upper bound for an MINLP minimisation problem. The second part is the master problem, which optimizes the discrete variables and introduces an increasing sequence of lower bounds. These two parts of the problem are arranged in a way that enables the non-convexity of the problem to be identified. The proposed two-phase strategy (Kocis and Grossmann, 1988) has been implemented in DICOPT (Discrete Continuous Optimizer), which is also used in GAMS (General Algebraic Modelling System, GAMS (2016)) as a solver.

Generic or evolutionary algorithms: evolutionary algorithms are inspired by biological evolution and are used as an optimisation tool to solve engineering problems. The main idea is to have a population of candidate solutions for a given problems and then create new-optimal populations using natural genetic variation and selections (Mitchell, 1999). A book by Holland (1975) (Adaptation in Natural and Artificial Systems) illustrates genetic algorithms as an extraction of biological evolution and presents the technical explanations for the algorithm. Some issues relating to genetic algorithms are constraint handling in cases of inequality and infeasible solutions and the requirement of penalty terms for the objective function (Francisco et al., 2005; Shokrian and High, 2014; Lee et al., 1999; Lopez Cruz et al., 2003; Michalewicz et al., 1992; Upreti, 2004).

Most of these algorithms are implemented in commercial and non-commercial solvers[1]. However, an interface or an application needs to be used to translate and connect mathematical programming models and solvers. Mathematical programming tools (introduced in Section 2.1.1.4) produce a component between the users and the solvers to build optimisation models as explained in the next section.

2.1.1.4. Mathematical Programming Modelling Tools

There are many tools for building mathematical programming models including General Algebraic Modelling System (GAMS), Advanced Interactive Multidimensional Modelling System (AIMMS), A Mathematical Programming Language (AMPL), IBM ILOG CPLEX Optimization Studio (IBM), and Pyomo (Pyomo, 2018). Apart from these software applications, MOSAIC handles code generation for many mathematical model based software

---

[1] A list of solvers for mathematical models can be used in GAMS can be obtained from the following link with short descriptions (link).

tools by translating a model written in Latex into the required special syntax (Kuntsche et al., 2011).

To demonstrate how these modelling languages (especially algebraic modelling systems used for mathematical programming) can be used, an example of material balance constraint implementation using GAMS is introduced with some observations. The example is a simplified material balance constraint from a mathematical programming model developed by Kondili et al. (1993) and presented in Equation (2.4). The information required for this material balance is given in Table 2.1 and Table 2.2 and the state-task network (STN) of a sample case is shown in Figure 2.2.

$$S_{s,t} = S_{s,t-1} + \sum_{i \in \overline{T}_s} \overline{\rho}_{i,s} \sum_{j \in K_i} B_{i,j,t-p_{i,s}} - \sum_{i \in T_s} \rho_{i,s} \sum_{j \in K_i} B_{i,j,t}, \qquad \forall s \in S, \forall t \in T, t \neq t_0. \tag{2.4}$$

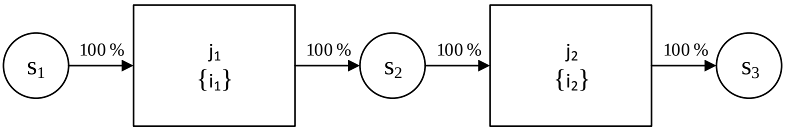


Figure 2.2.: STN of the illustrative material balance constraint example

Three steps for the development of this constraint (shown in Equation (2.4)) are presented. First, the equation without any restrictions is depicted in Equation (2.5) and then parameters ($\overline{\rho}_{i,s}$ and $\rho_{i,s}$) that have two dimensions (states and unit procedures) are added in Equation (2.6). Finally, Equation (2.7) includes the complete constraint. The restriction on the units (i.e. (K_i) expresses the capacity of performing a specific unit procedure) and the time restriction (i.e. ($t \neq t_0$) directly affects the equation instances) are added. After each equation the GAMS code and equation instances resulting in GAMS are shown.

Table 2.1.: Nomenclature details of Equation (2.4)

Symbol	**Explanation**	**Value**
Sets		
s	states	$S = \{s_1, s_2, s_3\}$
i	unit procedures	$I = \{i_1, i_2\}$
j	units	$J = \{j_1, j_2\}$
t	time period	$T = \{t_0, t_1, t_2\}$
Subsets		
T_s	set of unit procedures receiving material from state s	$T_s = \begin{array}{c c} & \begin{array}{cc} i_1 & i_2 \end{array} \\ \begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \end{array}$
$\overline{T}_s$	set of unit procedures producing material from state s	$\overline{T}_s = \begin{array}{c c} & \begin{array}{cc} i_1 & i_2 \end{array} \\ \begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} & \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$
K_i	set of units capable of performing unit procedure i	$K_i = \begin{array}{c c} & \begin{array}{cc} i_1 & i_2 \end{array} \\ \begin{array}{c} j_1 \\ j_2 \end{array} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$

Table 2.2.: Nomenclature details of Equation (2.4)

Symbol	Explanation	Value
Parameters		
$\rho_{i,s}$	the proportion of input of unit procedure i to state	$\rho_{i,s} = \begin{array}{cc} & \begin{array}{cc} i_1 & i_2 \end{array} \\ \begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} & \left(\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{array}\right) \end{array}$
$\overline{\rho}_{i,s}$	the proportion of output of unit procedure i to state	$\overline{\rho}_{i,s} = \begin{array}{cc} & \begin{array}{cc} i_1 & i_2 \end{array} \\ \begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} & \left(\begin{array}{cc} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{array}\right) \end{array}$
$p_{i,s}$	processing time for the output of unit procedure i to state	$p_{i,s} = \begin{array}{cc} & \begin{array}{cc} i_1 & i_2 \end{array} \\ \begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} & \left(\begin{array}{cc} 0 & 0 \\ 2 & 0 \\ 0 & 2 \end{array}\right) \end{array}$
Variables		
$S_{s,t}$	amount of material stored in state S, at the beginning of time period t	$S_{s,t} = \begin{array}{cc} & \begin{array}{ccc} t_0 & t_1 & t_2 \end{array} \\ \begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} & \left(\begin{array}{ccc} S_{s_1,t_0} & S_{s_1,t_1} & S_{s_1,t_3} \\ S_{s_2,t_0} & S_{s_2,t_1} & S_{s_1,t_3} \\ S_{s_3,t_0} & S_{s_3,t_1} & S_{s_1,t_3} \end{array}\right) \end{array}$
$B_{i,j,t}$	amount of material which starts undergoing unit procedure i in unit j at the beginning of time period t	$p_{i,s} = \begin{array}{cc} & \begin{array}{ccc} t_0 & t_1 & t_2 \end{array} \\ \begin{array}{c} i_1.j_1 \\ i_1.j_2 \\ i_2.j_1 \\ i_2.j_2 \\ i_3.j_1 \\ i_3.j_2 \end{array} & \left(\begin{array}{ccc} B_{i_1,j_1,t_0} & B_{i_1,j_1,t_1} & B_{i_1,j_1,t_3} \\ B_{i_1,j_2,t_0} & B_{i_1,j_2,t_1} & B_{i_1,j_2,t_3} \\ B_{i_2,j_1,t_0} & B_{i_2,j_1,t_1} & B_{i_2,j_1,t_3} \\ B_{i_2.j_2,t_0} & B_{i_2.j_2,t_1} & B_{i_2.j_2,t_3} \\ B_{i_3.j_1,t_0} & B_{i_3.j_1,t_1} & B_{i_3.j_1,t_3} \\ B_{i_3.j_2,t_0} & B_{i_3.j_2,t_1} & B_{i_3.j_2,t_3} \end{array}\right) \end{array}$

First Step:

$$S_{s,t} = S_{s,t-1} + \sum_{i}\sum_{j} B_{i,j,t-p_{i,s}} - \sum_{i}\sum_{j} B_{i,j,t}, \quad \forall s \in S, \forall t \in T. \tag{2.5}$$

GAMS code for 2.5:
$MaterialBalance0(s,t)..Ss(s,t) = e = Ss(s,t-1)+Sum((j,i),B(i,j,t-p(s,i)))-Sum((j,i),B(i,j,t));$

When the previous code is introduced to GAMS, the following equation instances are generated by GAMS.

```
Instances of Equation \ref{MassBalance0}:\\
---- MaterialBalance0 =E= \\
MaterialBalance0(s1,t0).. Ss(s1,t0) =E= 0 ; \\
MaterialBalance0(s1,t1).. - Ss(s1,t0) + Ss(s1,t1) =E= 0 ; \\
MaterialBalance0(s1,t2).. - Ss(s1,t1) + Ss(s1,t2) =E= 0 ; \\
MaterialBalance0(s2,t0).. Ss(s2,t0) + B(i1,j1,t0) + B(i1,j2,t0) =E= 0 ;\\
MaterialBalance0(s2,t1).. - Ss(s2,t0) + Ss(s2,t1) + B(i1,j1,t1) + B(i1,j2
    ,t1) =E= 0 ; \\
MaterialBalance0(s2,t2).. - Ss(s2,t1) + Ss(s2,t2) - B(i1,j1,t0) + B(i1,j1
    ,t2)- B(i1,j2,t0) + B(i1,j2,t2) =E= 0 ;\\
MaterialBalance0(s3,t0).. Ss(s3,t0) + B(i2,j1,t0) + B(i2,j2,t0) =E= 0 ;\\
MaterialBalance0(s3,t1).. - Ss(s3,t0) + Ss(s3,t1) + B(i2,j1,t1) + B(i2,j2
    ,t1) =E= 0; \\
MaterialBalance0(s3,t2).. - Ss(s3,t1) + Ss(s3,t2) - B(i2,j1,t0) + B(i2,j1
    ,t2) - B(i2,j2,t0) + B(i2,j2,t2) =E= 0 ;
```

Second Step:

$$S_{s,t} = S_{s,t-1} + \sum_{i \in \overline{T}_s} \overline{\rho}_{i,s} \sum_{j} B_{i,j,t-p_{i,s}} - \sum_{i \in T_s} \rho_{i,s} \sum_{j} B_{i,j,t}, \quad \forall s \in S, \forall t \in T. \tag{2.6}$$

GAMS code for (2.6):
$MaterialBalance1(s,t)..Ss(s,t) = e = Ss(s,t-1) + Sum((j,i),$ ***rho_out(s,i)*** $* B(i,j,t-p(s,i))) - Sum((j,i),$ ***rho_in(s,i)*** $* B(i,j,t));$

When the previous code is introduced to GAMS, the following equation instances are generated by GAMS.

```
Instances of Equation \eqref{MassBalance1}:\\
—— MaterialBalance1  =E= \\
MaterialBalance1(s1,t0)..    Ss(s1,t0) + B(i1,j1,t0) + B(i1,j2,t0) =E= 0 ; \\
MaterialBalance1(s1,t1)..  − Ss(s1,t0) + Ss(s1,t1) + B(i1,j1,t1) + B(i1,j2,t1) =E= 0     ;\\
MaterialBalance1(s1,t2)..  − Ss(s1,t1) + Ss(s1,t2) + B(i1,j1,t2) + B(i1,j2,t2) =E= 0   ; \\
MaterialBalance1(s2,t0)..  Ss(s2,t0) + B(i2,j1,t0) + B(i2,j2,t0) =E= 0 ;\\
MaterialBalance1(s2,t1)..  − Ss(s2,t0) + Ss(s2,t1) + B(i2,j1,t1) + B(i2,j2,t1) =E= 0  ; \\
MaterialBalance1(s2,t2)..  − Ss(s2,t1) + Ss(s2,t2) − B(i1,j1,t0) − B(i1,j2,t0)+ B(i2,j1,t2) + B(i2,j2,t2) =E= 0 ;\\
MaterialBalance1(s3,t0)..  Ss(s3,t0) =E= 0 ; \\
MaterialBalance1(s3,t1)..  − Ss(s3,t0) + Ss(s3,t1) =E= 0 ; \\
MaterialBalance1(s3,t2)..  − Ss(s3,t1) + Ss(s3,t2) − B(i2,j1,t0) − B(i2,j2,t0) =E= 0; }
```

Third Step:

$$S_{s,t} = S_{s,t-1} + \sum_{i \in \overline{T}_s} \overline{\rho}_{i,s} \sum_{j \in K_i} B_{i,j,t-p_{i,s}} - \sum_{i \in T_s} \rho_{i,s} \sum_{j \in K_i} B_{i,j,t}, \qquad \forall s \in S, \forall t \in T, t \neq t_0. \tag{2.7}$$

GAMS code for 2.7:
*MaterialBalance2(s,t)***\$(ord(t) ge 2)***..Ss(s,t) = e = Ss(s,t−1)+Sum((j,i)***\$(K(j,i))***, *rho_out(s,i) ∗ B(i,j,t − p(s,i))) − Sum((j,i)***\$(K(j,i))***, rho_in(s,i) ∗ B(i,j,t));*

When the previous code is introduced to GAMS, the following equation instances are generated by GAMS.

```
Instances of Equation \eqref{MassBalance2}:\\
—— MaterialBalance2  =E=\\
MaterialBalance2(s1,t1)..  − Ss(s1,t0) + Ss(s1,t1) + B(i1,j1,t1) =E= 0 ; \\
MaterialBalance2(s1,t2)..  − Ss(s1,t1) + Ss(s1,t2) + B(i1,j1,t2) =E= 0 ; \\
MaterialBalance2(s2,t1)..  − Ss(s2,t0) + Ss(s2,t1) + B(i2,j2,t1) =E= 0 ; \\
MaterialBalance2(s2,t2)..  − Ss(s2,t1) + Ss(s2,t2) − B(i1,j1,t0) + B(i2,j2,t2) =E= 0 ;\\
MaterialBalance2(s3,t1)..  − Ss(s3,t0) + Ss(s3,t1) =E= 0 ; \\
MaterialBalance2(s3,t2)..  − Ss(s3,t1) + Ss(s3,t2) − B(i2,j2,t0) =E= 0 ;
```

Finally, observations can be sorted as follows:

- As can be seen in the first equation instance of Equation (2.5), subset multiplication with subset(s) determines whether the equation instance element should be there or not. The first equation instance only has the $Ss(s1, t0)$ element. However, the first instance of the equation introduced in the second step (Equation 2.6) has other elements $(B(i1, j1, t0) + B(i1, j2, t0))$. In this case, these subsets ($\overline{\rho}_{i,s}$ and $\rho_{i,s}$ introduced in Equation 2.6) create connections between the equation instance and equation instance elements. Therefore, batch elements do not appear in the first step/first equation instance but are present in the second step/first equation instance.
    - The first observation is related with subset multiplication. Subsets affect how connections are created between equation instances by affecting equation instance elements. They do not exist if connections do not exist.
- Restriction of a set of units j with the subset of units (i.e. that can perform unit procedure i (K_i)) restricts the existence of sets in the equations. For instance, since i_1 can only be performed in j_1, expressions such as $B(i_1, j_2, t_1)$ or $B(i_2, j_1, t_1)$ are removed from each equation instance.
    - The second observation is related to excluding/removing infeasible variables or parameters. In this case, since they are infeasible in the equation instances, subsets support removing them.
- The addition of $t \neq t_0$ in Equation 2.7 directly affects equation instances by removing those with the set element t_0. This effect can be sampled by adding more control on instances such as $\forall s \in S_x$ where S_x belongs to another subset.
    - In the last observation, the exclusion of equation instances is discussed. This again can be achieved by a subset or directly removing a specific set from the equation instances.

2.2. Process Systems

PS are special kind of chemical, physical, or biological systems that are designed, built, and run in order to manufacture product(s) (inspired by Marquardt et al. (2010)). Enterprises, which can be managed by supply chains, are traditionally at the top levels of PS which contain data related to each entities and their management systems. “How to plan?”, “how to design?”, “how to operate?”, and “how to control?” PS are the most important questions of PSE (Takamatsu, 1983). Considering these PSE points of views related to PS, the ways of representing information are discussed in Section 2.2.1. Then, the management of this representation is introduced in Section 2.2.2.

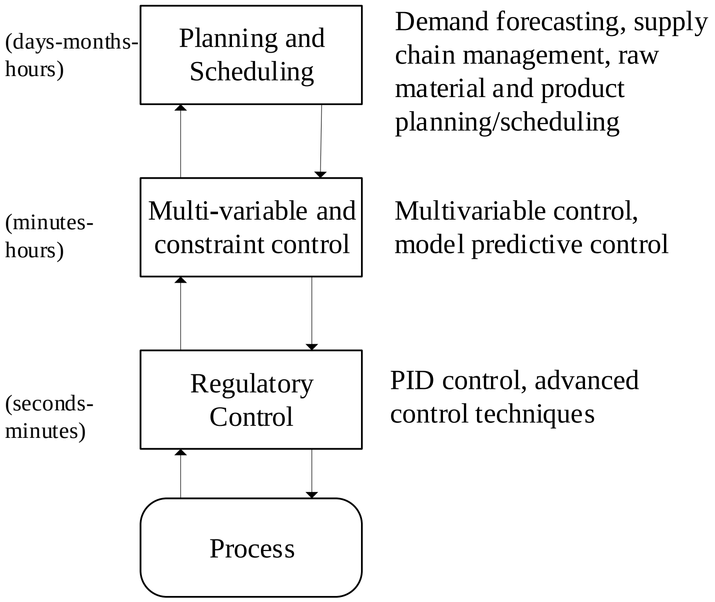


Figure 2.3.: Hierarchy of control decision in the chemical enterprise (adapted from Seborg et al. (2004))

2.2.1. Process Systems Representation

Each process system has unique data relating to processes and recipes that can be stored in data-sheets, databases, and text files. When the system has to work with other systems, normative documents (e.g. standards and technical documents) play an essential role in data and functionality integration. For instance, control decisions in a chemical enterprise are introduced in Figure 2.3. Hierarchical levels, which may affect decisions in the enterprise, are distinguished with a different time interval and possible application areas. In the same figure, the three decision-making levels in a supply chain (i.e. strategic, tactical, and operational) are combined. For instance, while planning and scheduling activities are introduced as strategic and tactical activities, multi-variable and regulatory controls are introduced as operational activities.

However, these control decisions are handled in another way when the management perspective is considered (Figure 2.4). While the idea is to create functional levels that can manage business planning and logistics (i.e. a period ranging from days to months), manufacturing operations can be considered from weeks to hours to minutes. Still, strategic, tactical and operational activities cannot be distinguished here by activity level. While strategic and tactical activities are introduced in this view as fourth-level activities, operational activities include the rest of the functional levels.

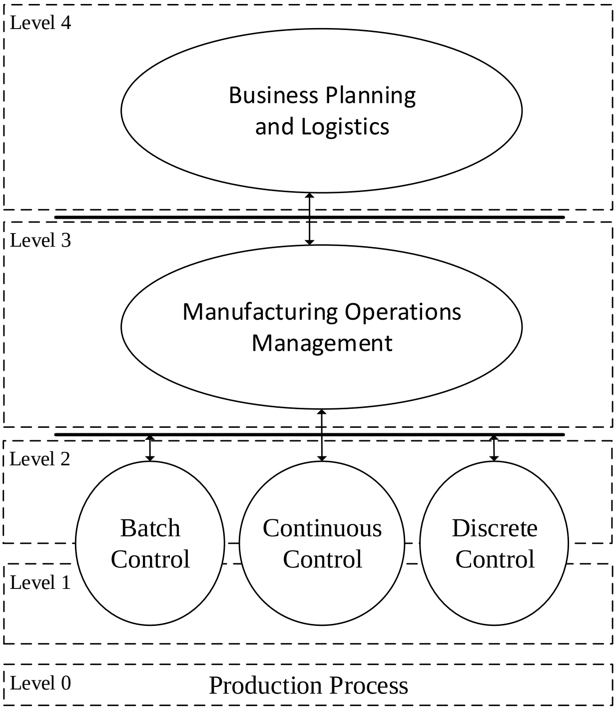


Figure 2.4.: Multi-level functional hierarchy of activities (adapted from ISA (2000))

“Models” to make decisions on strategic, tactical, and operational activities have been standardised based on a reference model for computer-integrated manufacturing: the Purdue Model (Williams, 1989). The Purdue Model is the basis for international batch and enterprise control standards (i.e. ISA88 (ISA, 2010) and ISA95 (ISA, 2000)). The models in the ISA88 Standard have also been implemented as XML Schema Definition (XSD) files (MESA, 2018) for enterprise control programs. The Purdue Enterprise Reference Model is the basis of the ISO 62264 Standard(ISO British Standards Institution, 2013), which focuses on third-level activities (work-flow, recipe, detailed production scheduling, reliability assurance) of the multi-level functional hierarchy(see Figure 2.4). ISO 62264 is used in ISO22400 (ISO British Standards Institution, 2014*a*,*b*) which has the key performance indicators for manufacturing operations management in automation systems. In the ISO 62264 Standard, 26 indicators are connected to time, real-time quality, performance and maintenance. Furthermore, NAMUR (Interessengemeinschaft Automatisierungstechnik der Prozessindustrie - User Association of Automation Technology in Process Industries) develops technical documents (recommendations), which are applicable to automation systems. In the NE-33 document, the requirements for recipe-based operations and the recipe concept are described (Namur, 2003).

The ISA88 Standard is one of the most commonly used standards for automation systems and improvement of methodologies and models in the literature (Sanchez et al., 2002; Novas and Henning, 2010; Muñoz et al., 2012, 2015; Godena et al., 2015; Vegetti et al., 2016). In PS, the ISA88 Standard (ISA, 2006*a*) is one of the most widely implemented standards for recipe management and batch control (McFarlane et al., 2003). Several software implementations are based on the ISA88 Standard for creating and operating PS, such as ABB (2018) and ControlDraw (2015). The standard provides a physical model for describing an enterprise, a procedural control model for execution sequences, and a process model for describing production processes (see Figure 2.5).

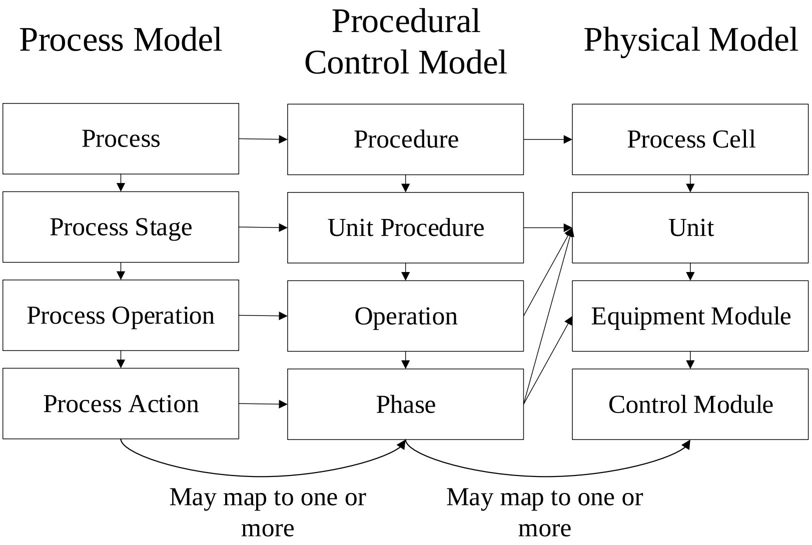


Figure 2.5.: Models in the ISA88 Standard (ISA, 2006*b*)

It is not easy to design and implement recipes in chemical plants due to the complexity of the physical, procedural control and process models, since recipes are correlations between physical model elements and procedural control model elements. In the ISA88 Standard, the recipe concept is detailed according to type and ranges from control, master, site and general recipes. They are classified according to their dependency on equipment, product types and physical model connections.

PS can also be identified in supply chain models and their elements. The main idea of supply chains is to send products (e.g. through production and storage) to customers from suppliers. The hierarchical information in each element that produces or stores materials can be presented using the three models from the ISA88 Standard and the ISA95 Standard. If this task is accomplished with direct transportation from a material supplier to customer, it is called a zero echelon supply chain. There are four echelon types in supply chain networks: suppliers, production, distribution centres, and markets, which are also called nodes. Multi-echelon supply chain models allow n-number of nodes and relations among the nodes. The number of arcs can change according to relationships among nodes (one of the distribution centres does not need to send all of the markets, or all the suppliers of the production plants do not need to send supplies to all production plants) (Goetschalckx, 2011).

2.2.2. Process Systems Management

The management of PS can be quite complicated when data content is complex in terms of properties. The advantage of a well-presented/organised representation of PS (using standards or globally accepted way) is that it reduces the time and effort of rearranging data or problem input. Therefore, data management is improved if an adequate PS representation technique is used, such as the one detailed in Section 2.2.1.

The represented data can be managed using analytical and transactional processing systems. In an enterprise, daily operations are supported by transaction processing systems and analytical processing calculations provided by DSSs. Transactional processing systems are based on structured data to automate the required activity. One of the main reasons improving the transactional processing systems to analytical processing systems is the need to process the data stored in a transactional processing system. Therefore, analytical processing systems are built to analyse data that is stored in databases through the transactional processing system (Bog, 2014). The analytical and transactional tools can also be distinguished contrary to data management (data storage or calculation) criteria. For instance, while transactional tools represent database applications in which significant amounts of data are stored in a database, analytical tools provide smarter solutions for DSSs, considering the information available through transactional processing, which stores the data in the database application (Grossmann et al., 2008). The analytical models are designed to support decision-making procedures through the integration of relevant data stored in databases or ontological models and analytical models (Laínez and Puigjaner, 2012). In addition, the integration of analytical and transactional systems can be exploited using ontologies (Muñoz, Capón-García, Laínez-Aguirre, Espuña and Puigjaner, 2017).

To manage PS efficiently, industrial plants need to be automated and continuously improved (Vogel-Heuser et al., 2015). From the Purdue reference model (Williams, 1994) to enterprise-batch control standards (ISA, 2006*b*, 2000; ISO British Standards Institution, 2013), industrial plants have undergone standardisation and implemented sophisticated automation systems. ISA Standards address batch control systems. In particular, the ISA88 Standard (ISA, 2006*b*) is used in batch facilities and affects plant design and processes (Schaefer, 1996; Nelson and Shull, 1997; Nortcliffe et al., 2001). Additionally, various systems need to be integrated for data management and to share the functionality (Harjunkoski et al., 2009). A range of systematic integration techniques can be performed, including agent-based techniques (Yu et al., 2014). Hence conceptual models in standards need to be consistent.

PS management for production contains recipes connected to physical and procedural control models. A recipe is “the necessary set of information that uniquely defines the production requirements for a specific product or operational task” (ISA, 2010). Recipe management is a complex task since it can have many phases: (i) new product and product recipe development and (ii) integration of a recipe into the available system. Another critical challenge related to recipes is knowledge management within the recipe environment to improve scheduling activities, since scheduling directly uses recipes. Muñoz, Capón-García and Puigjaner (2017) stated this problem as “(i) a set of products to be produced according to the processing order activity, (ii) a set of master recipes defining the production requirement and production path for the products in (i), and (iii) information about the resource availability and plant status provided by the process management and production information management activities” and standardised the information flow using knowledge management supported by the ISA88 Standard.

When the scale of interest changes from units to the enterprise, the problem view changes from scheduling and control activities to supply chain management (SCM). SCM is defined as “the planning and management of all activities involved in sourcing and procurement, conversion, and all Logistics Management activities” and it includes coordination and collaboration with partners, who may be suppliers, intermediaries, third-party service providers, and customers (Council of Supply Chain Management Professionals, 2013). In supply chain models, management is the task of fulfilling customer requirements to improve the competitiveness and profitability of the entire supply chain. This is achieved by integrating organisational units along a supply chain and coordinating flows relating material, information and economics.

Performance indicators play an essential role in PS activities, since they are measurements of organisations or an activity within an organisation. They can be categorised as economic worth, environmental impact, safety impact (Halim et al. (2011)) and social impact. Performance indicators play a supporting role in how decisions are made, or indicators may appear directly in the objective function(s).

One approach to determining performance indicators is to implementat sustainability ideas. Although the social impact of decision-making procedures in PSE studies is not straightforward, connections between the social sustainability dimension and economic-environmental dimensions are investigated by asking the question of which indicators directly affect social sustainability (Mota et al., 2015). Significant issues in performance indicators are how to determine direct and indirect indicators among different dimensions of sustainability. For instance:

- Figueira et al. (2015) used performance indicators in a DSS for production planning and scheduling of an integrated pulp and paper mill. The authors have determined the average and final backlog of orders, equipment utilisation ratios and total production output as performance indicators to determine the schedule.
- Seay (2015) developed a taxonomy related to sustainable engineering and sustainable process design. Social sustainability was presented as (i) safety performance and risk assessment, (ii) product stewardship (life-cycle analysis (LCA) - from the cradle to the grave), (iii) societal impact assessment and (iv) sustainability ethics (considering the recycling process).
- Goedkoop et al. (2009) created the ReCiPe characterisation that introduces 99 indicators and their mid and end point calculations for life-cycle analysis.
- Simon et al. (2008) reported indicators for chemical batch processes by separating the process model into three levels: plant, process and unit operations. A set of indicators have been presented to increase the productivity of the area in a production plant where the plant level corresponds to an area, the process level corresponds to process cell and the unit operations level managed in a unit according to the ISA88 Standard.
- A United Nations report (UnitedNations (2007)) describes a core set of 50 indicators with the subdivision themes of poverty, governance, health, education, demographics, natural hazards, atmosphere, land, oceans (seas and coasts), freshwater, biodiversity, economic development, global economic partnership, consumption and production patterns.
- Rubik and Scholl (2002) explained integrated product development which includes environmental aspects of sustainable development but excludes the social and economic dimensions. However, they linked the environmental dimension to the economic dimension with an environmental market transformation to increase sale shares.

When the perspective is expanded, performance indicators can be applied to improve the overall performance of supply chains. However, each supply chain is unique and might require special treatment (Sürie and Reuter, 2015). For instance:

Delivery performance: is the comparison between the real delivery date and the delivery date that is mutually by customers and retailers.

Supply chain responsiveness: describes the ability of the entire supply chain to react according to changes in the marketplace. Another indicator in this area is planning cycle time, which is simply defined as the time between the beginning of two subsequent planning cycles.

Assets and inventories: are economic indicators. One common indicator in this area is asset turns, which is defined by dividing revenue by total assets (i.e. revenue is all the money the customers pay for the offered products and services. Assets include all equipment and material that is involved in turning inventory into sales). Another indicator is inventory turns, defined as the ratio of total material consumption per time-period over the average inventory level of the same time-period. A common approach to increase inventory turns is to reduce inventories.

Costs: of goods should always be monitored with an emphasis on substantial processes in the supply chain. Value-added employee productivity is an indicator that is calculated by dividing the difference between revenue and material cost by total employment. Therefore, it analyses the value each employee adds to all sold products. In addition, the warranty costs indicator, which is one of the product quality indicators, should be observed.

2.3. Conceptual Modelling

Conceptualisation has supported advances in models, methodologies, and algorithms for decades by aggregating abstract mechanisms, classifying classes and generalising common elements in classes (Franz et al., 2007). Typical implementations of conceptualisation are knowledge based systems, which include knowledge representation and reasoning capabilities. Research in the field of knowledge representation and reasoning is usually focused on methods for providing high-level descriptions of the world that can be used effectively to build intelligent applications. In this context, “intelligent” refers to the ability of a system to find implicit consequences of its explicitly represented knowledge (Franz et al., 2007).

2.3.1. Ontologies as Conceptual Models

Artificial Intelligence (AI) research focuses on how to get machines or computers to carry out activities such as seeing, learning, using tools, understanding human speech, reasoning, and formulating plans. The aim is for machines to behave or think intelligently, regardless of whether internal computational processes are the same as in people or animals (Nof, 2009).

An ontology is one of the knowledge models that AI research has used to build methodologies, since ontological models can provide shared conceptualisation, a formal language for computers or domain representation. Ontologies are defined with their modelling elements (see Section 2.3.1.1), formal language structures (see Section 2.3.1.2), usage (see Section 2.3.1.3) and management tools (see Section 2.3.1.4) as in the following sections.

2.3.1.1. Ontologies and Their Elements

The definition of ontology depends on the subject area. For example, it can be defined as the part of philosophy that studies "what it means to exist" or it can be considered from a scientific perspective. An ontology can be defined as the "theory of existence" (Morbach et al., 2009) or "an explicit specification of a conceptualisation" in the modern computer world (Gruber, 1993).

Furthermore, an ontology is an engineering "artefact" (or fabrication): (i) it is constructed by a specific vocabulary used to describe an absolute reality and (ii) it relies on a set of explicit assumptions (including how concepts should be classified). Thus, an ontology describes the formal specification of a domain: (i) a shared understanding (scope of interest) and (ii) a formal and manipulative machine model of a domain (Bechhofer, 2007). These features support the required environment for experts to share domains and to improve associated applications.

Formally, an ontology can be defined as follows:

$$O = \{I, P, C, D, A\} \tag{2.8}$$

Ontology O has a set of concepts, C, and a set of object properties, P, in Expression (2.8); other members are explained in Table 2.3.

The main relation of concepts is the "is-a" (or subClassOf) relation which builds the skeleton of an ontology which is also called taxonomy. This relation is related to the existence and can be expressed as follows:

OWL syntax	Definitions
O	Ontology model
I	Set of instances (or individuals): basic ground level objects that have own meaning for each other.
P	Set of properties which can be defined as object properties; these relations define in which concepts and instances can be related to one another.
C	Set of concepts: concepts have sets and are classified with kinds of things and types of objects.
D	Set of properties which can be defined as data property.
A	Set of axioms defined over the union of all the members of the ontology.

Table 2.3.: Definitions of ontology elements

$$C_2 \sqsubseteq C_1 \tag{2.9}$$

$$C_3 \sqsubseteq C_1 \tag{2.10}$$

Expressions (2.9) and (2.10) introduce the main relation between concepts, the known as origin-destination concept pairs. In these expressions, "C_2 is-a C_1" and "C_3 is-a C_1". For instance, the "GeneralRecipe" and "MasterRecipe" concepts are from the ISA88 Standard. They are defined as "Recipe" and can be expressed in an ontology as follows:

$$GeneralRecipe \sqsubseteq Recipe \tag{2.11}$$

$$MasterRecipe \sqsubseteq Recipe \tag{2.12}$$

Ontologies that represent the model itself (through concepts, relations, and axioms) can be named as TBox (terminological box) and ABox (assertional box). that represent the model itself. TBox introduces abstract information with concepts and relations. ABox represents the reality with the instances. This distinction between TBox and ABox helps to represent systems beyond storing concept definitions and assertions. Moreover, it takes ontologies to another level with implicit knowledge that can be made explicit through inferences (details are given in Section 2.3.1.3) (Franz et al., 2007).

2.3.1.2. Ontology Representation

The Web Ontology Language (OWL) is a formal semantic language that is designed for applications that need to process the content of information instead of just presenting information to humans through graphs and relational trees (W3C, 2018). OWL facilitates greater machine interoperability of web content than supported by Extensible Markup Language (XML), Resource Description Framework (RFD), and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics (AI Foundations of Computational Agents, 2018). For instance, the examples introduced in Expressions (2.9) and (2.10) can be expanded by a graphic view of concepts as depicted in Figure 2.6. The language syntax for this example is presented in Figure 2.7.

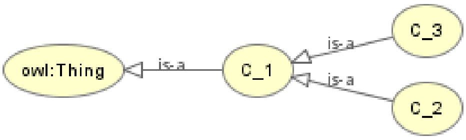
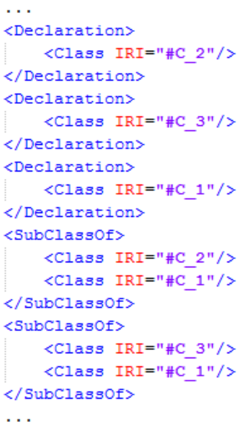


Figure 2.6.: Graphical demonstration of Expressions (2.9) and (2.10)



```
...
<Declaration>
    <Class IRI="#C_2"/>
</Declaration>
<Declaration>
    <Class IRI="#C_3"/>
</Declaration>
<Declaration>
    <Class IRI="#C_1"/>
</Declaration>
<SubClassOf>
    <Class IRI="#C_2"/>
    <Class IRI="#C_1"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#C_3"/>
    <Class IRI="#C_1"/>
</SubClassOf>
...
```

Figure 2.7.: OWL language demonstration of Expressions (2.9) and (2.10)

XML provides a syntax designed to be machine readable, but also it is possible for humans to read. It is a text-based language, where items are tagged in a hierarchical manner. The syntax for XML can be quite complicated, but at the simplest level, the scope of a tag is either in the form of $\langle tag.../\rangle$, or in the form of $\langle tag.../\rangle$ ... $\langle /tag/\rangle$.

RDF is a language built on XML, providing individual-property-value triples.

RDF-S can be used to define resources (and so also properties) in terms of other resources (e.g., using subClassOf). RDF-S also allows the user to restrict the domain and range of properties and can provide containers (i.e. sets, sequences, and alternatives).

This common formal semantic language is essential to use the models in a software environment and produce common usage libraries and information. OWL API is one of the most widely used JAVA libraries built by researchers and computer scientists for OWL files (explained in Section 2.3.1.4).

2.3.1.3. Ontology Usage

Ontologies can be used as dictionaries (Cea and Montiel-Ponsoda, 2007), database applications, knowledge management systems, and semantic web. However, this section focuses on how ontologies are used under Description Logic (DL). The main idea of DL is to use the ontology representation for a range of functionalities such as reasoning, consistency checking and inference. DL is a knowledge representation of formalisms that allows the representation and management of conceptual and terminological knowledge in a structured and semantic way with operational elements (summarised in Table 2.4) (Lutz, 2002).

Concept constructors	**Explanation**
$\neg$	negation
$\sqcap$	conjunction (membership assertion)
$\sqcup$	disjunction
$\forall$	universal value restriction of roles
$\exists$	existential value restriction of roles
$\sqsubseteq$	inclusion axiom

Table 2.4.: Concept constructors and explanations

DL is free from variables unlike first-order logic. The following expression gives an example of first-order logic ($\wedge$: intersection) with variables (“x”).

$$C(x) \wedge D(x) \tag{2.13}$$

DL implementations do not have any variables so allow the ontologies to be represented without instances (without ABox). In the following example, intersections of concepts are used to restrict the set of individuals under consideration to those that belong to both C and D concepts.

$$C \sqcap D \tag{2.14}$$

The benefit of this variable/instance-free approach is that it provides a modelling environment with a range of relations. DL divides an ontology into a TBox and an ABox for instance (or individual) management, so that TBox can be used to present terminological knowledge about the application domain, as in the following expressions:

$$Woman \equiv Person \sqcap Female. \tag{2.15}$$

$$ControlRecipe \equiv Recipe \sqcap ControlRecipeProcedure. \tag{2.16}$$

Such declarations (Expression (2.15) and (2.16)) are usually interpreted as logical equivalences, which provide both sufficient and necessary conditions for classifying an individual as a "Woman" or a "ControlRecipe".

ABox contains extensional knowledge about the domain of interest, that is, assertion about instances, usually called membership assertions ($\sqcap$) such as

$$Female \sqcap Person(PAULA) \tag{2.17}$$

$$ControlRecipeProcedure \sqcap Recipe(AFControl) \tag{2.18}$$

states that the "PAULA" instance is a female person and the "AFControl" instance is a recipe and a control recipe procedure. Given the above definition of woman, one can derive from this assertion that the "PAULA" instance is an instance of the concept "Woman" and the "AFControl" instance is an instance of the concept "ControlRecipe".

Similarly,

$$hasChild(PAULA, RAMON) \tag{2.19}$$

$$hasPhase(AFControl, addMonomerAN) \tag{2.20}$$

specifies that the "PAULA" instance has the "RAMON" instance as a child with the role assertions; the "AFControl" instance has the "addMonomerAN" instance as a phase. The basic reasoning task in an ABox is the instance checking, which verifies if a given individual is an instance of (belongs to) a specific concept.

Reasoners play a crucial role in developing the inference of ontologies. Automated reasoners (e.g. Pellet, FaCT++, HerMiT, ELK) take a collection of axioms that are written in OWL and offer a set of operations on the ontology's axioms. Reasoners can roughly be separated into consequence-driven reasoning and tableau-based approaches. Consequence-driven reasoners cannot use the "or" expression or (not (A and B)). They infer "M SubClassOf (p some B)" from "A SubClassOf: B" and "M SubClassOf (p some A)". Tableau-based reasoners seek to construct such a model using completion rules; the completion rules, which can be applied to (an abstraction of) such a model to expand it so that it satisfies all axioms.

In general, reasoners test basic tasks as follows (Franz et al., 2007):

Knowledge-based consistency test is related to the coherence of concept and instance levels and tests the consistency of each instance. For example, it tests whether the instance of any concept belongs to two disjoint concepts (an instance of the "Dog" concept cannot be an instance of a "Person" concept at the same time) (Franz et al., 2007).

Concept satisfiability tests are related to the interpretation of a model of a specific concept. "A concept is satisfiable if there exists a model of it. Otherwise it is unsatisfiable" (Meissner, 2011).

Concept subsumption tests infer new relations within the ontology, using existing relations (Franz et al., 2007).

2.3.1.4. Ontology Management Tools

Protege is a tool for building classes, relations and axioms within OWL files (Protege, 2015). It is used for ontology editing and knowledge acquisition. The interface allows the creation of ontological models and the import of relations and concepts through XML files. However, it is connected to state-of-the-art reasoners available as free sources that can be used to make inferences. Other than Protege, OWL-API has released the OWL library for JAVA as an open-source tool (Palmisano and Olwapi, 2011; OWL API, 2018). This allows users to build custom-made software applications based on OWL management tools. Another tool, Jython language, (Jython, 2018) has been used successfully for ontology management by Capón-García et al. (2017).

2.4. Unified Approaches

2.4.1. Integration of Decision-Making Models

The Committee on Challenges for the Chemical Sciences in the 21st Century (Hopf, 2004) indicates that the development of new and powerful computational methods, applicable from the atomic level to the chemical process and enterprise levels, is a key factor to enable multi-scale optimisation. This would broaden the scope of one of the main objectives attained by the PSE community, the systematisation of decision-making through modelling and optimisation, to a new generalised paradigm. However, integration has a broad meaning considering the objective, object, players and ways of integration. For instance, Scholten (2007*a*) has introduced five different sub-levels of integration: (i) common culture, (ii) common standards, (iii) information sharing, (iv) coordination (within the same organisation) and (v) collaboration (the highest level of integration, between competing organisations). However, integration can also be defined as a process. Kelly et al. (2013) introduced another five interconnected usages of integration as follows:

Integrated treatment of issues relates the issue under consideration to other social, environmental and economic issues to improve decision quality by reducing negative effects on others.

Integration with stakeholders links policies among stakeholders and the effects resulting from these decisions.

Integration of disciplines connects varying perspectives with understanding of the system related to the same issue.

Integration of processes combines many types of processes in the same perception to make decisions (e.g. the operation of energy storage according to market demands is one of the practical implementations).

Integration of scales of consideration is related to the aggregation of an interest. Scale consideration does not mean keeping these physical boundaries; it means coming to an understanding between actors and physical models and/or the linkage between model components and the complexity of the problem within the defined area.

There is no doubt that a “model” is required for further discussion of integration issues within the outlook of this thesis. The main application of the thesis is PS, which are built, designed, and run to manufacture products under the enterprise structure aspect (from single to multi-echelon supply chains). The required model is multi-level organisational hierarchies that are organised to make various types of decisions (from control, manufacturing to business planning). Section 2.2.1 introduced the PS representation of these multi-level hierarchies with control decisions (see Figure 2.3), activities (see Figure 2.4) and control models (see Figure 2.5). Here, Figure 2.8 introduces multi-level organisational hierarchies with units for the decision to be made. Coordination of decision units and process control are achieved by considering the hierarchy of layers.

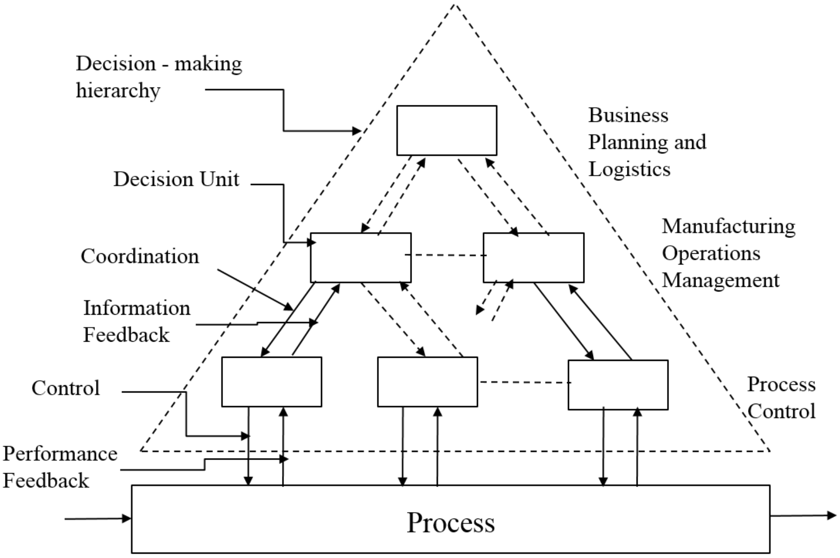


Figure 2.8.: Multi-level organizational hierarchies (based on (Mesarovic et al., 1970; ISA, 2000; Seborg et al., 2004; Baldea and Harjunkoski, 2014))

Integrated multi-echelon supply chain management within multi-level organisational hierarchies is the key factor for improving overall performance. Hierarchical systems are used to (re)construct or (re)design an (existing) system to improve overall performance. Decisions made among hierarchical levels affect each level's performance. During construction of complete integrated control of a chemical enterprise, complete control is rarely allowed due to the integration of complex models, expensive computations, and technical constraints (Mesarovic et al., 1970). To support decisions in the hierarchical system, a flexible and reliable integration system is required. DMM integration in a hierarchical system is a complex issue that requires advanced solution methods (i.e. (i) hierarchical, (ii) iterative and (iii) full-space methods (Maravelias and Sung, 2009)). These methods may provide computationally effective solutions, communication between DMMs and the opportunity to implement complementary strategies. In practice, DMMs that appear to solve different decisions in multi-level hierarchies are (i) developed through various mathematical models, (ii) solved by different sets of users and (iii) solved using different algorithms. In this line, Shobrys and White (2000) have introduced two major challenges for DMM integration: changing human behaviour and changing organizational behaviours. For the human behaviour, it is hard to gain acceptance and use of sophisticated, complex tools. Organizational behaviour is more complex, since decision-making integration requires changes in organisational behaviour so that different parts of the organisation migrate to better tools.

Considering the previous discussions on DMM integration, bibliographical research on state-of-the-art is introduced under topics such as treatment of issues, scale consideration and supply chain integration.

2.4.1.1. Treatment of Issues

Integrated treatment refers to the need to analyse an issue by considering all the required perspectives, which are usually related to the main pillars of sustainability (i.e. social, environmental, and economic objectives). The integration procedures are generally related to the solution of designing supply chains and production processes with models containing more than one conflicting objective, and aiming to improve overall supply chain performance (Chen et al., 2003; Mele et al., 2011). Multi-objectives are used to solve the uncertainty that may arise from unknown parameters, market demands and price changes (Guillén-Gosálbez et al., 2005). To solve these problems, Pareto curves have been used to support the decision (Medina-González et al., 2017).

2.4.1.2. Scale of Consideration

Integration of the scale consideration simply expresses aggregated information in the area of interest (the domain). Scale consideration does not mean keeping physical boundaries; it refers to an understanding between actors and physical models and/or the linkage between different model components and the complexity of the problem within the defined area. One of the solution strategies for these issues is to produce comprehensive DMMs using multi-parametric programming for decision-making integration (Pistikopoulos and Diangelakis, 2016).

Integration of decision levels (see Figure 2.8: business/planning, operations/scheduling, and control) has been the focus of many studies over the years (Shobrys and White, 2000). One way to integrate is to concentrate on two consecutive levels (i.e. integration of planning and scheduling decisions) rather than considering all levels. Recently, papers have been published on different angles of integration planning and scheduling decisions within PS. For instance;

- Data integration between levels is a crucial starting point for procedures related to the integration of decisions. A critical aspect is the introduction of the ISA95 Standard approach for production systems and problem solving using structured

XML files (Harjunkoski, 2014). Additionally, Vegetti and Henning (2015) have introduced an ontological framework to integrate planning and scheduling input-output activities in PS.

- Data integration is also used to address the linkage between scheduling DMMs and planning information, aimed at supporting decision-making procedures in batch plants in the chemical-pharmaceutical industry (Moniz et al., 2014). Moreover, standards have played an essential role in the integration of system elements (e.g. recipes).
- An agent-based solution approach has been used to manage and monitor the data exchanged between planning and scheduling (Luo et al., 2015).
- Muñoz et al. (2015) dealt with the complexity of planning and scheduling integration using a Lagrangian decomposition approach. Scheduling and planning sub-problems are created for each facility/supply chain entity and their dual solution information is shared through an ontological framework.

Concerning the integration between scheduling and control, the challenge of different model features adds a new dimension to the problem. Production scheduling and advanced control have been designed to work together to obtain more efficient operations in production facilities (Engell and Harjunkoski, 2012). Some relevant studies in the field are summarised below:

- One way to introduce DMM integration in this topic is to transform one of the models (i.e. accepting that they have different features such as mathematical programming and dynamic programming) into a new model so they can be solved simultaneously. In this case, Harjunkoski et al. (2009) followed following steps: (i) conversion of a mixed-integer dynamic optimisation problem into a MINLP problem, (ii) decomposition of the overall problem into scheduling and control sub-problems (e.g. using benders decomposition, Lagrangian decomposition) and (iii) use of a heuristic based approach such as agent-based simulations.
- This can also be achieved by performing a “vertical” decomposition in which the problem is decomposed into an upper-level dynamic trajectory to be optimised or re-optimised and a lower level (non-linear) model predictive control problem. The goal is to control the upper-level trajectory with the results obtained from the model predictive control problem (Kadam et al., 2002).
- Muñoz et al. (2011) proposed an integrated information environment for scheduling and process control problems. Information flow is determined by an ontology. This environment can be used to create from master recipes to control recipes within the integration strategy.

- Another suitable method for the integration of DMMs is closed-loop input-output monitoring of the overall process features (control and scheduling). The idea is to introduce the scheduling requirement and production demand into the control problem (Chu and You, 2012; Nie et al., 2012; Zhuge and Ierapetritou, 2012; Baldea and Harjunkoski, 2014; Park et al., 2014; Baldea et al., 2015).

Integration of the scale consideration also encompasses the integrated design and planning of DMMs (Laínez et al., 2009) and multi-time-scale integration considerations (Biondi et al., 2017).

2.4.1.3. Supply Chain Integration

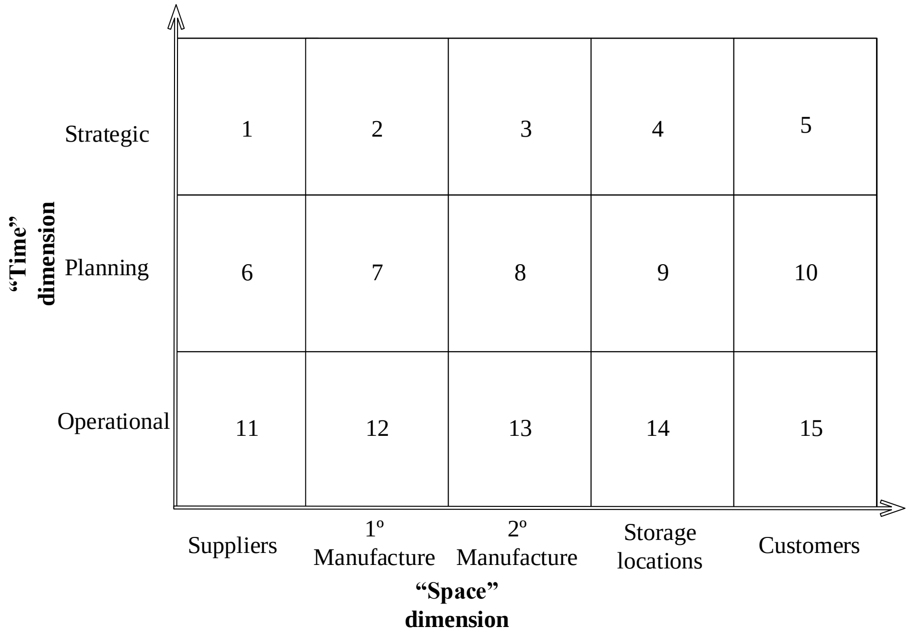
Figure 2.9 illustrates the general supply chain problem as defined by Shah (2005). The vertical dimension has operational, tactical and strategic sections as a time dimension. The horizontal dimension depicts states in the supply chain from raw material (from the suppliers) to product shipment (to the customers). Supply chain activities take place at manufacturing sites and storage locations (i.e. between two ends: suppliers and customers). This straightforward structure may change from one supply chain to another. Based on this graph, problems that occur in a supply chain may be defined within one or more regions. For instance,

- Redesign of the logistics network (regions 4 and 5, i.e. a strategic activity looking primarily at warehouses and customers).
- Campaign planning at a primary manufacturing site (region 7).
- Real-time supply chain management and control (regions 11-15).
- Negotiation of long-term supply contracts (region 1).
- Long-term manufacturing capacity planning and value chain management (regions 1-3).

One way to tackle the integrated management of supply chains is closed-loop management that focuses on environmental awareness (i.e. waste collection, product re-manufacture, product disassembly, parts refurbishing and waste disposal blocks) (Georgiadis and Papageorgiou, 2008).

To support closed-loop management, another concept, “industrial symbiosis” (or “circular economy”), has been introduced to promote environmental awareness and mutually benefit all parties in the supply chain.

An important issue for modern society is industrial symbiosis or sustainable development, considering the strength of reducing and reusing waste production. The United Nations



	Suppliers	1º Manufacture	2º Manufacture	Storage locations	Customers
Strategic	1	2	3	4	5
Planning	6	7	8	9	10
Operational	11	12	13	14	15

Figure 2.9.: Supply chain problem space (based on Shah (2005))

Sustainable Development Knowledge Platform states, in one of its 2030 goals that it is vital to “ensure sustainable consumption and production patterns”. The United Nations suggests that waste generation should be reduced using conventional methods (i.e. prevention, reduction, recycling, and reuse) and through support for countries to strengthen their scientific and technological capacity and move towards more sustainable patterns of consumption and production (United Nations, 2015). Efforts are being made by worldwide organisations, as well as local non-profit organisations such as GreenCape in Cape Town, South Africa. GreenCape provides information for local small businesses seeking an opportunity for entrepreneurship. In the list, there are options including textile processing, pallet recycling (i.e. from recycling to refurbishing and sale to the market) and contaminated bentonite processing (processing contaminated bentonite and reselling it to the market) (Carroll et al., 2014; GreenCape, 2017).

Research has been carried out on the development of DMMs for eco-industrial parks that aim to integrate resources (carbon-hydrogen-oxygen symbiosis network) in plants located in the park (El-Halwagi, 2017). One of the main issues in industrial symbiosis is which information technologies to use for data collection and storage. Efforts have been made to develop web services to share experiences and practices, detect potential opportunities and retrieve the required information (Cecelja et al., 2015; Álvarez and Ruiz-Puente, 2016).

Industrial symbiosis is an emerging area that can be directly implemented in the industry through different projects[2]. For instance, a company/process by-product, wastewater, can be used as input in another plant[3]. To generalise this effort, advanced DMMs where are required in which there is no strict information on how to model them.

Another topic underlying supply chain management is integration with stakeholders (i.e. the policies among different stakeholders and the effects resulting from these decisions). The application of this type of integration approach can be categorised under the topic of multi-enterprise decision-making strategies that are introduced as a new method of DMM integration (You and Grossmann, 2009; Hjaila et al., 2017).

2.4.2. Use of Conceptual Models in Process Systems Engineering

Earlier in Computer-Aided Chemical Engineering, Douglas (1988) presented a hierarchical conceptual modelling methodology considering five hierarchical design decisions to build a process model. The hierarchical decisions are systematised as (i) batch versus continuous, (ii) input-output structure of the flowsheet, (iii) recycle structure of the flowsheet, (iv) general structure of the separation system (vapour and liquid recovery system) and (v) energy integration. In this line, Sargent (2005) mentioned that “the idea of an automatic generation of a dynamic mathematical model from a purely qualitative description of the process pens new perspectives and deserves a little bit more discussion”. In summary, in the context of a “non-existing perfect model”, a set of fundamental or empirical equations can be provided or selected from a previously stored library considering the purpose of the model that is to be built. Model construction can be achieved by controlling the complexity of models and/or selecting suggested models from several alternatives.

Conceptual models for PS integrated management currently under discussion in the PSE community. Although conceptual models lead to potential solutions to these problems, expected practical implementations and the required scope of the achievements are not straightforward. One way to tackle these conceptual models is to build an ontology and introduce algorithms, connections and common understanding of the problems to produce solutions. Some advances in the topic can be summarised using the PSE subjects introduced in Section 1.1 as follows (accepting that each example is also a tool and may contribute to more than one subject):

- Process and Product Design

  Pharmaceutical product engineering: Remolona et al. (2017) introduced an ontology learning algorithm for pharmaceutical products.

---

[2] `<http://www.e4water.eu/industrial_sites_case_study_cbd.html>`

[3] Kalundborg Symbiosis `<http://www.symbiosis.dk/en>`

Process synthesis: Kokossis et al. (2016) introduced an ontological framework to analyse new process design solutions.

Capturing and reusing process and product designs: Brandt et al. (2008) introduced a framework supported by ontologies to capture and reuse previous process and product design information.

Pharmaceutical product development and manufacturing: Venkatasubramanian et al. (2006) built an infrastructure to support decision activities during product design. An ontology was proposed for managing pharmaceutical products and their ingredient, process, and recipe information.

- Enterprise-wide Optimisation

Risk management: Behdani and Srinivasan (2017) introduced a framework to capture supply chain disruptions. The framework is supported by a supply chain ontology and constructed simulation models to support decision-making activities.

Competitive supply chain design: Roth et al. (2017) introduced an ontology that captures stakeholder information structure.

Enterprise-wide scheduling: (Capón-García et al. (2015)) created a knowledge-based model for features that appear in enterprise-wide issues and scheduling DMMs.

Integration of planning and scheduling activities: Vegetti and Henning (2015) proposed a network containing ontological models (e.g. the ISA88 ontology, RTN ontology, STN ontology, mathematical model ontology, scheduling reference ontology) to tackle interoperability issues that may arise from conceptual differences between applications.

Semantics for industrial symbiosis networks: Cecelja et al. (2015) introduced an industrial symbiosis ontology for industrial symbiosis network synthesis. It considers geographical locations and a set of environmental criteria.

Bio-refinery feedstock and technology matching: Trokanas et al. (2015) introduced an ontology that contains raw material and processing technology types for bio-refineries to match and check availabilities of feedstocks and technologies.

- Production Scheduling

Knowledge management for scheduling activities: Moniz et al. (2014) introduced a knowledge based conceptualisation approach without ontologies to integrate scheduling and decision-making. It relies on links between recipes and mixed-integer linear programming using standards Williams (1994); ISA (2006*b*, 2000).

Knowledge management in batch process and scheduling: Muñoz et al. (2010) suggested a framework to solve some batch process problems processes. An ontology called Batch Process Ontology (BaPrOn), which has some concepts from the ISA88 Standard (ISA, 2006*a*), is used to address robust systems in control system's hierarchical levels. BaPrOn models the information in the system and integrates different software tools.

Reactive scheduling: Novas and Henning (2010) introduced an approach for the reactive scheduling problem considering a support environment and a methodology with explicit representation of domain knowledge in object-oriented representation. The main aim of the framework is to reschedule the problem while keeping the previous decisions fixed or at least with a limited number of changes.

- Process Control

Sensor measurement and knowledge management: Roda and Musulin (2014) introduced intelligent data analysis for sensor measurements of process dynamics. This approach can detect faults in monitored measurements.

Process supervision using knowledge management: Musulin et al. (2013) introduced a domain ontology on process supervision and fault diagnosis. It is based on the ISA88 and ISA95 Standards (ISA, 2006*b*, 2000). The system can determine the risk level of a hazard, with consequences and required actions, if applicable, using the support of reasoners and queries.

- Supporting Tools

Automated process modelling: Elve and Preisig (2017) introduced executable program codes using ontological modelling and generic model templates to produce process models.

Lagrange decomposition: Muñoz et al. (2015) solved the integration of supply chain planning and scheduling problems with the support of a knowledge management environment. An ontology plays a role in the integration of master and slave problems.

Modelling language: Hai et al. (2011) introduced a generic modelling language supported by an ontology to address operational process modelling.

Software support for chemical process engineering: Morbach et al. (2007, 2009) introduced a formal ontology (OntoCAPE) that can handle chemical process system information, mathematical relations, physical dimensions and processing units.

Finally, formally represented knowledge of PSE, which is based on conceptualisation, has introduced several advances in each subject, as shown in Table 2.5. The volume and diversity of the research on conceptual modelling in PSE provide an interpretive context that is not available on any subjects or topics. These subjects explicitly mention the commitment of conceptualisation. However, many other studies include conceptual modelling implicitly in knowledge-based models and methodologies.

Table 2.5.: PSE subjects and significant accomplishments in the area of conceptual modelling

Subject	**Topic**
Process and Product Design	Pharmaceutical product engineering Process synthesis Capturing and reusing process and product designs Pharmaceutical product development and manufacturing
Enterprise-wide Optimization	Risk management Competitive supply chain design Enterprise-wide scheduling Integration of planning and scheduling activities Semantics for industrial symbiosis networks Bio-refinery feedstock and technology matching
Production Scheduling	Knowledge management for scheduling activities Knowledge management in batch process and scheduling Reactive scheduling
Process Control	Sensor measurement and knowledge management Process supervision using knowledge management
Supporting Tools	Automated process modelling Lagrange decomposition Modelling language Software support for chemical process engineering

Chapter 3.

Thesis Outlook

This chapter concludes the overview of PSE challenges related to the thesis topics and the background of the challenges. A list of challenges for this thesis and their background connections are introduced in Section 3.1. Then, the chapters in which the challenges are addressed are identified in Section 3.2.

3.1. PSE Challenges and Background

The overview starts with the PSE perspective and outline of the thesis. Then, Chapter 2 introduces the background related to methods and tools focusing on decision making, conceptualisation, and PS. It is structured as follows:

Section 2.1 explains the DMMs that are widely used in the PSE area supporting the decision-making process.

Section 2.2 introduces the representations of physical, procedural, and process models of PS and their management.

Section 2.3 introduces the conceptual modelling methods and tools.

Section 2.4 introduces unified approaches of background which contains the integration of DMMs and the conceptual modelling applications in PSE.

Given the importance of these sections on the background, there has been relatively little research on these topics. However, the lack of research reflects the unified approaches that are taken. For instance, the difficulties in conceptual modelling lead to an interest in automated DMM building rather than training experts on individual and problem-oriented DMM construction. The challenges addressed in these topics are introduced as follows[1]:

---

[1] Each challenge has been tagged to be used in next section and they will be recalled in the final remarks chapter for the explanations of contributions.

- There have been many attempts to use ontologies in PSE (Marquardt et al., 2010; Muñoz et al., 2010; Trokanas et al., 2015; Zhou et al., 2017). Each study contains a manual identification step on concepts and relations to be included in the ontologies contained by a domain. The time and effort that are required are high and expensive, as each study is related to a thesis or a project. Therefore, a fast, accurate domain ontology construction methodology is required to identify concepts and relations for the domain of interest **(C1)**.

- Ontology construction is a complex procedure. Under some circumstances, it can be quite abstract due to (i) implementation on difficulties in different domains, (ii) the size of concepts and relations, (iii) the complexity of domains and (iv) validation. One way to facilitate the use of these methodologies is to improve the quality of the sources by detecting inconsistencies (checking consistency) **(C2)**. Additionally, outcomes of ontology construction methodologies (concepts and relations) may not be fully ready for use in practice, even though one of the criteria of ontology building is the usefulness of a produced domain ontology. One of the challenges is to guarantee the use of outcomes (concepts and relations) of ontology construction methodologies **(C3)**.

- Decision-making is a highly complex process that becomes even more challenging when decisions are made for systems that may have interacting or interdependent entities. One of the challenges for the PSE community is to solve the problems that appear in PS, which are designed, built and run to manufacture products. The complexity is even greater when these systems contain multi-level functional hierarchies. DMMs are required to solve decision-making problems that appear in these systems. The challenges relating to decision-making are to model the PS Domain more comprehensively than in previous PS Domain implementations **(C4)** and use this information to solve problems in multi-level functional hierarchies **(C5)**.

- Generic mathematical models have been the focus of PSE. The community has developed general formulations that can be applied to scheduling and planning problems and automated process modelling tools (Fedorova et al., 2015; Elve and Preisig, 2017). However, the challenge is to produce generic DMMs that can be used to solve new problem instances. This challenge requires a common understanding of the concepts that appear in DMMs that are more comprehensive than sets, parameters and the constraints that appear in mathematical programming. The challenges are to conceptualise DMMs **(C6)** and integrate the conceptual models of the DMM domain and the PS Domain **(C7)**.

- Automated construction of mathematical models is an emerging area, since support from computers and software is essential to save time and effort and to produce mathematical models. PSE has been using and developing many support tools such as GAMS (GAMS, 2016) and MOSAIC (Kuntsche et al., 2011), to assist with mathematical modelling.However, in the era of intelligent solutions, it is essential to think about how these models can be created automatically. Therefore, the challenge of automated DMM construction can be characterized as follows:

- modelling multi-level functional hierarchical activities **(C4)**,
- maintaining DMMs in a shared and common environment **(C6)**, and
- connecting those domains **(C7)**, using this information for automated DMM construction by developing algorithms and required sequences **(C8)** and solving these constructed DMMs **(C9)**.

The challenges (from **C1** to **C9**), which are shared and combined to build up previous items, are connected to related background as in Table 3.1.

Table 3.1.: Challenges of the thesis and their background

Challenge	**Background related to the challenge**
C1: An ontology construction methodology C2: Consistency checking of textual sources C3: Usage of ontology construction methodology outcomes (concepts and relations)	Section 2.2 (Process Systems) and Section 2.3 (Conceptual Modelling) have been the background.
C4: More comprehensive modelling of PS information within a common and shared domain C5: Usage of modelled information in solutions of multi-level functional hierarchies	Section 2.1 (Decision-Making) and Section 2.2 (Process Systems) have been the background.
C6: Conceptual modelling of DMMs C7: Integration of DMMs and PS conceptual models	Section 2.1 (Decision-Making) and Section 2.3 (Conceptual Modelling) have been the background.
C8: Development of algorithms and sequences for DMMs construction C9: Solution of these constructed models	Part I, Part II, and Part III have provided the background, knowledge and domains.

3.2. PSE Challenges and Addressed Solutions in This Thesis

The general structure of this thesis has been formulated considering the challenges introduced in the previous section. Table 3.2 introduces the connections between these challenges and the thesis structure. The thesis consists of five main parts: an overview, the PS Domain and conceptualisation, the CC Domain and conceptualisation, integrated exploitation of domains and final remarks.

Table 3.2.: Challenges and their address in the thesis

Challenge	**Addressed section in the thesis**	
	Part	**Chapter**
C1: A domain ontology construction methodology	Part II	Chapter 4: Systematic Domain Construction and Implementation
C2: Consistency checking of textual sources using an ontology construction methodology		
C3: Usage of ontology construction methodology outcomes (concepts and relations)		Chapter 5: Integrated Management of Process Systems
C4: More comprehensive modelling of PS information within a common and shared domain		
C5: Usage of modelled information in solutions of multi-level functional hierarchies		
C6: Conceptual modelling of DMMs	Part III	Chapter 6: Introduction to Constraint Conceptualization Chapter 7 Conceptual Constraint Domain Building
C7: Integration of DMMs and PS conceptual models		
C8: Development of algorithms and sequences for DMMs construction	Part IV	Chapter 8: Decision-Making Models Building Sequences and Algorithms
C9: Solution of these constructed models		

Part II.

Process Systems and Conceptualization

Chapter 4.

Systematic Construction of Conceptual Models: Domain Ontologies

A domain is the scope of an interest related to an area of knowledge; domains can be represented through domain ontologies, which are usually characterised not only by their specialisation but also by providing particular emphasis to certain types of relations. Building and maintaining consistent domain ontologies become advantageous for conceptualisation, management and sharing of knowledge. However, there is no one unique, efficient and systematic methodology to construct domain ontologies instead there are many ways depending on the sources and aims to build ontologies.

The determination of concepts and relations of domain ontologies may be achieved manually or with the support of automated tools; in both cases, the source of knowledge is usually based on textual sources (documents) technically describing the subject. However, depending on the domain dimensions, manual methods may be time-consuming. Conversely, automated tools may produce noisy and uncontrolled domain ontologies. Furthermore, outcomes may be inconsistent ontologies since the meaning of concepts in a technical domain usually differs from those used in common language. At this point, a systematic methodology to extract knowledge from normative documents would allow constructing domain ontologies in a fast and accurate way.

This chapter presents a novel methodology for building domain ontologies based on processing of normative documents (Section 4.2), application of the methodology to a standard (Section 4.3) and conceptual improvement of textual sources (Section 4.4).

4.1. Introduction

Knowledge conceptualization, management and sharing can be supported by ontologies which can be defined as “an explicit specification of a conceptualisation” (Gruber, 1993). Therefore, ontologies allow sharing a consistent view of structured information, which enhances re-usability and scalability. These characteristics of ontologies let them to build and share consistent models and methodologies. For instance, the Process Systems Engineering (PSE) community has a growing interest in knowledge management and integrating information across the enterprise control systems and decision-making procedures. There are several reasons to use ontologies in the PSE community since ontologies can be used as knowledge models for chemical processes (Muñoz, 2011), intelligent data analysis tools for databases (Roda and Musulin, 2014) and knowledge management models (Venkatasubramanian, 2009*b*) in addition to other level of applications such as intelligent software applications (Morbach et al., 2009). Additionally, the PSE community is interested in ontology building procedures from different aspects, for instance, for capturing and tracing ontology development processes (Vegetti et al., 2016) and ontology learning frameworks for the pharmaceutical product development (Remolona et al., 2017).

The construction of ontologies, particularly domain ontologies for specialised fields, requires experts to produce the ontology based on their knowledge. The task of building a domain ontology demands time and training since experts need to understand and extract the bases of the domain to transfer their knowledge to computers. There are some common strategies that expert can use for identifying concepts depending on their scope (i.e. from the most important ones to the most specific ones or reverse) (Corcho et al., 2003).

An emerging discipline facing domain ontology construction in a general way is Learning by Reading (LbR) that implies three processes: (i) deciding what has to be learned, (ii) deciding the order of learning and (iii) deciding source of knowledge (McFate et al., 2014). In addition to the LbR concerns, the method of learning is crucial (e.g. manual vs automatic, determination of concepts, being a concept in a domain).

There are also several automated tools to support to domain ontology construction by extracting concepts and relations from textual sources, for instance, OntoLearn (Navigli and Velardi, 2004) and Text2Onto (Cimiano and Völker, 2005). These tools are based on natural language processing; for instance, patterns introduced by Hearst (1992). However, an effort to filter the noise obtained from automated tools is required since the results may be uncontrollable for validation. One way to tackle the noise obtained from automated search is to work on concept extraction on the Internet or to develop methods to extract information from specialised textual sources.

Additionally, general mechanisms (extraction of concepts and relations) for building ontologies from documents fail during the construction of domain ontologies because of the lack of figures and table processing. The presence of non-textual information (e.g. tables, charts, figures) is not an easy task unless the use of tools developed for the non-textual information retrieval is appropriate and possible (Malmberg et al., 2011).

The differences among the domains characteristics as well as the differences among the corresponding specialised languages lead to a diversity of ontology building methodologies rather than a general approach. For instance, general approaches are using textual sources (e.g. an intelligent key-concept finder for ontology development (Kang et al., 2014), a semantic approach for extracting domain taxonomies from texts (Cimiano et al., 2003) and a linguistic approach that focuses explicitly on verbs (Meijer et al., 2014)). Additionally, the source could be changed from a textual source written in the natural language to computer oriented textual sources (e.g. lexical ontologies (Farreres et al., 2010), database schemas (Ra et al., 2012) or dictionaries as structured texts (Amar et al., 2016)). Also, applied domains may vary from the biomedical domain (Vivaldi and Rodríguez, 2010*b*) to the wind energy domain (Küçük and Arslan, 2014).

Domain ontology construction methodologies generally aim to find important concepts (also called terms) in the domain. However, concept extraction for domains can be complex since concepts in domains do not follow the same formation rules of the common language (Vivaldi and Rodríguez, 2010*a*). Additionally, these concepts should satisfy three conditions for a word (simple or multi-word) to be a concept in a domain: (i) valid combinations of morphological categories, noun, $noun\_noun$, $adjective\_noun$, $noun\_preposition\_noun$, (ii) higher frequency of occurrences in domain documents than general texts and (iii) existence in a domain application or document.

In parallel to the efforts for domain ontology construction, expert teams have devoted an effort to produce normative documents (e.g. standards, technical documents) to share domain knowledge; normative documents are considered for building a domain ontology since they define models and terminology, including the explanation of processes and data structures; they can be a solution of facing the third requirement related to the source of LbR. Therefore, the hypothesis is that intelligent selection of texts will reduce noise and allow fast and straight identification of concepts and binary relations. The textual source defining normative concepts seem very well suited to be taken as a source to synthesise manual and automatic approaches in the most efficient way. As a result, the proposed domain ontology construction methodology, the semi-automatic construction of domain ontologies from normative documents methodology (SECOND Methodology), is significantly different from other strategies by systematically combining both automatic and supervised procedures, using normative documents as sources for straightforward concepts and relations determination, generic for normative documents, and open to improvements against new advances on the subject.

This chapter presents (I) the proposed methodology in Section 4.2, (II) the application of the methodology in Section 4.3 and (III) the improvements of the normative documents in Section 4.4. Finally, Section 4.5 discusses the conclusions and implications of the proposed SECOND methodology.

4.2. SECOND Methodology

The input of the SECOND Methodology is a normative document that describes model and terminology of a domain (Figure 4.1). The selection of textual source overcomes and reduces the frequency issues: (i) noise coming from large documents, (ii) decision of domain relations on concept selection, and (iii) challenging decision on relations between origin and destination concept pairs. The decision of relation type in each cycle results in concepts of a domain.

The SECOND Methodology runs cyclically through the following steps: *(i)* extract relation, *(ii)* determine concept pairs, *(iii)* build branches, *(iv)* prune the ontology, *(v)* solve complex cases and *(vi)* continue with the extracting relations in *(i)* for the next cycle. Every cycle may result in some complex cases and in the step of solving complex cases; it may not be possible to solve issues arisen by the concepts and their relations. Therefore, an element called the complex case repository is added to the system for the collection of complex cases to be considered in the next cycles.

The methodology follows these cyclic steps owing to focusing one type of relation in each step and in order to provide a clear and consistent structure for the ontology. In each cycle, a new type of relation is processed with new concept and relation sets (also explained in Section 2.3.1 in detail).

The output of this methodology is a domain ontology, formally described by Expression 4.1 (the first time introduced in Expression (2.8)) and detailed in Table 4.1:

$$O = \{C, I, P, D, A\} \tag{4.1}$$

In this thesis, the set of concepts C and the set of object properties P are extracted for the construction of the ontology model, O. Instances, I, and data properties, D, which are

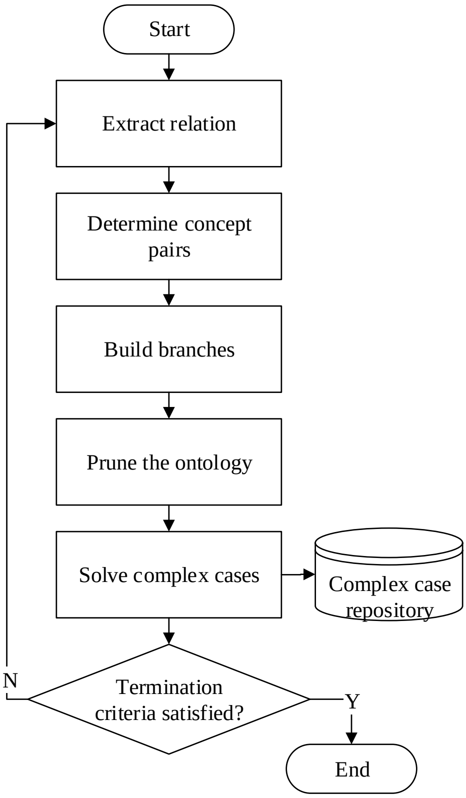


Figure 4.1.: Cyclic view of the SECOND Methodology

related to the real problems that can be identified in domain ontologies, are not introduced to this methodology. Additionally, axioms, A, are calculated only through the C and P.

4.2.1. Extracting Relations

In the SECOND Methodology, each cycle processes one relation type. In natural language, a relation among concepts is usually represented using a verb relating two noun phrases (NP). While the verb represents a relation type, the two related NPs represent origin and destination concept pairs [1].

---

[1] One use of object properties, P, in Expression 4.1 is to represent relations among concepts. Although n-ary relations may occur in textual sources (Banek et al., 2010), this chapter focuses on binary relations because they are significantly more frequent.

OWL syntax	Definitions
O	Ontology model
C	Set of concepts
I	Set of instances
P	Set of object properties
D	Set of data properties
A	Set of axioms defined over the union of all the members of the ontology

Table 4.1.: Ontology definitions

This relations type can also be expressed as object properties that assert general facts about the members of concepts. The SECOND Methodology focuses on relation types; relation types are introduced as follows:

$$P:\ C_1\ (Origin)\ \rightarrow\ C_2\ (Destination) \tag{4.2}$$

For extracting relations from a textual source, verb analysis must be executed to detect the possible list of relations in the set of each relation types. The number of different verbs in a text is a good estimator of the maximum number of possible explicit relation types that may be learned from a text. However, not all verbs will generate a relation type; several verbs may lead to the same relation type, or, even the same verb may generate different relation types.

The most significant relation type is the taxonomic relation ("is-a"), represented mainly by the verb "to be", which builds the skeleton of an ontology. Within an ontology, all concepts must be related via the "is-a" relation. For this reason, the SECOND Methodology proposes to always start with the "is-a" relation for the first cycle.

Additionally, much information is implicit in textual sources since they are written for human reading. Therefore, these non-explicit relations would not be automatically detected by a computer program and can only be solved during human supervision.

4.2.2. Determining Concept Pairs

To determine concept pairs, pattern matching and shallow parsing techniques have to be applied to the document. Patterns for each relation types and parsing details are given in the case study section. The purpose is to find places where selected verbs occur (which may represent the chosen relation type) and determine origin and destination concept pairs connected through these verbs. In each cycle, this step may introduce new concepts that have not been detected in previous cycles. Since a new relation type may connect a new concept to an existing one. So, from a $< P : C_1 \rightarrow C_2 >$ triple, at least one of the two involved NP has to correspond to an existing concept. Finally, determining concept pairs step produces concept pairs connected by the current relation type in the cycle.

4.2.3. Building Branches

The building branches step composes the domain ontology from the determined concept pairs using three phases (i.e. (i) introducing concept pairs, (ii) introducing non-textual information and (iii) introducing common-sense information).

4.2.3.1. Phase-1: Introducing Concept Pairs

Concept pairs are introduced into the ontology without any consideration; the software Protege has been used (Protege, 2015).

4.2.3.2. Phase-2: Introducing Non-textual Information

Usual mechanisms (extraction of concepts and relations) for building ontologies from documents fail during the construction of domain ontologies because of implicit and non-textual information (e.g. tables, charts, figures) unless there are exceptional tools developed for the non-textual information retrieval. Therefore, the human supervision during this step is crucial. A human expert extracts graphical information subject to the relation ruling the cycle; newly detected concept pairs are introduced to enrich the ontology further.

4.2.3.3. Phase-3: Introducing Common-Sense Knowledge

So far, the introduced concepts and relations from the previous step include neither common sense nor the knowledge shared within a domain. For a domain-specific ontology, common-sense knowledge from the domain has to be explicitly included in the ontology, as new concepts and relations.

4.2.4. Pruning the Ontology

The pruning process of an ontology aims to produce an established domain ontology with concepts and relation types.

One of the ways is to remove redundant relations when they are detected. For example, the elimination of transitive relations presented through other transitive relations.

Pruning also deepens (i.e. expands regarding deep) the taxonomy by moving the concepts from the top of the ontology to their proper places in the taxonomy (in the case of taxonomic relations).

Additionally, in this step problems that may be caused by the normative document writing are considered such as synonym, polysemy and adjective usage (discussed in Section 4.4.5). The synonym is used for enrichment and fluent reading of the text; however, the usage of more than one expressions for expressing the same concept introduces confusion regarding automated learning tools. The same issue arises in the polysemy, which is the usage of same expressions for different concepts. Again, the human experts can understand the usage of the polysemy since two or more different concepts may appear in the thinking. The inappropriate usage of adjectives may cause another issue for concept extraction; for instance, the comparative adjectives and adjectives with an antonym.

4.2.5. Solving Complex Cases

Complex cases are arisen by the lack of solutions to inconsistencies from the textual sources and cannot be solved during the processing of the previous steps. The resolution of these cases can be sustained with the human supervision supported by the expertise in managing the domain and the source. Complex cases may be detected by concepts being the origin or destination from/to multiple concepts within the same relation; for example, one concept being the origin of many destination concepts via “is-a” relation and the human expert should decide on which concept is the destination concept by taking into account the context of the source. Alternatively, it can be detected while building branches. When a complex case is detected, a segment (i.e. concepts from the root of the ontology from origin concept to the end of the relations to destinations concepts) should be investigated for the solution.

Each cycle essentially attends to solve the complex cases; however, they may not be solved with the currently available information that they need to be solved with the help of next relation types in the following cycles. Therefore, there is a complex case repository, which functions to collect all the cases that may be recalled in the next cases.

4.2.6. Termination Criteria

The cyclic methodology has an end depending on the purpose of building a domain ontology; the termination of the cycle is related to the LbR that how much to be learned. Termination criteria is a way to decide when to stop the cyclic relation type introduction may be determined considering the total introduced number of relations or relation types considering the complete document or maybe focus on specific relation types depending on the application purposes of the domain ontology.

4.3. SECOND Methodology Application: the ISA88 Standard Case

The presented methodology has been applied to ISA88 Standard as a case study. The ISA88 Standard is an enterprise control standard that regulates and describes models and terminology of data structure, recipes and production records (ISA, 2006*b*, 2001, 2003, 2006*c*). The standard includes models behind its text and figures as illustrated in Figure 4.2.

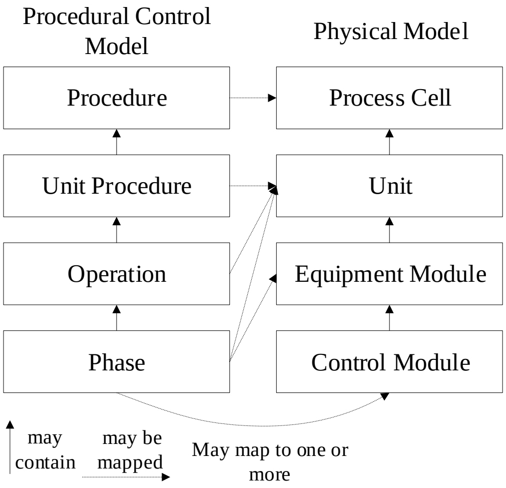


Figure 4.2.: Some models and relations from the ISA88 Standard (adapted from (ISA, 2006*b*))

The physical source of the ISA88 Standard is a complex PDF document (four parts, almost 400 pages) including page headers and footers (almost each page), figures with text (more than 100), tables (more than 100). Each sentence has to be tokenised and part-of-speech tagged. The tokenisation has been done using the Freeling tool-suite (Carreras et al., 2004).

4.3.1. Methodology Planning for Relation Types

The SECOND Methodology is a cyclic one where one relation type rules each cycle as stated before. Thus, verbs are used as relation encoders.

The list of different verb lemmas (not forms) has been extracted from the ISA88 Standard, giving a size of 96. Figure 4.3 illustrates a histogram with the frequencies of the topmost verb lemmas which collects different forms of the verbs as lemmas (e.g. “define” lemmas is a collection of “define”, “defines”, “defined” verb forms).

The first cycle starts with the taxonomic relation type represented by the verb “to be”[2]. The taxonomic relation type is the primary backbone relation in an ontology, related to the form “is-a” (e.g. “A reactor is a unit “). The verb “to be” has an individual frequency greater than 30% (see Figure 4.3). That is considerable compared with the verb “define”, which has a frequency of about 5%. This observation confirms that the verb “to be” should be chosen for the first cycle.

The relation type in the second cycle has to be as general as the taxonomic relation type, one conveying a relevant structural meaning. Candidates are “define” (5.43%), “use” (4,99%), “contain” (3.13%), and “include” (2.78%) as shown in Figure 4.3. Yet, frequencies have to be calculated by semantics, not by textual similarity. The verbs, “include” and “contain” convey the same meaning of structural composition, and they show the highest combined frequency of 5.91%. Thus, the meronymy relation type (“part-of”) has been chosen for the second cycle. Moreover, this frequency analysis has been conducted without including the figures and tables from the source document, despite the fact that the verb “contain” appears many times in the figures of the standard. The new relations that can be extracted from the figures increase the frequency of the selected relation type and supports the choice. While planning the second cycle, other less frequent verbs, also implying the meronymy relation type, have been considered too (as detailed in Section 4.3.3).

4.3.2. First Cycle: Taxonomic Relation Type

The first cycle extracts taxonomic relations resulting from a set of patterns that are detected within the text (Hearst, 1992). In some simple cases, this detection can be carried out by

---

[2] The “to be” verb contains “is”, “are”, “be”, “was”, “been”.

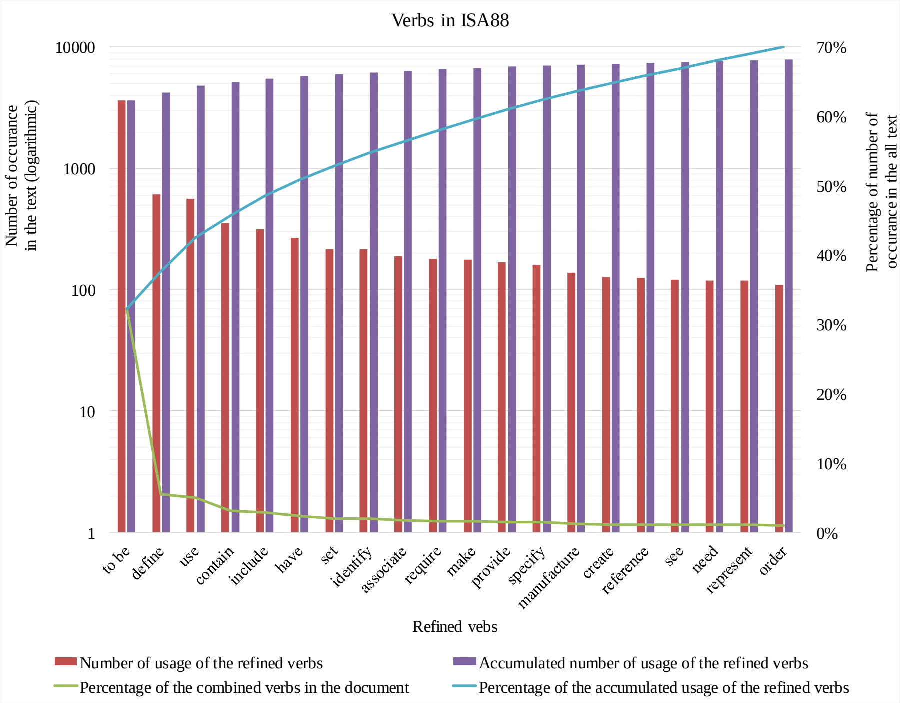


Figure 4.3.: Verb lemmas in the ISA88 Standard

applying plain pattern matching. However, more complex cases may require some detailed linguistic analysis. The pattern extraction has been achieved using a linguistic analyser, Freeling (Carreras et al., 2004), a pattern matching tool, the regular expression module of Python (Van Rossum, 1993), and a grammar parser, Pyparsing (McGuire, 2007).

4.3.2.1. Determining Concept Pairs

Three productive patterns proposed by Hearst (Hearst, 1992) have been used to detect the origin and destination concept pairs.

Pattern 1 (“is-a”): The pattern related to “is-a” is defined as:

$$Concept(Origin)\ is\ a\ Concept(Destination) \tag{4.3}$$

To extract this pattern and properly detect the NPs, a full syntactic analysis had to be performed. A concept is defined in equation 4.4 (?: zero or one, *: zero or many, +: one or many):

$$Concept = Article?\ Adjective^{*}\ Noun+ \tag{4.4}$$

Sentences have been processed to extract taxonomic relations whenever two NP with this pattern ("is-a") or some variant ("is-an", "is-the") appeared. For instance:

(i) The phrase "The general recipe is an enterprise level recipe that serves as the basis for lower-level recipes." is detected to match the pattern.

(ii) A linguistic analysis is applied to detect which NPs are connected through the "is-an" pattern.

(iii) Results: NP1("The general recipe") V("is") NP2("an enterprise level recipe").

Thus, the pattern NP1 V(is) NP2 allows deriving that NP2 is the destination of SN1, which in the example reflects that the concept "EnterpriseLevelRecipe"[3] is the destination of the "GeneralRecipe" concept for the taxonomic relation "is-a".

The application of Pattern 1 ("is-a") resulted in 104 candidate concept pairs.

Pattern 2 (definition): The first part of the ISA88 Standard is dedicated to the definition of normative terms. Thus, one pattern variant to detect taxonomic relations considers the special condition of the text in definitions, taking the form of "concept: definition"; the colon takes the place of the "is-a" construction in the previous pattern:

$$Concept(Origin)\ :\ Concept(Destination)+ \tag{4.5}$$

For matching this pattern, no linguistic analysis can be performed due to the lack of a fully structured sentence. Thus, a shallow parsing is performed to detect the first concept after the colon. For instance:

(i) "control recipe: A type of recipe ..." appears in the text.

(ii) The origin concept is the "ControlRecipe" and it is a 'type of recipe' that leads to the "Recipe" concept.

---

[3] In this thesis, concept are written using CamelCase representation and instances start with the lower case.

(iii) Thus, “Recipe” concept is the destination of this concept pair.

Pattern 2 (definitions) has resulted in 71 candidate concept pairs.

Pattern 3 (such as): In (Hearst, 1992) a pattern is proposed as “such NP as NP”. But in the text of the ISA88 Standard this pattern hasn’t been detected. Alternatively, a lot of “NP such as NP” are found in the text. Thus, the pattern has been adapted to this second case as follows:

$$Concept(Destination)\ such\ as\ Concept(Origin)+ \tag{4.6}$$

No linguistic analysis can be performed for this pattern since the analyser does not identify the “such as” construct as anything grammatically relevant. Thus, a pattern matching followed by a shallow tagging have been applied. For example:

(i) The phrase “... equipment entities such as units, equipment modules, and control modules.” matches the pattern.

(ii) The “EquipmentEntity” concept has been detected as the destination concept of each concept pair.

(iii) And the “Unit”, “EquipmentModule” and “ControlModule” are the origin concepts.

Pattern 3 (“such as”) resulted in 305 candidate concept pairs.

4.3.2.2. Building Branches

Phase-1: Introducing Concept Pairs: 480 candidate concept pairs between 633 candidate concepts have been extracted from introduced patterns (Each concept pair contains two concepts and these concepts can be used by other concept pairs).

A manual validation has been conducted, resulting in:

- 219 concept pairs - found correct,
- 71 concept pairs - found partially correct and needed manual edition,
- 187 concept pairs - found incorrect, and
- 3 concept pairs - remained undetermined.

The decision has been to continue with the correct and manually corrected concept pairs (290 concept pairs relating 334 concepts) which means a 60% of success rate.

Phase-2: Introducing Non-Textual Information: The ISA88 Standard contains figures (e.g. Figure 4.2) that cannot be processed via pattern matching due to the lack of figure recognition tools and algorithms. Thus, in this phase, graphical information which is extracted during the human supervision is introduced. During this process, new concepts are detected. Summary is given below:

(i) 7 concept pairs are detected,

(ii) 3 of the detected concept pairs were already introduced in Phase 1, and

(iii) 4 new taxonomic relations for the concept pairs are introduced (e.g. “Phase ‘is-a’ ProceduralElement”, “ProcessCell ‘is-a’ Equipment”).

Only the relation between these concept pairs are introduced since the concepts already appear in the ontology.

Phase-3: Introducing Common-Sense Knowledge: Rules expressed in following expressions are used to process the common-sense knowledge:

$$\mathrm{Noun}_{\mathrm{A}} + \mathrm{Noun}_{\mathrm{B}} \xrightarrow{\mathrm{is-a}} \mathrm{Noun}_{\mathrm{B}} \tag{4.7}$$

$$\mathrm{Adjective}_{\mathrm{A}} + \mathrm{Noun}_{\mathrm{C}} \xrightarrow{\mathrm{is-a}} \mathrm{Noun}_{\mathrm{C}} \tag{4.8}$$

New taxonomic relations are introduced following Expressions (4.7) and (4.8): For instance,

- the “UnitRecipe” concept, composed of two nouns (Expression (4.7)), is connected as an origin concept with destination to the “Recipe” concept
- “RegulatoryControl”, composed of an adjective and a noun (Expression (4.8)), is connected as an origin concept with destination to the “Control” concept.

141 concepts have been refined in this way and extra relations have been added to the ontology.

4.3.2.3. Pruning the Ontology

At this stage, 41 concepts with more than one taxonomic relation are detected; their unique topologies are individually analysed. The focus on the work is to concentrate on normative terms; for this reason, 26 cases do not appear in the normative terms have been considered irrelevant at this phase and not considered. In 5 cases, redundant relations (relations that can be achieved through destination concepts) are removed. One example of redundant relation removal is given in Figure 4.4. The figure shows the before and after versions of the “RecipePhase” concept.

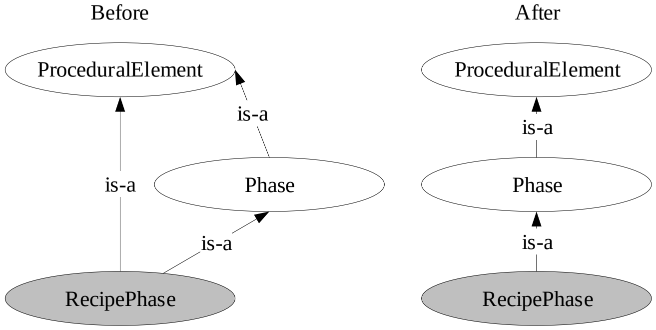


Figure 4.4.: “RecipePhase” concept before and after removing redundant relation

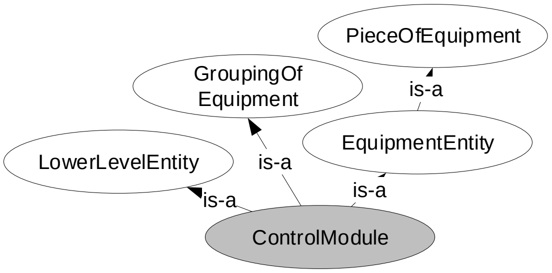


Figure 4.5.: “ControlModule” concept example before the pruning

Finally, 10 cases couldn’t be automated and needed human supervision to be solved (these 10 cases are discussed in Section 4.4.4). For instance, Figure 4.5 shows that the “ControlModule” concept has three destination concepts (“LowerLevelEntity”, “GroupingOfEquipment” and “EquipmentEntity”; “PieceOfEquipment” concept is also introduced). Only monitoring the full picture, there is enough information for the expert to make a decision on the pruning. The “LowerLevelEntity” concept is using a comparative adjective that introduces ambiguity to the text and it is not clear if the root concept is “LevelEntity” or “Entity”. Considering the suggestion in the guideline (Guideline: Avoid using comparative and unnecessary adjectives - from Section 4.4.5), the decision related to the “LowerLevelEntity” has been to remove the concept from the ontology.

As a consequence of inspecting this example, it has been clear that many concepts in the standard have been used as synonyms. “PieceOfEquipment” and “GroupingOfEquipment” give the impression of different ways of naming “EquipmentEntity”. Instead of removing these concepts, they have been kept, but marked as synonyms (Figure 4.6).

4.3.2.4. Solving Complex Cases

The “GeneralRecipe” concept has been chosen as an example of a complex case since the taxonomic type of relation appear more than one time from the “GeneralRecipe” concept to other six concepts. There are decisions to be made by human experts considering the whole document. Figure 4.7 shows the “GeneralRecipe” segment, which is a taxonomy up to the topmost element “Thing” so that not only six concept but also their relations are introduced to the solution of the complex case.

The “GeneralRecipe” concept has 6 different taxonomic relations and they have been solved as follows:

(i) A redundant relation to the destination concept “Recipe”.

- This relation has been removed.

(ii) A relation to the destination concept “Container”.

- This relation has been send to complex case repository to be solved in solution of complex cases in the second cycle with the “Combination” concept (see Section 4.3.3.4).

(iii) Three relations to the destination concepts “EnterpriseWideRecipe”, “EnterpriseLevelRecipe”, and “CorporateRecipe”

- All these concepts appear in the text a few times. It is clear that they are not as important as the “GeneralRecipe” concept. Yet, they are used in the document as a synonym of the “GeneralRecipe” concept. So, relations are changed to “is-synonym.”

(iv) The last relation is to the destination concept “EquipmentIndependentRecipe”.

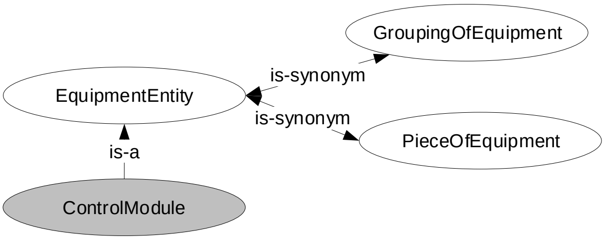


Figure 4.6.: “ControlModule” concept example after the pruning

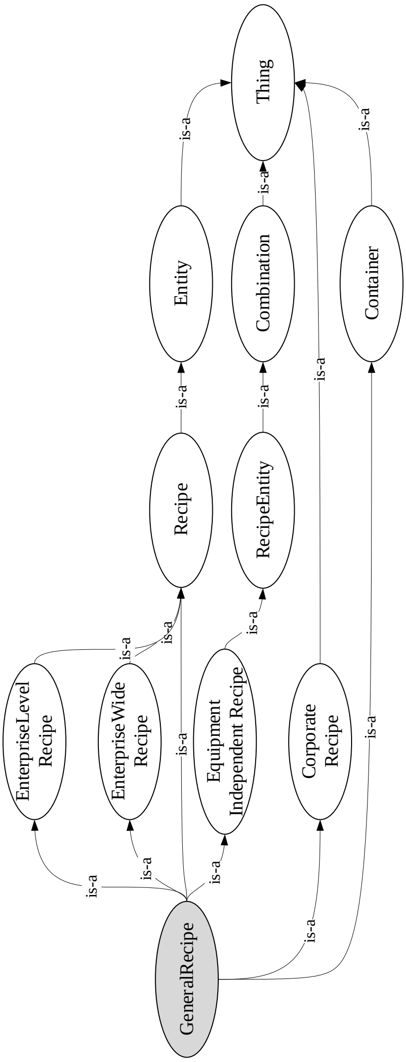


Figure 4.7.: “GeneralRecipe” segment before taking decisions

- The synonym relation has not been introduced there since “EquipmentIndependentRecipe” has one more origin relation (“GeneralRecipe” and “SiteRecipe”); the “SiteRecipe” concept is a different concept than the “GeneralRecipe” concept.

Additionally, the document uses the “RecipeEntity” concept as “Recipe” itself. So, another “is-synonym” relation has been defined between these concepts. Finally, Figure 4.8 shows the solution of the complex case of “GeneralRecipe”. In the case of solving complex cases, decisions are not only applicable to the main concept but also other connected concepts (e.g. "Recipe" and "RecipeEntity" concepts). Also, it is important to keep the synonym to capture more relations from the normative document since the concepts with synonym relations produce more than one relation.

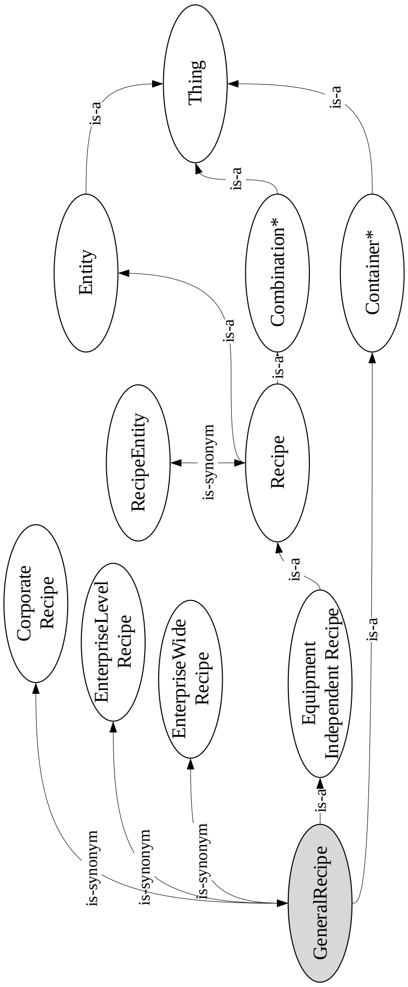


Figure 4.8.: "GeneralRecipe" segment after taking decisions in the first cycle

4.3.3. Second cycle: Meronymy Relation Type

As a result of the verb frequency analysis in Section 4.3.1, the meronymy relation type ("part-of") is chosen in this second cycle; this is not a simple relation type since the usage can be considered as a member of a collection (Table 4.2), an object contained within a container, and being part of a whole.

Although all meronymic candidate relations have been extracted, the validation has been restricted to those normative concepts resulting from the first cycle. Thus, new concepts detected from patterns are left for further developments; only new part-of relations detected for the current set of concepts and new concepts appear in figures are introduced since this work does not report another cycle and focuses on the methodology details.

4.3.3.1. Determining Concept Pairs

For the concept pairs determination of the meronymic relation type, patterns are suggested considering the main expression/verb of meronymy ("part-of" for the "to be part of"), the most frequently used meronymic relation from the verb lemmas (i.e. "contain" and "include"), and another verb lemma "consist"[4] suggested by Girju et al. (2006).

Pattern 4 ("part-of"): This pattern finds concept pairs that are related by the construct "part-of" within the text using following expression (Concept is characterized by Expression 4.4):

$$Concept(Origin)\ part-of\ Concept(Destination) \tag{4.9}$$

For example, the sentence "... A unit procedure that is part of equipment control..." matches with the pattern. The "UnitProcedure" concept is defined as the origin concept of meronymy relation, and the "EquipmentControl" concept is defined as the destination concept of meronymy relation.

Pattern 4 ("part-of") resulted in 79 candidate concept pairs for meronymy relation.

Pattern 5 ("includ"):* This pattern searches the "includ*" verb form within the text and connected concepts by this verb as follow:

---

[4] The verb lemma "consist" is added to the pattern list even though the occurrences amount the verbs is less than 1% to introduce the idea of verb clustering under different verbs.

$$Concept(Origin/Destination)\ includ^{*}\ Concept(Origin/Destination) \quad (4.10)$$

The verb forms such as “included”, “includes”, “including” fall in this pattern build from the “include” verb lemma. Both concepts appear in Expression 4.10 may be appear as origin or destination concepts since there is a passive form of the verb is fit into this pattern (this is also the same for the contain* pattern). For example, the definition “... The formula is a category of recipe information that includes process inputs, process parameters, and process outputs.” is detected by the pattern. The “Formula” concept (Destination) includes (or is composed by) three other concepts (Origin): “ProcessInput”, “ProcessParameter”, and “ProcessOutput”. That is, these three concepts are parts of a formula (see Figure 4.9).

Pattern 5 (“includ*”) resulted in 169 candidate concept pairs for meronymy relation.

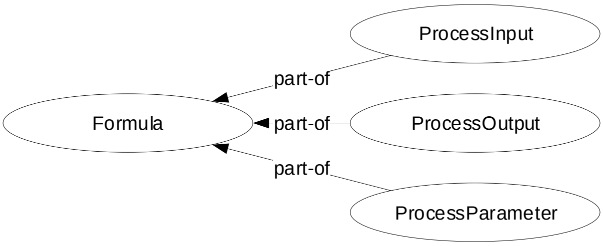


Figure 4.9.: “Formula” concept example resulting from pattern 4

Pattern 6 (contain):* This pattern finds concepts in a phrase related by the construct “contain*” that codifies a meronymy relation and it can be expressed as follows:

$$Concept(Origin/Destination)\ contain^{*}\ Concept(Origin/Destination)+ \quad (4.11)$$

Words such as “contain”, “contains”, “contained” fall in this pattern. For example, The sentence “... An area may contain process cells, units, equipment modules, and control modules.” matches the pattern. Thus, the parts of the destination concept “Area” are the origin concepts “ProcessCell”, “Unit”, “EquipmentModule”, and “ControlModule”.

Pattern 6 (“contain*”) resulted in 193 candidate concept pairs for meronymy relation.

Pattern 7 (consist):* This pattern finds concepts within a phrase that are being related by the “ consist*” verb form within the text as in the expression:

$$Concept(Destionation)\ consist^{*}\ Concept(Origin)+ \tag{4.12}$$

Words such as “consist of”, “consists”, etc. of fall in this pattern. For example, the sentence “... Each process stage **consists of** an ordered set of one or more process operations...” matches this pattern. Concluding that origin concept “ProcessOperation” is part of the destination concept “ProcessStage”.

Pattern 7 (“consist*”) resulted in 26 candidate concept pairs containing meronymy relation.

4.3.3.2. Building Branches

Phase-1: Introducing Concept Pairs:

The patterns have generated 346 candidate concept pairs among 177 candidate concepts. The evaluation of meronymy relation type has required the careful participation of the expert because the phrasing is more variable than in the case of taxonomic relation type. The same phrase can codify more than one instance of meronymy relation, and it is difficult to establish automatically whether a concept is an origin or a destination in some of the patterns.

After the evaluation, 254 relations have been obtained (92 manually added) between 205 concepts.

Phase-2: Introducing Non-Textual Information:

The procedure has been applied as in the first cycle: information extracted from graphics is included during the human supervision. During this process, new concept pairs and concepts are detected:

(i) 22 concept pairs are detected,

(ii) 8 of the detected concept pairs are already present in the ontology, and

(iii) 14 relations are introduced in the ontology (e.g. “UnitProcedure ‘part-of’ Procedure”, “ProductSpecificInformation ‘part-of’ GeneralRecipe”) .

(iv) 6 new concepts are added to the ontology (e.g. “BatchSize”, “OperatorSystem”, “GeneratedInformation”, etc.).

Phase-3: Introducing Common-Sense Knowledge:

New “partOf” relations are introduced following the rule in Expression (4.13). 133 candidate relations among 176 different concepts have been detected.

$$\mathrm{Noun_A} + \mathrm{Noun_B} \xrightarrow{\text{partOf}} \mathrm{Noun_A} \tag{4.13}$$

Result of the expression:

- 13 relations were already present in the ontology (i.e. 6 relations directly, 7 relations because of their super classes), for instance, “RecipeFormula partOf Recipe”, “RecipeInformation partOf Recipe”, ’BatchProductionRecordReport partOf BatchProductionReport”, etc.
- 35 new relations have been found correct such as “FormulaValue partOf Formula”, “RecipeElement partOf Recipe”, “VersionNumber partOf Version”, etc.
- 85 relations have been declined, for example, “RecipeManagement partOf Recipe” (management is an activity), “EquipmentRequirement partOf Equipment”, “SiteRecipe partOf Site” (SiteRecipe is part of the recipe model and Site is part of PhysicalModel), etc.

Another particular case to “part-of” relations arises from aggregation concepts (e.g. “Container”, “Combination”, “Collection”). They appear as destination concepts of taxonomic relations after the first cycle; these aggregation concepts express “being the composition of other concepts” and this is the information to be introduced to the domain. For instance, knowing that “a concept is a container” is not useful unless “what is contained” is known. In this case study, 24 concepts have been refined in this way, transforming them into proper meronymic relations. Table 4.2 introduces the summary of aggregation concepts.

An example of the applied logic to the “RecipeEntity” concept is illustrated as follows:

(i) The detected relation is “RecipeEntity ‘is-a’ Combination”.

(ii) The previous relation is detected from this sentence: “... recipe entity: the combination of a procedural element ...”.

(iii) The relation between “RecipeEntity” and “Combination” concepts is removed.

(iv) The concept “Combination” is removed.

(v) One relation is added: “ProceduralElement ‘part-of’ RecipeEntity”.

Name of the concept	Number of origin concepts (removed relations)	Number of introduced relations	Number of new added concepts
Container	2	6	5
Collection	6	9	4
Combination	2	4	3
Component	2	0	0
Group	1	1	0
Grouping	2	6	2
List	2	0	0
Set	2	0	0

Table 4.2.: Removing aggregation concepts

The aggregation concepts also lead to an issue in the document related to the "Unit" concept. The detected relation in the first cycle is "Unit 'is-a' Collection"; the relation is detected from this sentence: "... unit: A collection of associated control modules and/or equipment modules ...". According to the rule the "ControlModule 'part-of' Unit" and "EquipmentModule 'part-of' Unit" relations should be introduced. However, domain experts detect that "Unit" concept has been used as two different concepts ("Equipment" and "EquipmentEntity"). In this case, the mentioned "Unit" concept implies the "UnitEntity"[5].

4.3.3.3. Pruning the Ontology

The pruning of the ontology in terms of meronymy can be exemplified with the "Formula" concept segment. Actions must be taken to prune the part of the ontology shown in Figure 4.10. The behaviour of "part-of" relation is transitive, a fact which allows to remove some of the existing relations without losing information, resulting in Figure 4.11.

4.3.3.4. Solving Complex Cases

"GeneralRecipe" segment: The first part of this segment has been introduced in Section 4.3.2.4 (the last status is in Figure 4.12) and kept in the complex case repository; the same complex case that is to further investigate with the guide of new information appear in the second cycle.

In this cycle, the critical point is the aggregation concepts (i.e. "Container" and "Combination") which have resulted in introducing more concepts to the segment.

For the "Container" concept:

---

[5] This arisen issue in the ISA88 Standard has been corrected; the standard has changed the mentioned concept to "UnitEntity" in the newer version of the standard (ISA, 2010)

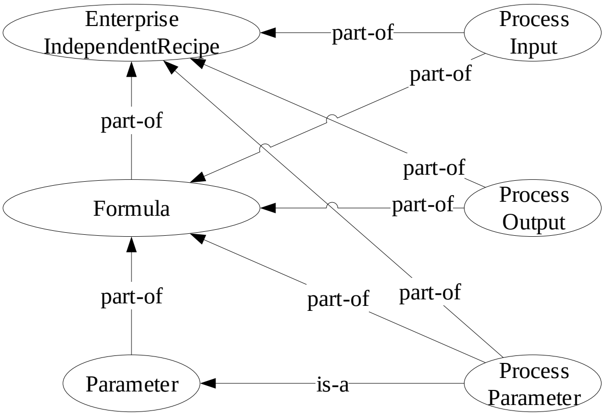


Figure 4.10.: Pruning example of the second cycle (before)

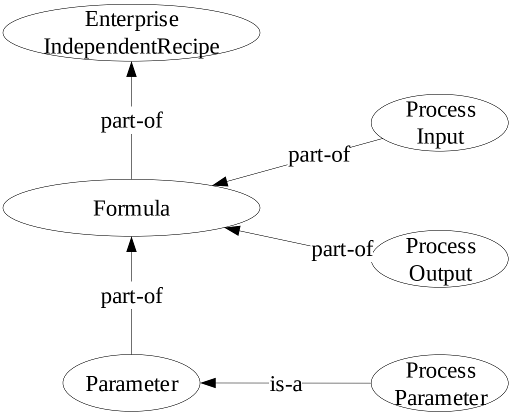


Figure 4.11.: Pruning example of the second cycle (after)

- The current relation is “GeneralRecipe ‘is-a’ Container”.
- This relation is taken from this text: “... A general recipe is a container of production information ...”.
- The text actually produces a meronymy relation from the “ProductionInformation” origin concept.

For the “Combination” concept:

- The current relation is “RecipeEntity ‘is-a’ Combination”.

- This relation is taken from this text: "... recipe entity: the combination of a procedural element ..."
- The text actually produces a meronymy relation from the "ProceduralElement" origin concept.

Finally, Figure 4.12 shows the final diagram of the "GeneralRecipe" segment.

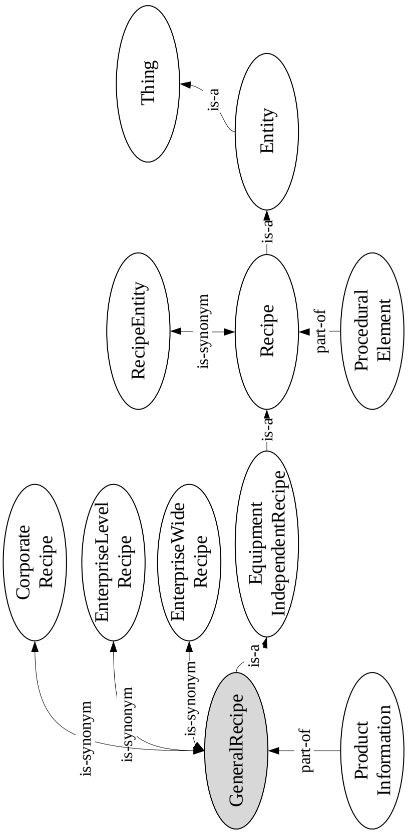


Figure 4.12.: "GeneralRecipe" segment after the second cycle

The "Entity" concept is left for the next possible cycles because of human experts suspicion about a polysemy case; it may cause two different concepts.

4.3.4. Summary of the ISA88 Case Results

One general topological aspect taken into account to measure the performance of the methodology is the width and depth of an ontology. The very first task of the first cy-

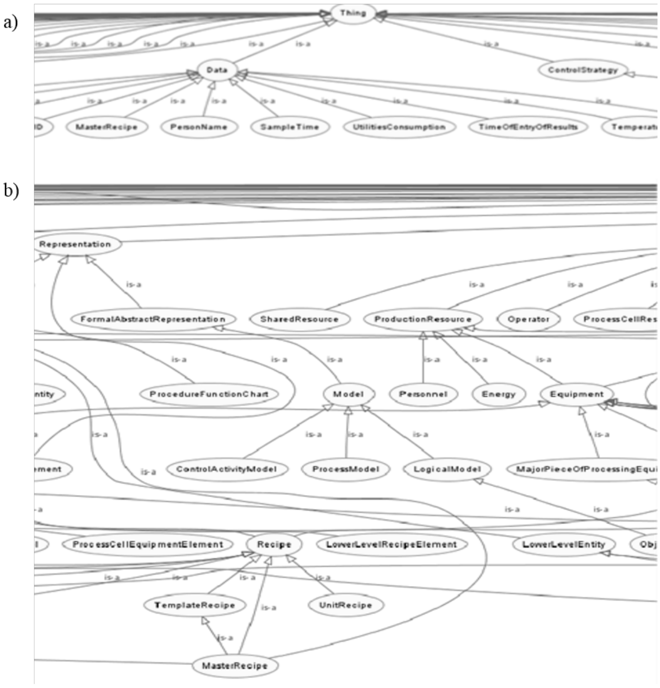


Figure 4.13.: Depth of the ontology in terms of taxonomic relation type a) after building branches from the first cycle b) end of the first cycle

cle has been to start constructing the Process Systems Management (PSM) Ontology by introducing concept pairs with taxonomic relations. This task has resulted in a flat ontology as shown in Figure 4.13a. Yet, other phases and the second cycle have introduced more connections between concepts, resulting in Figure 4.13b, illustrating a broader ontology which may be considered as a sign of quality and improvement. A detailed view per step is shown in Figure 4.14, illustrating the number of concepts per level at each step; it shows that depth seven is only reached after step 5. In the end, the ontology has 499 concepts with 689 taxonomical relations and 2446 total relations cause by both meronmy and taxonomical relation types. The ontology has 11 “is-synonym” relations between concept pairs. The resulting concepts for PSE Ontology is given as the supplementary material at Appendix D.

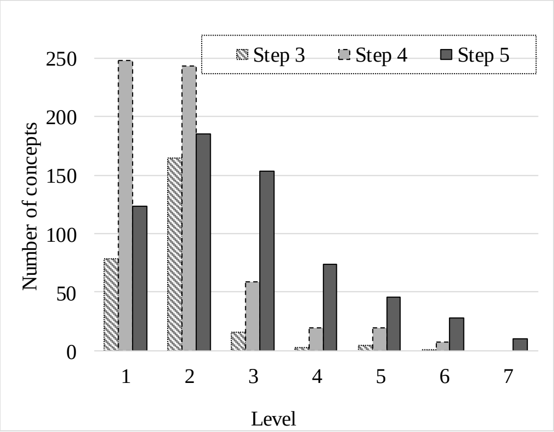


Figure 4.14.: Number of concepts per level

4.3.5. Comparison with Other ISA88 Sourced Ontologies

There are two ways to compare ontologies: quantitative and qualitative. The quantitative comparison uses the numbers and concept names and relations between ontologies to give an idea about the size of an ontology. However, the quality comparison of the ontologies is not straightforward since other factors like complexity can affect the comparison.

Two currently available ontologies: (i) produced manually for different purposes from the ISA88 Standard: BaPron Ontology (Muñoz, 2011) and (ii) Henning Ontology (Vegetti and Henning, 2015) are considered for the comparison of the SECOND Methodology outcome (i.e. PSM Ontology). Table 4.3 introduces the number of concepts and branches for each ontology. The volume of the Process Management Domain Ontology is expected to be higher since it includes automated steps for a fast procedure. of the steps presented in this work has been executed automatically. On the other hand, one of the critical comparisons is to match common branches in the ontologies. Table 4.4 shows the intersections of three ontologies. It is interesting to note that the PSM Ontology contains branches in both ontologies; however, manual ontologies do not share any branch. Even though these ontologies use the same source (ISA88 Standard), it is understandable that manual ontologies are built for specific purposes, and the current version of the PSM Ontology is built document-oriented.

Comparison between manual and automatic methods is difficult in qualitative terms. Suitable metrics should be proposed and used, including the effort and the quality of the ontology obtained. Qualitative analyses may provide an estimation of the completeness of the ontologies regarding size. It is clear that the ontology produced by the SECOND

Table 4.3.: Number of concepts for each ontology

Ontology	**Number of Concepts**	**Number of Relations (concept pairs)**
PSM Ontology	499	689
BaPrOn (Muñoz, 2011)	281	156
Henning Ontology (Vegetti and Henning, 2015)	68	51

Table 4.4.: Detected common branches between the PSM, Bapron and Henning Ontologies

Concept Pairs	**PSM Ontology**	**Bapron**	**Henning Ontology**
ControlActivity - RecipeManagement	✓	✓	
CoordinationControl - Allocation	✓	✓	
Parameter - ProcessParameter	✓		✓
ProceduralElement - Operation	✓	✓	
Procedure - EquipmentProcedure	✓	✓	
Resource - ManPower	✓		✓
Resource - Material	✓		✓

Methodology is producing a more significant ontology with a lower development effort. Therefore, qualitative results of the ontologies are reported considering the current developments based on the previously mentioned ontologies. Recently, the BaPrOn Ontology has been used to support semantic modelling of recipe based knowledge management in (Muñoz, Capón-García and Puigjaner, 2017). On the other hand, the Henning Ontology supports the software engineering for scheduling systems Henning (2017*b*).

4.4. Improvement of Normative Documents

The use of normative documents (e.g. standards and technical documents) is one of the modern accomplishments of industrial activities (Brayton et al., 2015). In the era of research and mass production, a huge number of products are being manufactured by multiple devices performing the same operation (Jung et al., 2016). One of the strengths of standardization is the provision of a context where devices can be interchanged for performing the same task, as all of them will understand instructions in the same way. That is, they share the same view of reality: the one that is established by the standard (ETSI and European Telecommunications Standards Institute, 2015; ISO and International Organization for Standardization, 2015; IEC and International Electrotecnical Commision, 2015). Furthermore, there are various guidelines for writing standards (ISO and International Organization for Standardization, 2014; IEEE and Institute of Electrical and Electronics Engineers, 2014; ISO/IEC GUIDE 17:2016, 2016), aiming to define a way of explaining things that strive for precision and avoid ambiguity; yet, they do not discuss on conceptual models and modelling.

It is generally accepted that semantic modelling is a mature technology in automation. It is broadly used as a unifying tool when managing sequential function charts (Bauer et al., 2004), for ensuring systems integration and operability (Pauwels and Terkaj, 2016), to provide connection between models with standards (BS-ISO/IEC 25010:2011, 2011), or to integrate meta-models in control loops (OMG, 2008). Moreover, it is also studied to improve the event-driven control systems since the interpretations from different team members in batch industry exist for creating specifications (Sanchez et al., 2002). However, the capacity to ensure a very high level of conceptual consistency throughout the system plays an essential role in the implementation (Godena, 2009). This is also true for the concepts defined by the ISA88 Standard (Schaefer, 1996), which is the representative case addressed in this study. Towards this end, this work provides a first approach to the construction of conceptual models from technical standards, as well as an innovative reverse approach, which is the systematic analysis and improvement of the text of the standard using the semantic domain knowledge.

Domain ontology development is an active research area nowadays. There are works applied to other domains to create ontologies (Henderson-Sellers et al., 2014; Gonzalez-Perez et al., 2016; Vegetti et al., 2016), but it is hard to find applications in a same domain allowing rigorous comparative studies. Next, paradigms are presented in this section to contextualize, discuss and explain the formalization of conceptual models.

4.4.1. Paradigms

The effort of writing a standard for automation is overwhelming. It requires the team effort of the best experts in a domain to agree and to describe a model of how systems and processes work in the most clear and precise way. In order to explain the current situation and propose improvements, three paradigms are next discussed (see Figure 4.15).

Paradigm 1 illustrates the current state of the art in standard development. Currently, domain experts have to make agreements for writing normative documents. A conceptual model needs to be fit to their particular views of the world, and outlying views need to be identified and discarded, in the same way equations are fit to data. This conceptual model has to be flexible enough to allow all the different visions of the problem, but precise enough to prevent misinterpretations. Once such technical document is written, approved and available, area experts read it and strive to produce standard-complying developments. Thus, automation experts develop automation code according to their own interpretation of the standard and to a specific purpose to solve an automation problem.

Paradigm 1 poses three main problems (Figure 4.15). The first one is the inherent ambiguity of natural language (NL) in which the document defining the standard is founded. Even the most careful technical writing suffers a certain degree of imprecision. The second problem is the interpretation of the text, which is done by the reader based on her/his own understanding and her/his pre-conceptions, as well as on the purpose behind the reading effort. The third problem is that the reader (i.e. the area expert) should also be the code developer. Having these two proficient technical profiles is unlikely, and separating them is desirable.

To avoid the problems of Paradigm 1, Paradigm 2 is suggested and depicted in Figure 4.15. The method presented in this paper is depicted as the second paradigm, which supports conceptual consistency of already established technical standards. Hence, a semi-automatic methodology enables developing an ontology from the text of the standard and the aid of a team of area experts understanding the standard. The aim of this procedure is to establish a sole interpretation of the world, so that different codes can be developed for different purposes, just like in Paradigm 1, using the ontology as the underlying knowledge source. Accordingly, area experts, who interpret the standard to create the ontology, and code creators, who only need the ontology as input for their automation code, have separate roles. It is worth mentioning at this point other efforts for creating conceptual representations of standard systems such as B2MML (MESA, 2018) and SysML (Object Management Group, 2017).

Regarding further development, this paper suggests a further evolution step into Paradigm 3 (Figure 4.15). Once a team of domain experts meets and commits to compile a text for

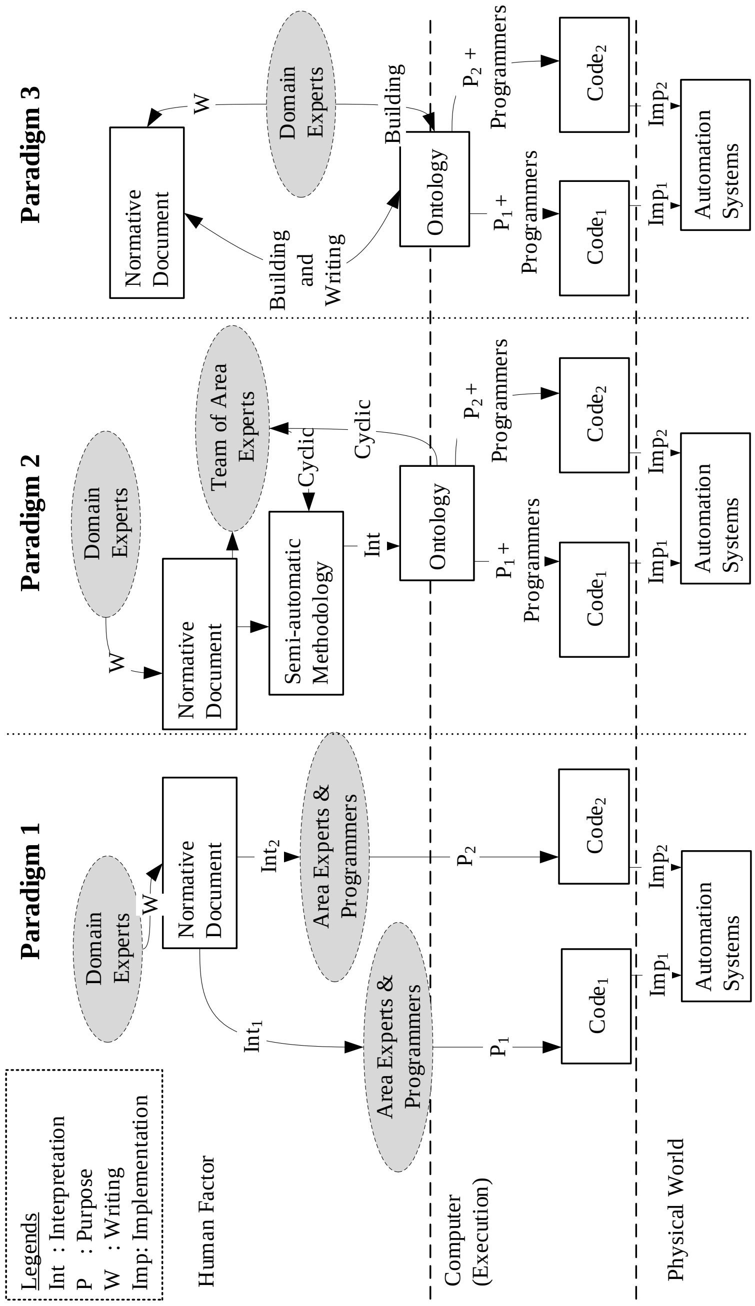


Figure 4.15.: Demonstration of explained paradigms

establishing a standardized view of a domain, the effort of developing an ontology in parallel is easily affordable, while the benefit would be significantly increased. The ontology establishes without the inherent ambiguity of NL one concrete view of a domain. In a single step, two standard contents could be established for a consistent interpretation of reality: one for humans to read, in NL, and another readily usable for automation without ambiguities.

Three main advantages can be envisaged for the application of the proposed methodology depicted in Paradigm 2. First, a normative document can easily lead to an ontology of the domain helping programmers to develop standard-complying software. Secondly, an ontology can help domain experts to interpret a normative document in a precise and simpler way. Finally, a semantic analysis based on the ontology may produce useful feedback for producing an improved version of the normative document written in NL. This feedback from the ontology to the written document is a novel and an original approach to enhance standardization in the field of automation.

4.4.2. Why Ontologies for Normative Documents?

Any text written in NL is intrinsically ambiguous and imprecise. NL is not the best tool for establishing a view of a domain that is the result of an agreement among the experts in technical domain, but it has been the best available tool for the latest millennia. If the intention is to transmit the knowledge to other persons, NL is the best medium available (Noy and McGuinness, 2001). If NL is considered as a way of representing events or entities of the real world in terms of human language, the mapping is not bi-univocal: a linguistic unit (a word, a phrase, a sentence) can refer to different real world units (polisemy) while a real world unit can be uttered in NL using several variants (synonym). Table 4.5 summarizes the main differences between textual documents and ontologies. Given NL inherent ambiguity, there is no mechanism to validate and ensure that the mental models built by different readers from the same text match. Thus, texts seem not enough for automation systems; they are intended and satisfactory for transmitting knowledge to humans.

Table 4.5.: Text and ontology comparison

Text	Ontology
1a: ambiguity of NL: polysemy, synonym, ellipsis, implicit knowledge	**1b:** formal and strict definition of concepts, meanings and relations
2a: need for interpretation (while reading)	**2b:** no need for interpretation
3a: different interpretation for each human	**3b:** one interpretation always and for everyone
4a: written for humans, unusable by computers	**4b:** structured for computers, complex for humans

Ontologies, on the other hand, are formal representations of a domain (Gruber, 1993). They do not carry the problems of synonym, polysemy, ellipsis or implicit knowledge. Ontologies represent concepts in a unique way by means of their relations to other concepts. Although concepts may be identified by words of NL, ontologies allow differentiating only one sense among all the senses a word may have: synonym has to be represented as a relation between concepts. Ellipsis or implicit knowledge does not occur in an ontology, all knowledge must be explicitly represented, due to the closed world assumption usually accepted.

Researchers are using ontologies for coordinating multiple standards and models (Pardo et al., 2012). In the process of writing a standard, the use of ontologies from the very beginning removes the effects of inconsistencies between models, the confusion resulting from synonym, and eventual terminological conflicts not only between different hierarchical levels, but also between different standards. In the field of PSE, several works (Morbach, 2009; Venkatasubramanian, 2009*a*; Muñoz et al., 2010; Cecelja et al., 2015; Muñoz et al., 2015; Henning, 2017*b*) have recently used ontologies to

(i) build intelligent software systems,

(ii) use them as a support tool for systematic analysis of data,

(iii) use them as a knowledge management tool and

(iv) address an emerging technology to look beyond the traditional modelling and solution methods.

This section presents results circumscribed to Paradigm 2 in Figure 4.15, obtained from processing the text of ISA88 and extracting an ontology representing the knowledge in it. Undefined concepts, extended use of synonym, and misleading use of adjectives have been detected in the document, which have been identified as problems, not only for the implication of automatic processes, but also for human readers. Thus, the contribution is an analysis and suggestions for developing normative standards.

4.4.3. Application: Analysis of the ISA88 Standard

This section presents the analysis of inconsistent cases performed after the development of the ontological model as described in the previous section. The study has been limited to the normative concepts (i.e. the concepts defined in the standard) and it allows suggesting ways to enhance the procedure for creating standards by means of additional guidelines of good practices. For comparative purposes, this work uses the 2006 and 2010 versions of ISA88 Part 1 (ISA88R2006 (ISA, 2006*b*) and ISA882010 (ISA, 2010)), and the latest version of Part 2 (ISA, 2001), Part 3 (ISA, 2003) and Part 4 (ISA, 2006*c*). For the sake of clarity, abbreviations and definitions are given in the glossary section.

Only the taxonomy of the ontology is considered and misused concepts and definitions are detected by using the methodology. Detecting inconsistencies results in 41 issues (concepts with more than one parent) that are detected from the taxonomy (ontology skeleton). 26 of them are excluded from the list of concepts not relevant to batch control. 10 concepts are reported in this paper, as they are normative concepts and lead to different discussions.

The rest of the examples are also solved with the subsumed relation approach, which removes the unnecessary relations that can be inferred through other relations (e.g. "QualityInformation is-a Information", "QualityInformation is-a ProductionInformation" and "ProductionInformation is-a Information" at the same time: this is solved by removing the relation from "QualityInformation" to Information since it can be inferred through the "ProductionInformation" concept). Next, the "Parameter" concept is shown as a specific example presenting no problems in Figure 4.16. In the "Parameter" example, all the relations between the concepts are clear and do not lead to any confusion or ambiguity.

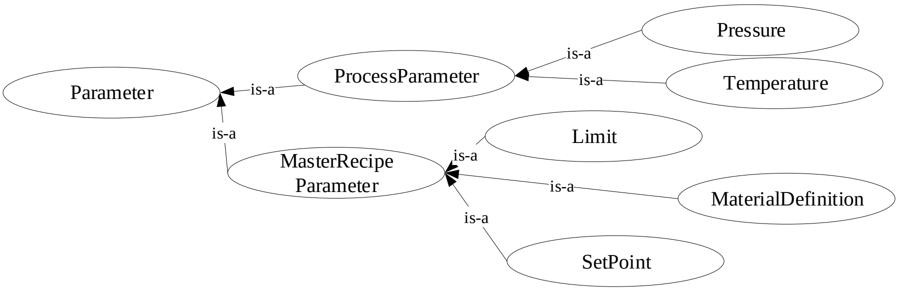


Figure 4.16.: "Parameter" concept and its taxonomy

4.4.4. Analysis of Detected Inconsistencies from the Ontological Point of View

Detected cases and solutions are explained in this section. All cases are selected from normative concepts with two or more super-concepts where an inconsistency can be detected and an outline for explanation of these cases is given as follows:

(i) The pattern matching has located a set of phrases in the text of ISA88 Standard that define or add information to the concept classification. They are shown in Tables 4.6

and 4.7. In the table, bold words represent concepts and underlined words represent connectors in the patterns except definition pattern, which is depicted with ":".

(ii) Concept names are distinguished from normal text using CamelCase in this thesis.

(iii) Issues related to the selected normative concept from ISA88R2006 are presented in the following subsections, see also the corresponding fragments in Tables 4.6 and 4.7.

(iv) Discussion on the solution after identifying the problem is given and the revised ontology view is depicted in a figure.

Table 4.6.: Selected cases from the ISA88 Standard - I

Case Name	**Sentence**	**PArt1 R2006**	**Part1 2010**	**Other parts**
Batch Control	"**batch control**: **Control activities and control functions** that ..."	Y	Y	-
Recipe Element	"**recipe element**: a **structural entity** that ..."	Y	N	-
	"A **recipe element** is a **representation** of ..."	N	N	P4
Control Module	"...**equipment entities** such as units, equipment modules, and **control modules**."	Y	N	-
	"...the lower level entities, such as equipment modules and control modules."	Y	Y	-
	"**control module**: The lowest level **grouping of equipment** ..."	Y	Y	-
Procedure	"...the **procedural element**, such as **procedure**,	N	N	P4
	"**procedure**: The **strategy**..."	Y	N	-
Control Recipe	"...include **data** such as **control recipes**,..."	N	N	P4
	"**control recipe**: A type of **recipe** which..."	Y	Y	-
Equipment Module	"...**equipment entities** such as **units, equipment modules** ..."	Y	N	-
	"**equipment module**: A **functional group of equipment**..."	Y	Y	-
	"...**lower level entities**, such as **equipment modules**..."	Y	Y	-

Table 4.7.: cont. Selected cases from the ISA88 Standard - II

Case Name	**Sentence**	**PArt1 R2006**	**Part1 2010**	**Other parts**
Unit Procedure	“...**elements of batch production** such as **campaigns, unit procedures**, ...”	Y	N	-
	“... **recipe or equipment procedural element** smaller than a complete batch, such as a **unit procedure**,...”	N	N	P4
	“...**procedural element**, such as **procedure, unit procedure**,...”	N	N	P4
	“**unit procedure**: A **strategy** for...”	Y	Y	-
Phase	“**Phase**: The smallest **element of procedural control** that...”	Y	N	-
	“...**recipe or equipment procedural element** smaller than a complete batch, such as a **unit procedure, operation, or phase**.”	N	N	P4
Master Recipe	“...**data** such as **control recipes, master recipes**,...”	N	N	P4
	“**master recipe**: A type of **recipe** ...”	Y	Y	-
	“...**sources** such as other types of **schedules, master recipes**, ...”	Y	Y	-
	“A **master recipe** is a **template recipe**...”	N	N	P4
General Recipe	“**general recipe**: A type of **recipe**...”	Y	Y	-
	“The **general recipe** is an **enterprise level recipe**...”	Y	N	-
	“A **general recipe** is a **container** of ...”	N	N	P3
	“A **general recipe** is a **corporate recipe**...”	N	N	P3
	“A **general recipe** is an **enterprise-wide recipe**...”	N	N	P3
	“A **general recipe** is a type of an **equipment-independent recipe**.”	N	N	P3

4.4.4.1. Pruning the “BatchControl” Concept

Figure 4.17 shows how the “BatchControl” concept concludes with two parents in the ontology constructed from the text. These relations are accepted because this is the real idea taken from the text of the standard.

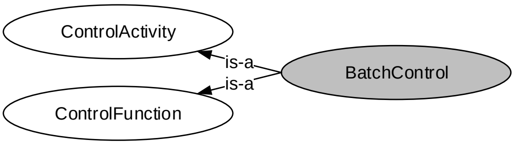
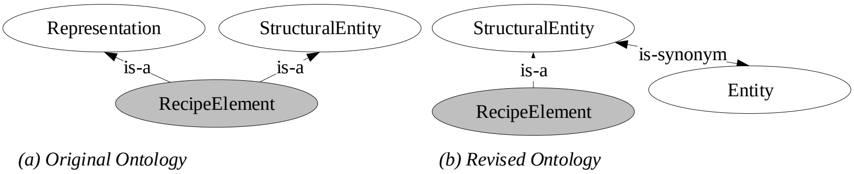


Figure 4.17.: “BatchControl” case

4.4.4.2. Pruning the “RecipeElement” concept

Figure 4.18a shows how the “RecipeElement” concept pertains to two super-concepts: “Representation” and “StructuralEntity”. The decision made here is to remove the “Representation” concept, which is detected from Part 4, and to keep the relation with “StructuralEntity” being used as a synonym of “Entity” concept in ISA88 (shown in Figure 4.18b).



(a) Original Ontology

(b) Revised Ontology

Figure 4.18.: “RecipeElement” case

4.4.4.3. Pruning the “ControlModule” Concept

Figure 4.19a shows how the “ControlModule” concept has three super-concepts: “LowerLevelEntity”, “GroupingOfEquipment”, and “EquipmentEntity”. Other super-concept “PieceOfEquipment” is shown in the graphic to have enough information for making a decision. Upon inspecting this segment of the ontology, it is clear that many concepts are being used as synonyms: “PieceOfEquipment” and “GroupingOfEquipment” give the impression of different ways of naming “EquipmentEntity”. Instead of just removing these concepts, they have been kept, but marked as synonyms in Figure 4.19b. Additionally, the “LowerLevelEntity” is removed since the explanations related to the degree of the entity is not clear.

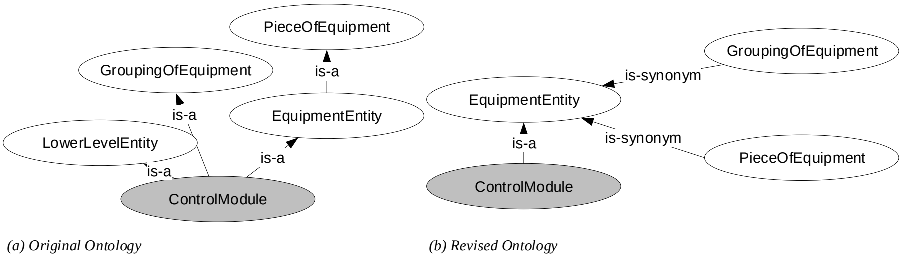


Figure 4.19.: “ControlModule” case

4.4.4.4. Pruning the “Procedure” Concept

Figure 4.20a shows the ontology segment for the “Procedure” concept and how this concept pertains to two different super-concepts: “ProceduralElement” and “Strategy”. The decision here was to remove the “Strategy” concept since control strategy is defined as strategy in the standard taxonomy, and the “ProceduralElement” concept is not at the same level of control strategy. Figure 4.20b shows the results after the pruning phase of the “Procedure” concept. Additionally, this decision is consistent with ISA882010, where the definition has been reformulated removing strategy in favor of the “ProceduralElement” concept.

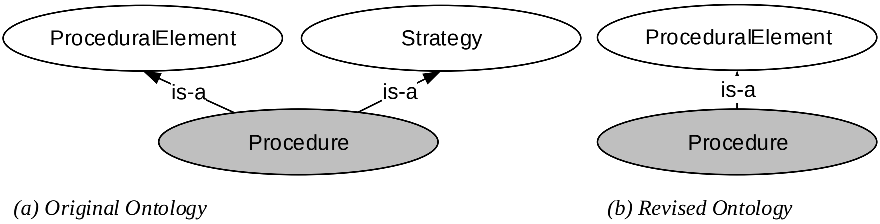
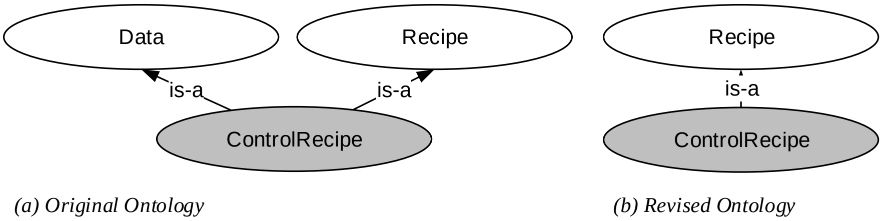


Figure 4.20.: “Procedure” case

4.4.4.5. Pruning the “ControlRecipe” Concept

Figure 4.21a shows how the “ControlRecipe” pertains to two super-concepts: “Data” and “Recipe”. The decision here was to remove “Data”, which comes from Part 4, considering the recipe model is modelled in the batch control systems. Figure 4.21b shows the results after the pruning phase.



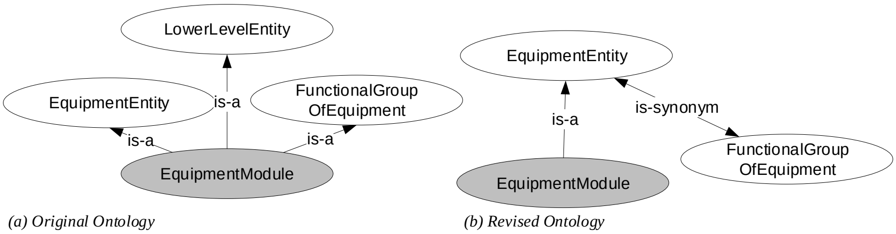
(a) Original Ontology

(b) Revised Ontology

Figure 4.21.: “ControlRecipe” case

4.4.4.6. Pruning the “EquipmentModule” Concept

Figure 4.22a shows the section of the ontology where the “EquipmentModule” concept is revealed to pertain to three super-concepts: “FunctionalGroupOfEquipment”, “LowerLevelEntity” and, “EquipmentEntity”. Decisions from the pruning procedure are that “EquipmentEntity” and “FunctionalGroupOfEquipment” are synonyms; the “LowerLevelEntity” concept is removed since it gives ambiguous information. Figure 4.22b shows the results after the pruning.



(a) Original Ontology

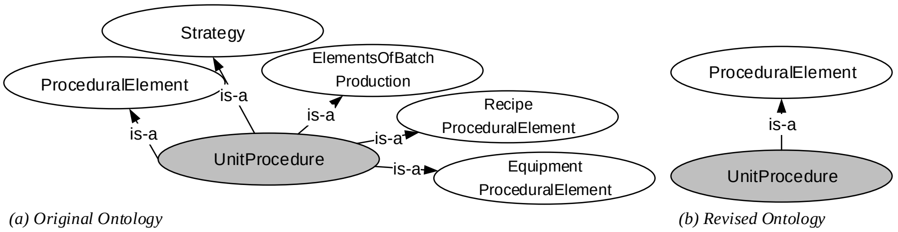
(b) Revised Ontology

Figure 4.22.: “EquipmentModule” case

4.4.4.7. Pruning the “UnitProcedure” Concept

Figure 4.5a shows that the “UnitProcedure” concept pertains to five super-concepts: “ProceduralElement”, “RecipeProceduralElement”, “ElementsOfBatchProduction”, “EquipmentProceduralElement”, and “Strategy”. In the pruning step, the relation between the “UnitProcedure” concept and the “Strategy” concept is removed by persisting the decision made for the “ControlStrategy” concept case, and the relation between the “UnitProcedure” concept and the “ElementsOfBatchProduction” concept is removed since all the concepts in the ISA88 Standard are considered as the “ElementsOfBatchProduction” concept. The

"ProceduralElement" is a super-concept of the "RecipeProceduralElement" and "EquipmentProceduralElement" concepts and the "UnitProcedure" concept is in the same level of these "RecipeProceduralElement" and "EquipmentProceduralElement" concepts. Finally, the relation between the "UnitProcedure" concept and the "ProceduralElement" concept is kept depending on the pattern from Part 4 and Procedural Control Model figure (Figure 7 in Part 1) in the standard. The revised ontology is depicted in Figure 4.22b. Furthermore, this is again shown to be consistent, since the "RecipeProceduralElement" and the "EquipmentProceduralElement" concepts are no longer included in ISA882010.



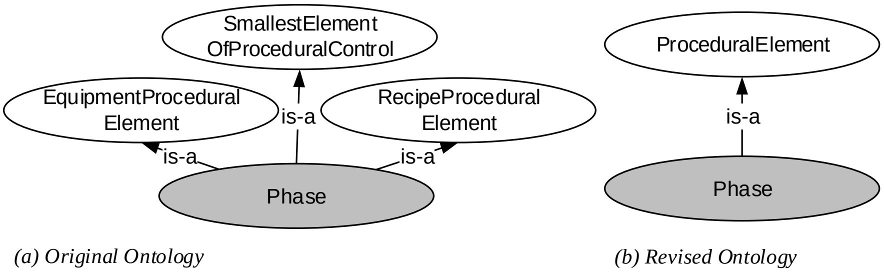
(a) Original Ontology

(b) Revised Ontology

Figure 4.23.: "UnitProcedure" case

4.4.4.8. Pruning the "Phase" Concept

Figure 4.23a shows that the "Phase" concept pertains to three super-concepts: "ProceduralControl", "EquipmentProceduralElement", and "RecipeProceduralElement". Since the "RecipeProceduralElement" concept and the "EquipmentProceduralElement" concept are not part of the procedural model in the standard, these relations are removed and a relation to the "ProceduralElement" concept is added. In addition, the "SmallestElementOfProceduralControl" concept is removed since it gives part-of relation between those concepts. As a result, the final decision on the concept is depicted in Figure 4.23b.



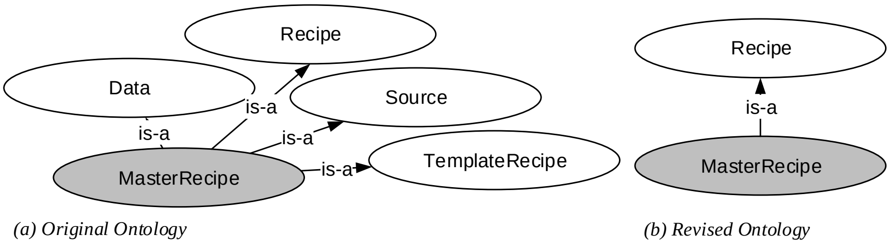
(a) Original Ontology

(b) Revised Ontology

Figure 4.24.: “Phase” case

4.4.4.9. Pruning the “MasterRecipe” Concept

Figure 4.25a shows how the “MasterRecipe” concept is simultaneously related to four super-concepts: “Data”, “Recipe”, “Source”, and “TemplateRecipe”. Relations coming from the Part 4 (Data and TemplateRecipe) are removed from the taxonomy because the Part 4 of the standard focuses on batch production database records. In addition, the relation to the “Source” concept is removed because of the consistency of recipe model. The “Recipe” concept is allowed. Final decision is depicted in Figure 4.25b.



(a) Original Ontology

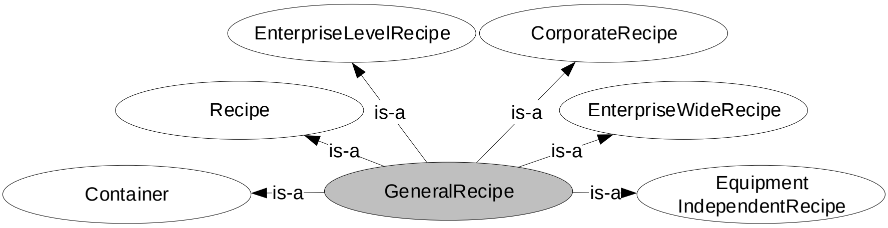
(b) Revised Ontology

Figure 4.25.: “MasterRecipe” case

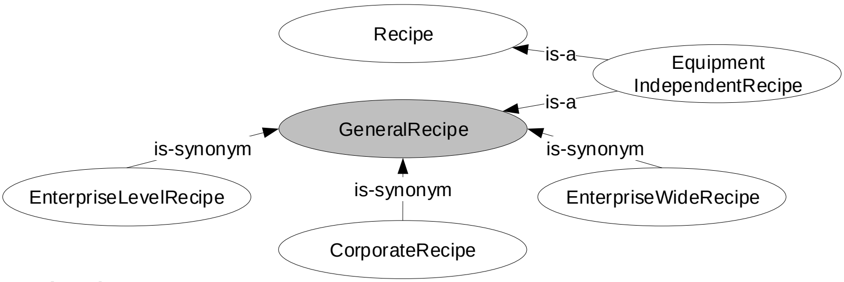
4.4.4.10. Pruning the “GeneralRecipe” Concept

Figure 4.26a shows that the “GeneralRecipe” concept pertains to six super-concepts: “Container”, “Recipe”, “EnterpriseLevelRecipe”, “CorporateRecipe”, “EnterpriseWideRecipe”, and “EquipmentIndependentRecipe”. Relations extracted from the Part 3 contain the “GeneralRecipe” segment and this generates misperception by giving additional descriptions with synonym concepts to the “GeneralRecipe”. Other concepts are created as different concepts such as the “EnterpriseWideRecipe”, the “EnterpriseLevelRecipe”, and the

"CorporateRecipe" concepts. Since all they represent the same concept, the is-a relations are changed to the is-synonym relation. The relation between "GeneralRecipe" and "Container" is removed since containment leads to part-of relation. Finally, the is-a relation between the "GeneralRecipe" concept has a "is-a" connection to the "Recipe" concept through the "EquipmentIndependentRecipe" as shown in Figure 4.26b.



(a) Original Ontology



(b) Revised Ontology

Figure 4.26.: "GeneralRecipe" case

4.4.5. Guidelines for Normative Document Writing

Cases in Section 4.4.3 show how a topological analysis of the ontological model allows detecting inconsistent lexical issues that may undermine the precision of the model to be standardized. As another result of this work, this section gives some guidelines to avoid these problems in a technical document defining a model. Hence, recommendations can be suggested to improve the development and writing of technical standards. Issues are explained with following logic: (i) problem, (ii) guideline, and (iii) example from ISA88.

Figures: The methodology presented relies on the automatic matching of textual patterns in the document. However, figures in a document cannot be processed in this way. The development of an automated process to infer semantic models from unprecedented, non-normative figures are out of the scope of this work, as well as currently unlikely. Thus, an expert has to inspect figures manually to extract information from them. In the case of ISA88, the lack of a legend in several figures made it difficult to extract information properly. This issue is more affecting when arrows of different kind occur in the same figure and no legend explains their different meaning (e.g. Figure 11 in (ISA 2000) is updated in the new version (ISA 2010) of standards).

Guideline: Add textual information to figures (legends, notes) when ambiguity exists.

Synonym: Although synonym is a powerful rhetorical resource of the language, having multiple names for the same concept goes against precision and clarity. Terms and particularly normative definitions should be strictly followed in standard. For example, the case in Section 4.4.4.6 shows how several names are used for referring to the same concept in ISA88: "Equipment", "GroupingOfEquipment", "EquipmentEntity", and "PieceOfEquipment". This diversity of concept names damages both human comprehension and computer automation.

Guideline: All references to the same concept should stick to the chosen name. The synonym usage should be avoided or explicitly declared especially for the normative concepts.

Polysemy: Polysemy is the usage of similar names for different concepts. Although it can be understood by humans, it is a challenging task for computers. For instance, the concept "Phase" is a procedural element in the procedural model but also the "RecipePhase" concept exists in the standard, which is part of the "RecipeProcedure" concept.

Guideline: Use different names for different concepts.

Adjectives: ISA88 Standard is using adjectives with great care. Qualifier adjectives are likely good indicators of an "is-a" relation between concepts such as "ProceduralControl is-a Control". The "Procedural" is used in 38 different concepts thus creating new concepts for instance, "ProceduralElement", "ProceduralControl", "ProceduralHierarchy", "ProceduralElementReference". More confusing are grade adjectives and comparative adjectives such as used in the "LowerLevelEquipment" concept, since they suggest the existence of possible superclasses ("LevelEquipment" or "Equipment"), and further relations.

Guideline: Avoid using comparative and unnecessary adjectives.

Adjectives with antonym: Another adjective usage is adjectives creating antonym as used in the “CommonResource” concept, which is also discussed by Fischer Bandy and Emerson (1996). These adjectives should be used with great care since they create the antonym of the concept and the ambiguity arises owing to searching the antonym of the concept.

Guideline: To avoid the ambiguity, adjective usage should limited to determining adjectives.

4.4.6. Discussion of Normative Documents Supported by Ontologies

Prior works have documented different ways of producing ontologies or conceptual models. They are based on either the exhaustive search of a great amount of noisy data. On the other hand, writing technical documents requires a great effort because the intrinsic ambiguity of NL, which tends to increase the length of these documents. This work is part of the continuing development of methodology for semi-automatic ontology construction, aimed at creating domain ontologies in a systematic way using technical standards as resource. It contributes an original use of semantic modelling for improving the development of automation standards and the subsequent development of automation software, according to the standard. Consequently, the main objective of the method is to systematically check the consistency of technical documents and provide suggestions for consistency improvement.

ISA88 Batch Control Standard is used as the case study in this methodology and the issues regarding to conceptual modelling that have appeared are now discussed and shared with the automation community. Quantitative assessment of the performance of the methodology presented is difficult due to the lack of convenient metrics for quantitative comparison. Furthermore, the most significant drawback is the lack of repetitive cases for sampling and comparing. However, the paper addressed a comparison between ISA 2006 and 2010 in regard of the issues automatically detected by the proposed approach and those issues that were detected and improved by a team of experts after a revision procedure.

As a result, improvement suggestions arisen from the analysis performed in this work are compared with the newer version of the standard when it is applicable. On the other hand, technical standards appear in many communities, engineering fields and study areas, and this methodology may have a significant impact in applications with large sets of models.

An evolution of different paradigms for standards development has been explained in Section 4.4.1. ISA88 is shown to be an example of the first paradigm, where a team of domain experts meets to agree a model for a domain model. A mid step of this evolution could be exemplified by B2MML (MESA, 2018), a mark-up language that is based in ISA88 and ISA95. In this case, a team of area experts has examined ISA88 and has built an object diagram of concepts and relations together with a language for information interchange based in XML or specifications. A first step in this evolution towards Paradigm 3 would be those automatic pattern matching processes that help area experts to build an ontology. This ontology would be more powerful than an object model since it allows performing inferences through axioms.

In third step proposed, an ontology would be constructed in parallel and in coordination with a NL text document. The result of this would be twofold: one document for humans to be read with a high degree of precision and clearness, and an information resource directly usable by computers. There would be no need for a two-step process as in Paradigm 2 where two teams meet, first to write a standard, and later to understand it and give an interpretation of it. In Paradigm 3, the text and its interpretation would be coordinated and simultaneous.

4.5. Concluding Remarks

This chapter has addressed the construction of domain ontologies that provide a basis regarding concepts and relations using ontological modelling techniques. The developed approach uses a semi-automatic construction of domain ontologies from normative documents methodology (SECOND Methodology) that combines automated and manual procedures resulting in a methodology capable of fast and accurate detection of concepts from normative documents (i.e. technical documents, standards).

Automated methods used to build ontologies tend to produce the extraction of irrelevant concepts and misleading relations. These issues are caused by ambiguous figures, undefined concepts, extended use of synonyms, and use of adjectives in the text. Thus, the idea has been to introduce steps to combine both automated and supervised processes. This work has described how an ontological analysis of the text of a standard is a powerful tool that can help to enhance the precision of the text and speed up the implementation of automation software complying the standard, as well as finding a practical application to a theoretical study. Therefore, the starting assumption has been the intelligent selection of texts will reduce the noise and allow a fast and straight identification of concepts and relations.

This chapter presented (I) the SECOND Methodology for building domain ontologies based on the processing of normative documents as well as (II) implementation of the methodology to a standard and (III) improvement of the sources.

(I) The starting point of the five-step SECOND Methodology is (i) the extraction of relation (e.g. the ISA88 Standard among many papers and books) and (ii) the determination of binary relations between these terms (i.e. origin and destination concept pairs). Afterwards, the methodology is completed by (iii) building branches by introducing extracted concepts, non-textual information (from tables, charts and figures), implicit information, (iv) pruning the ontology and (v) resolving inconsistencies and contradictions by solving complex cases. In the end, another cycle starts with a new relation from the step (i) for the enrichment of the domain ontology. The presented steps have many aspects to be improved using AI techniques and paradigms as well as with appropriate automated tools to support human supervision. The developed SECOND Methodology is expected to be a source of ontology construction methodologies since each step is open to improvements against new advances (e.g. image processing, decision-making).

(II) The ISA88 Standard case study is used to show the potential of the SECOND Methodology. The cycles of the methodology have been implements. The approach has captured comprehensive information from the standard comparing the previously developed ontologies from the same ISA88 Standard. Additionally, the resulting ontology of the case study has been used as a basis for further developments in this thesis. Provided concepts and relations are used in Chapter 5 to solve problems that appear at the multi-level functional hierarchical activities in decision-making; these concepts and relations based on the PS Domain have been connected to a DMM. The domain has supported the conceptual modelling of DMMs (see Chapter 7), concepts and relations. In addition, the integration of the CC Domain and PS Domain has been achieved in Chapter 7. Finally, application, sequences and algorithm for DMM construction procedures are presented in Chapter 8.

(III) The SECOND Methodology also allowed detecting inconsistencies arisen during the domain ontology construction and has improved the quality of normative documents by suggesting improvements. Due to this consistency checking, the formalisation of conceptual models and subsequent writing of normative documents have been simultaneously analysed; moreover, new guidelines have been proposed for their application to future normative document writing.

The contributions of this chapter can be listed as follows:

- The SECOND Methodology has been developed; the ISA88 Standard is used as a case study of the methodology and a more comprehensive modelling of PS information within a common and a shared domain has been built
- The consistency checking to improve normative document sources has been achieved using the SECOND Methodology.

As for future work, this research line includes the enrichment of the SECOND Methodology with further cycles so that the methodology can learn more about the normative document. Thus, a relation clustering method should be further developed to determine the relation clusters for next cycles. Clustering may require a broader grammatical investigation on verbs that may conclude an improved ontology.

Furthermore, the use of other technical documents connected to the domain and automatically search algorithms from the internet to get implicit knowledge may be explored. It may also be interesting to use other technical documents related to the domain and apply automatic search algorithms to get implicit knowledge from the Internet.

The contribution of this chapter can be summarized as follows:

Contribution 1[6]*:* The SECOND Methodology has been developed; the ISA88 Standard is used as a case study of the methodology and a more comprehensive modelling of PS information within common and shared domain has been built.

Contribution 2: The consistency checking to improve normative document sources has been achieved using the SECOND Methodology.

---

[6] Contribution numbers are given as in the Conclusions in Chapter 9

Chapter 5.

Integrated Management of Process Systems

Decision-making assigned to different hierarchical levels requires using complex mathematical models and high computational efforts for problem-solving, not including the need of extensive management of data and knowledge in Process Systems (PS). This chapter addresses decision-making in integrated systems by managing the data entered in a Decision-Making Model (DMM). It also proposes a comprehensive solution approach to the data and knowledge management framework, and guidelines for Computer-Aided Process Engineering (CAPE) tools for managing the corresponding cyber-infrastructures. The methodology presented in this chapter has been developed with the support of concepts from a domain ontology, which is one of the outcomes of the SECOND Methodology (explained in Chapter 4). Concepts were also taken from an early version of the Process Systems Management (PSM) ontology, and then relations for the knowledge management framework were added. This development of knowledge also started to shape the PS Domain. In this chapter, the PSM ontology is the connector between the introduced data, the DMMs developed to solve decision-making problems, and the necessary information to build the required problem instances. The methodology used in this chapter demonstrated its capability to exploit different decision-making processes in complex cases, which lead to new applications and/or extensions of these flexible and robust DMMs. Finally, the solution methodology presented in this chapter has become one of the bases of the Conceptual Constraint (CC) Domain that was explained and exploited in Part III and Part IV, for the use of previously built DMMs for solving new problem instances.

5.1. Introduction

Recent research has addressed integrated management of PS. However, it is generally accepted that academia and industry do not achieve a common ground for PS integration because mapping between (real) complex systems and case studies is hard to accomplish.

On the other hand, many works have recently started to establish a link between hierarchical levels using conceptual models (Vegetti and Henning, 2015; Vegetti et al., 2016; Muñoz et al., 2015) (see Section 2.4.2 for further explanations). An essential element in linking planning and scheduling activities is the ANSI standards: ISA88, ISA95 (ISA, 2010, 2000), which deals with the procedural, physical, and process models, and data exchange in PS. Similarly, supporting models, like BatchML in XML schema files, for these standards can be found (MESA, 2018).

An extensive study of the general problem of vertical and hierarchical integration is need and advanced tools and extensive DMMs have to be used to support the decisions in this integration procedure. Three main strategies have been suggested for solving the integration problem: (i) hierarchical, (ii) iterative, and (iii) full-space methods. All of them require computationally effective formulations, communication between the models and complementary strategies (Maravelias and Sung, 2009). Recently, Muñoz et al. (2015) have used a full-space method to solve the integration problem, and also used an ontological approach for communication between master and slave formulations supported by a Lagrangian decomposition approach. Concurrently, Fedorova et al. (2015) have used generic model templates constructed with an ontology, that provided several layers to create new models from already existing templates in the domain of process modelling. Besides these integration techniques and generic approaches, this work introduces a further step to bring a new solution to the integration of planning and scheduling levels.

This chapter addresses integration problems using ontologies to develop information flow channels, and proposes a general solution strategy for decision-making in integrated management systems. The goal of the methodology is to use mathematical programming models that solves different problems by manipulating the input data structure. The details of developed methodology are explained below by using separated modules.

5.2. Methodology

The proposed methodology systematically analyses the procedure by using four modules. These modules introduce different support routines for the problem solving as follows:

Module 1: Data Flow to collect data,

Module 2: XML file(s) to transform data,

Module 3: OWL file(s) to transfer data, and

Module 4: DMM(s) to solve the problem.

In addition, there is an ontology management tool for preparing the input structure of the DMMs in the proposed methodology.

An overview of these modules and their connections to create functionalities are shown in Figure 5.1. Afterwards, each module and the ontology management are explained.

5.2.1. Module 1: Data Flow

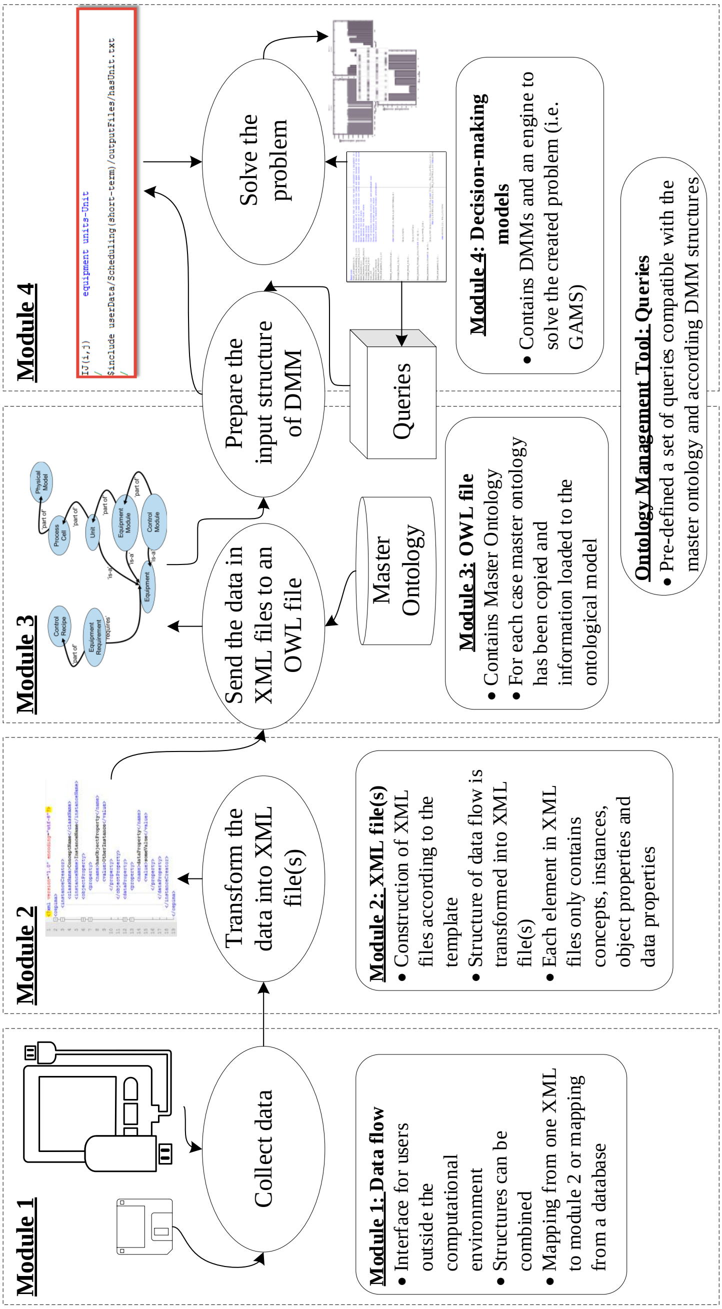
This module was built to maintain the data flow from/to different interfaces and/or data structures. Module 1 collects the data based on the ISA88 Standard and was developed to sustain the functionality of the methodology, considering its compatibility with other modules. The following is a list of some of the potential tools and methods used by this module for collecting data introduced as follows:

- Pre-modelled interfaces according to concepts and relations behind the main functionality,
- Structure creation without using interfaces,
- Mapping by following modules to previously structured files (e.g. .mat files, JSON files, XML files, databases),
- Drawing tools that can capture network structures and
- Introducing an environment that allows users to create their own excel files that can be read by the system.

5.2.2. Module 2: XML file(s)

Module 2 stores the XML file(s) used to maintain the communication between module 1 and module 3. Each node in an XML file holds concept names, instance names, object properties, and data properties to be loaded to the OWL file. This module is essential for the connection of separately structured data and needs to be developed in coordination with the master ontology (in module 3) and data flow structure.

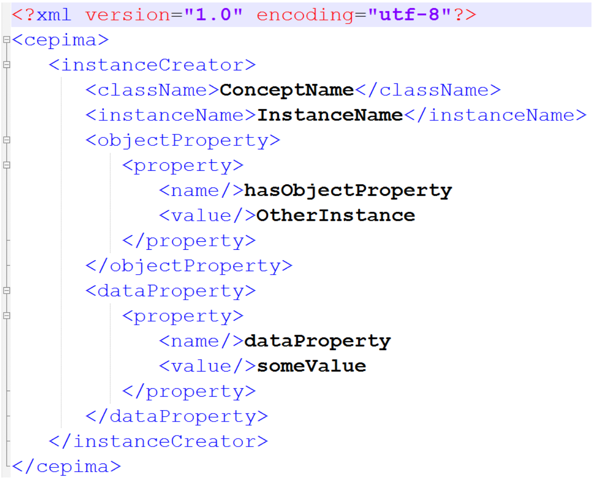
Figure 5.2 shows a template of an XML file. These XML files are loaded into OWL files according to the case study requirements. The concepts and relations (object properties) between concepts, including instances of concepts, and data property of instances can be loaded into OWL using this XML template. The crucial challenge is to be able to convert any data introduced through module 1 into the structure for uploading into the ontology by using this template.



```
<?xml version="1.0" encoding="utf-8"?>
<cepima>
  <instanceCreator>
    <className>ConceptName</className>
    <instanceName>InstanceName</instanceName>
    <objectProperty>
      <property>
        <name>hasObjectProperty</name>
        <value>OtherInstance</value>
      </property>
    </objectProperty>
    <dataProperty>
      <property>
        <name>dataProperty</name>
        <value>someValue</value>
      </property>
    </dataProperty>
  </instanceCreator>
</cepima>
```

```
IJ(i,j)        equipment units-Unit
/
$include userData/Scheduling(short-term)/outputFiles/hasUnit.txt
/
```

Figure 5.1.: Integrated management of PS methodology overview



```xml
<?xml version="1.0" encoding="utf-8"?>
<cepima>
   <instanceCreator>
      <className>ConceptName</className>
      <instanceName>InstanceName</instanceName>
      <objectProperty>
         <property>
            <name/>hasObjectProperty
            <value/>OtherInstance
         </property>
      </objectProperty>
      <dataProperty>
         <property>
            <name/>dataProperty
            <value/>someValue
         </property>
      </dataProperty>
   </instanceCreator>
</cepima>
```

Figure 5.2.: XML template of the integrated management of PS

5.2.3. Module 3: OWL file(s)

Module 3 has a master ontology that is the static structure in this methodology and in this case study the master ontology is an OWL file with concepts and object properties. The first step is to copy the master ontology as a case study of OWL files. Afterwards, these case study OWL files are populated with instances and their object property and data property connections according to data from XML files. XML files are loaded to the case study OWL files using a Java-based instance manager.

The master ontology remains in its present form as long as a DMM(s) in module 4 remains the same. The improvement of the master ontology occurs when a new DMM appears in module 4 (explained in Section 5.2.4), and the queries are built using the ontology management tool (explained in Section 5.2.5).

5.2.4. Module 4: Decision-Making Models

This module may contain DMMs with and already built in different range of functionalities (e.g. scheduling, planning, long-term planning, design). The goal is to store DMMs and model their input structure using an ontology management tool. Information from a case study (stored in OWL files) will then be automatically retrieved and sent to a DMM for a solution. In this work, DMMs are modelled using GAMS (GAMS, 2016) modelling software, even though the methodology is suitable for connecting different tools and software applications.

5.2.5. Ontology Management Tool

OntoCEP[1] has been used as the ontology management tool in this methodology because it enables searching the established connections between concepts in an ontology, saving queries from these connections, and retrieving information from a case study OWL among other functionalities. Plus answers to queries can be saved as input syntax of GAMS by this software application. The query construction is one of the essential tasks in this methodology since each DMM has particular input structures connected to PS. Consequently, the objective is to build the input structure of each DMM inserted into the system, and provide a set of queries to receive data from case studies. This task is accomplished with the following steps: (i) select a DMM, (ii) establish input structure of the DMM in master ontology and (iii) create a set of queries for information retrieval.

Apart from its primary task, OntoCEP can check inconsistencies in the loaded information in the case study OWL files, and detect missing information in the case of an empty query. The detection of missing information provides feedback to other modules to reduce faults that may appear during the data collection and processing.

5.3. Case Study: Area Manager

The methodology is implemented as a CAPE tool, a case study with different scenarios is used to show the adaptability of the application.

The case study is organised from an area manager point of view who is responsible for the planning and scheduling of two products in three process cells. The physical model of this multi-process cell is illustrated in Figure 5.3. The state-task network (STN) of the case study can be seen in Figure 5.4 (drawn with yEd graph editor), in which each scenario has been built using the same recipe.

5.3.1. Scenarios

There are three different scenarios that consider these process cells (together and separated)to satisfy the demands that are planned as follows:

Scenario 1: The first scenario includes the original case study from Kondili et al. (1993) identified as Process Cell 1 (PC1). Additionally, two more process cells are created by increasing and decreasing the reactors' maximum batch size by 10% in Process Cell 2 (PC2) and Process Cell 3 (PC3), respectively. Furthermore, the total demand

---

[1] For more information please contact the group CEPIMA (`CEPIMA.upc.edu`).

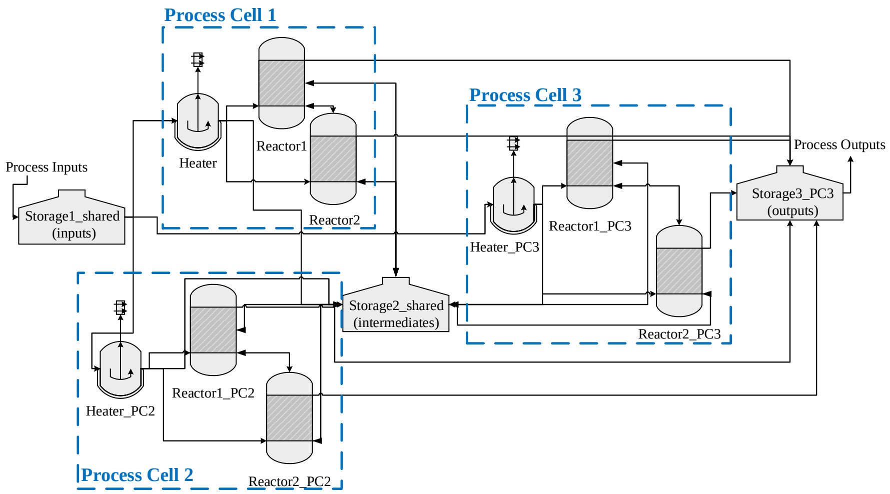


Figure 5.3.: Case study: Physical model of multi-process cell

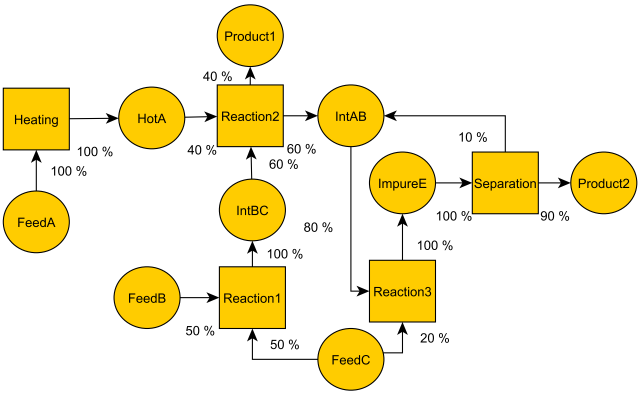


Figure 5.4.: Case study: STN

(1500 units of Product1 and 1200 units of Product2) is equally divided among all the cells, so that the optimisation problems are solved separately for each process cell.

Scenario 2: The second scenario has a holistic view of the problem where all process cells are allowed to share intermediates. The total demand is then assigned to the entire area (i.e. a collection of three process cells) instead of addressing each process cell separately. Furthermore, intermediate storage amount is tripled for this scenario.

Scenario 3: The third scenario consists of mapping the amount of demand and process outputs for each process cell according to process cell capacity change. The demand for each product from PC2 is increased by 10%, while the same amount has decreased in PC3. Like the previous scenario, intermediate storage amount has been tripled.

5.3.2. Application of the Methodology

The application of the methodology using modules is explained using three steps as follows:

(i) Start the data flow module, which is used for the introduction of case study data to the system,

(ii) Translate the case study data of unique OWL files in the case study and

(iii) Open the OWL files that are used to create and solve the DMM.

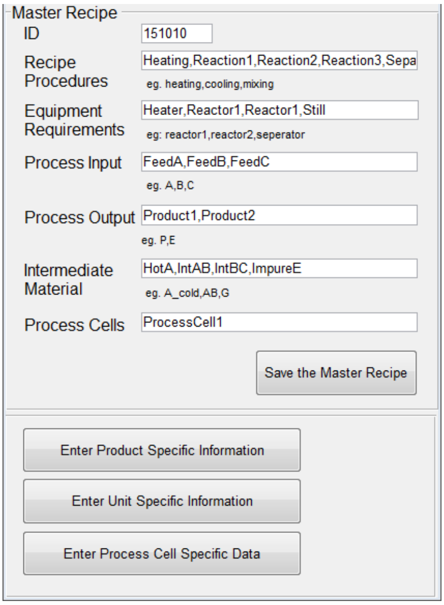
Modules are supported by the mark-up languages such as OWL (web ontology language) and XML (extensible mark-up language). Additionally, specific tools are used in the implementation of the methodology:

- A user interface is created using Matlab GUI (graphical user interface),
- Structures to build XML files are constructed using Matlab,
- Connections between XML and OWL are achieved with a JAVA-based ontology management tool and
- Mathematical formulations are implemented in GAMS.

5.3.2.1. Data Flow (Module 1)

For this case study, a Matlab-based GUI implementation and a graph editor to support the system's graphical representation were used. The GUI window used to introduce the recipe instance, which belongs to the case study, is shown in Figure 5.5.

Afterwards, the STN was drawn as in Figure 5.4 using the yEd graph editor (YWorks, 2018). This drawing has been saved as a .txt file for the next step (shown in Figure 5.4).



ID: 151010

Recipe Procedures: Heating,Reaction1,Reaction2,Reaction3,Sepa

Equipment Requirements: Heater,Reactor1,Reactor1,Still

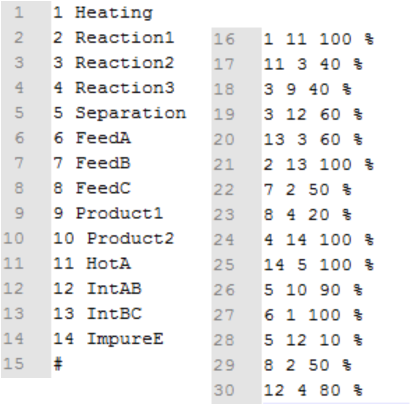
Process Input: FeedA,FeedB,FeedC

Process Output: Product1,Product2

Intermediate Material: HotA,IntAB,IntBC,ImpureE

Process Cells: ProcessCell1

Figure 5.5.: Implemented interface view



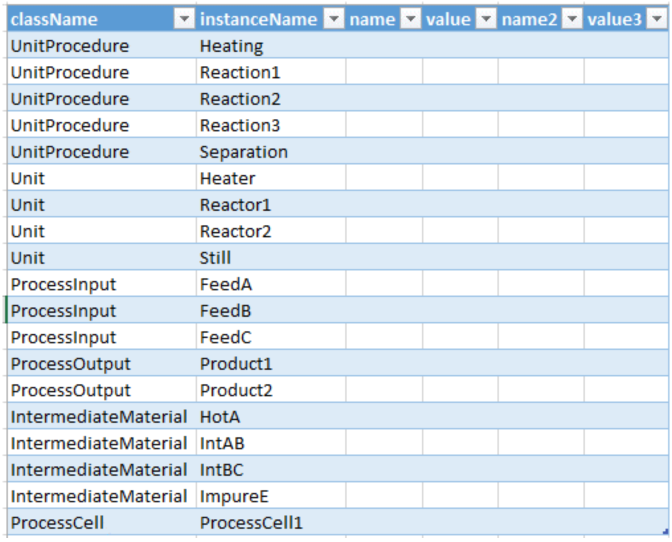
```
1 Heating
2 Reaction1
3 Reaction2
4 Reaction3
5 Separation
6 FeedA
7 FeedB
8 FeedC
9 Product1
10 Product2
11 HotA
12 IntAB
13 IntBC
14 ImpureE
#
1 11 100 %
11 3 40 %
3 9 40 %
3 12 60 %
13 3 60 %
2 13 100 %
7 2 50 %
8 4 20 %
4 14 100 %
14 5 100 %
5 10 90 %
6 1 100 %
5 12 10 %
8 2 50 %
12 4 80 %
```

Figure 5.6.: Text view of yEd file of the case study

5.3.2.2. Data to OWL (Module 2 - from Module 2 to Module 3)

This step receives information from the data flow module, saves it as a known structure following the XML template, and loads the information to the OWL files as required by the scenario under evaluation (Interface —> XML —> OWL). XML files are loaded to the case study OWL files used by the scenario that proviodes the flexibility for each scenario.

The data introduced from Figure 5.5 was transformed into an XML file, see Figure 5.7. The information saved in the XML file does not only include an instance of process input and output but it also includes the “classNames” tag that is introduced through the GUI application.



className	instanceName	name	value	name2	value3
UnitProcedure	Heating				
UnitProcedure	Reaction1				
UnitProcedure	Reaction2				
UnitProcedure	Reaction3				
UnitProcedure	Separation				
Unit	Heater				
Unit	Reactor1				
Unit	Reactor2				
Unit	Still				
ProcessInput	FeedA				
ProcessInput	FeedB				
ProcessInput	FeedC				
ProcessOutput	Product1				
ProcessOutput	Product2				
IntermediateMaterial	HotA				
IntermediateMaterial	IntAB				
IntermediateMaterial	IntBC				
IntermediateMaterial	ImpureE				
ProcessCell	ProcessCell1				

Figure 5.7.: Excel view of the saved XML file

The XML files related to the case study have been selected for each scenario and loaded to the case study OWL files. For instance, the “UnitProcedure” concept received 5 different unit procedures as shown in Figure 5.8 or the “ProcessCell” concept has 3 instances.

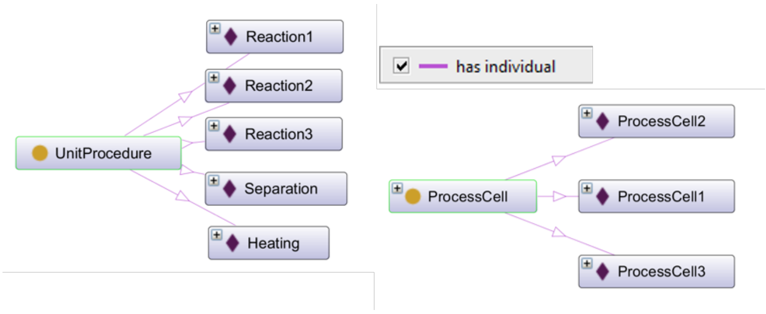


Figure 5.8.: "UnitProcedure” and “ProcessCell” concepts and their instances

5.3.2.3. From OWL to DMM (from Module 3 to Module 4)

This step presents a high complexity since it requires the definition of a DMM, as well as introducing the corresponding queries of the DMM input structure to the ontology management tool. Here, a short-term scheduling formulation (Kondili et al., 1993) is used to solve the case study scenarios. The input structure of the DMM was analysed and 26 different input structures (resulting queries) were detected. For instance, the "ProcessInput.set" file is included in the GAMS file that is supposed to receive a set of inputs from an OWL file (i.e. belong to the "ProcessInput" concept) as in Figure 5.8. Table 5.1 shows sample interaction between DMM input queries and data retrieved for each input.

Table 5.1.: Examples of normative rules and conclusions

DMM Query	**Retrieved input data**	**Meaning in the formulation**
Identification > ProcessInput >	FeedA, FeedB, FeedC, HotA, ImpureE, ...	Raw material set in planning
Entity > RecipeEntity > BuildingBlock > ProceduralElement > UnitProcedure >	Heating, Separation, Reaction1, Reaction2, Reaction3	Unit procedures in order to manufacture products
Previous line and Entity > RecipeEntity > BuildingBlock > ProceduralElement > UnitProcedure > [hasUnit :]	Heating.FeedA, Separation.ImpureE, Reaction.HotA, etc.	Inputs of unit procedures

Finally, the case study information has been entered in the model, and the DMM input structure presented in Table 5.1 has been used for its solution. In the end the DMM was solved using GAMS 23.8.2 and the Cplex solver.

5.3.3. Summary of the Implementation

All generated scenarios use the same recipe information (i.e. STN, procedural model), but each process cell (PC1, PC2, and PC3) has been built with separate information. When solving scenario 1 (depicted in Figure 5.9), XML files for each process cells are loaded to three different OWL files. The DMM is then solved three times for different problem instances. Scenario 2 and 3 include all the XML PC1, the XML PC2, and the XML PC3 and recipe for solving the problem, except that the demand information should be shared a constraint for the demand balance that is included for the solution of scenario 3. To ensure the information introduced for each scenario does not repeat the recipe information that is shared by each added process cell addition to easy combination and flexible usage. Scenario 2 and scenario 3 include all XML files (PC1, PC2, and PC3), as well as the recipe information to solve the problem. However, the scenario 3 includes process cell-specific demand information. As required by the scenario 3, a constraint for demand balance is also added to the DMM. Recipe information is shared by all process cells, enabling greater flexibility by avoiding redundant data.

5.3.4. Results

The results have been organized as computation results (see Section 5.3.4.1) and metrics of master ontology (see Section 5.3.4.2) for each scenario.

5.3.4.1. Solution of Scenarios

Table 5.2 summarizes the results for presentation to an area manager who is responsible for making decisions related to planning and scheduling. Each scenario is designed as a different decision-making problem and is proposed to demonstrate the methodology's capabilities for complex cases. For instance, if the problem requires assigning process cells to specific production orders, the manager can choose a process cell from scenario 1 by inspecting the optimal scheduling solutions in the make-span column from Table 5.2.

Scenario 2 contains the optimal scheduling data when the demand is not assigned to specific process cells, while scenario 3 considers that the demand is assigned to each process cell (a more restricted/constrained case). Scenario 2 has the highest profit since the problem is constructed monolithically and constraints on demand are removed when compared to scenario 3. Furthermore, computational effort is shown in Table 5.2. The lowest execution time is obtained when constraints on product demand are removed (i.e. scenario 2), while scenarios 1 and 3 require higher computational efforts, with scenario 3 the one having the highest execution time. The required computational effort is lower when the maximum batch sizes increase (scenario 1).

5.3.4.2. Ontology Metrics

Table 5.3 introduces a summary of ontology metrics for all scenarios[2].

For the first scenario, the problem has been solved three times with the same structured information that is shown in Table 5.3. In the table, the first part of this scenario (scenario

---

[2] AL: Attributive language with concept intersection, atomic negation, universal restrictions, E: Full existential qualification, (D): Use of data properties, data values or data types, and H: Role hierarchy - sub-properties

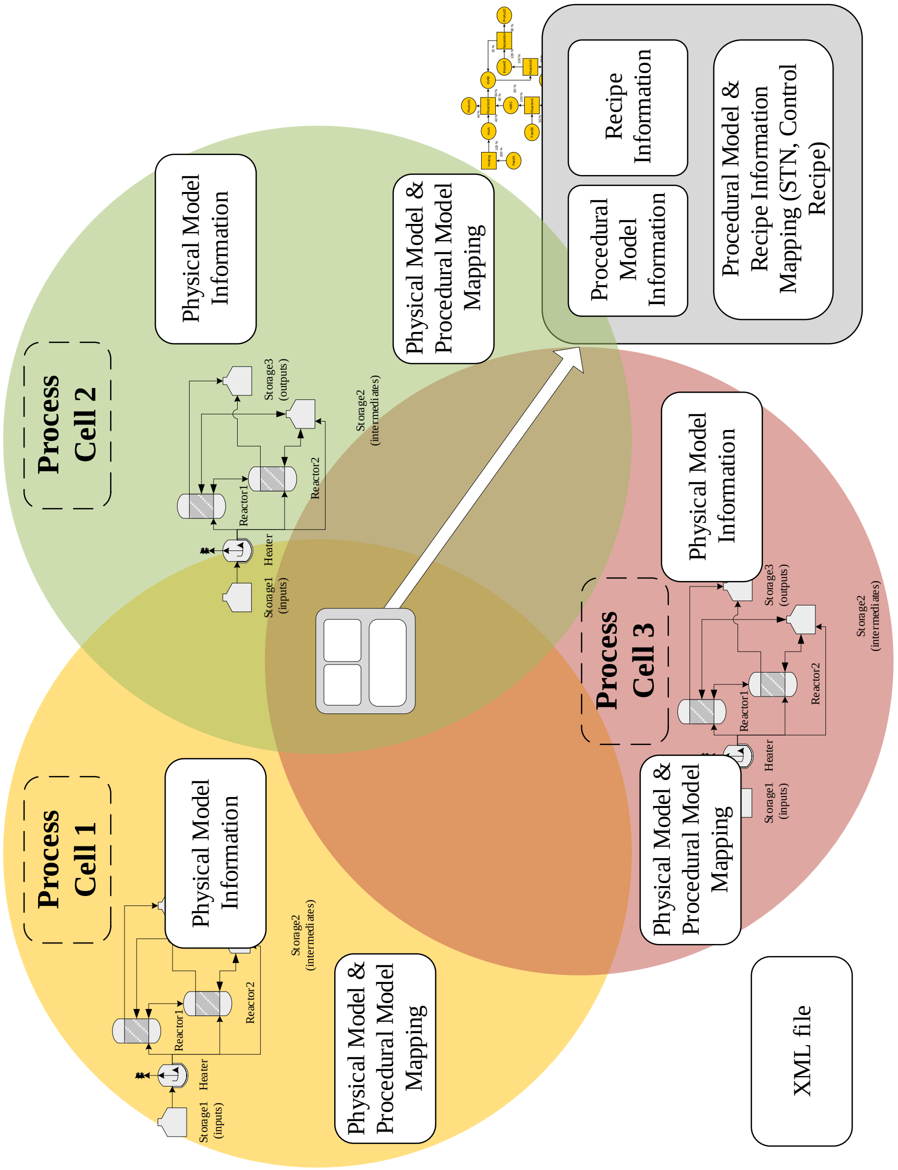


Figure 5.9.: Preparation of scenarios in practice

1a) has been included using the same information structure but with different data values for scenario 1b and 1c. The number of instances (class assertion), object property assertion, and data property assertion, increased in scenarios 2 and 3 by less than three times when compared to scenario 1a. This is due to the fact that all scenarios share the same

Table 5.2.: Result of scenarios

Scenario	Process Cells	Product	Demand	Make-span	Profit	CPUs
1a	PC1	1	500 units	40 hour	13432 units	0.593
		2	400 units	31 hour		
1b	PC2	1	500 units	40 hour	13535 units	0.453
	(10% more)	2	400 units	21 hour		
1c	PC3	1	500 units	40 hour	13393 units	0.889
	(10% less)	2	400 units	21 hour		
2	PC1, PC2,	1	3*500 units	40 hour	40705 units	0.125
	PC3	2	3*400 units	22 hour		
3	PC1	1	500 units	40 hour	40602 units	0.562
		2	400 units	25 hour		
	PC2	1	550 units	39 hour		
		2	440 units	33 hour		
	PC3	1	450 units	32 hour		
		2	360 units	30 hour		

Table 5.3.: Ontology metrics comparison from master ontology to each scenarios

Ontologies:	Master Ontology	Scenario 1a	Scenario 2	Scenario 3
Metrics				
Class count	501			
Objective property count	15			
Data property count	22			
Instance count	0	94	199	205
DL expressibility	ALE + (D)	ALEH + (D)	ALEH + (D)	ALEH + (D)
Individual axioms				
Class assertion	0	94	199	205
Object-property assertion	0	129	217	229
Data-property assertion	0	86	169	175

information. The difference between scenario 2 and 3 is a result of the separated demand assignment to 3 process cells and 2 products.

5.4. Concluding Remarks

This chapter proposes a methodology for integrated management of Process Systems (PS). It also includes a modular approach for data management and a flexible way of making decisions on the planning level have been entered. The chapter uses ontology and the Process Systems Management (PSM) ontology from the previous chapter. This ontology enables the link between semantic models and Decision-Making Models (DMMs), and is part of the PS Domain.

The data needed to solve the different optimisation problems in different production scenarios is entered in to a problem formulation through a single interface, and the PSM ontology then determines the problem instance to be solved. The methodology showed robustness and flexibility for developing more complex cases and is flexible enough to use different auxiliary tools (like sophisticated drawing tools for efficiently feeding data to the ontology). Some flexible solution strategies are applied to solve various problems that are developed by extending the base problem. Mainly, alternative physical models, holistic models, and information connections are investigated. The research of developments related to the concepts that appear in the PS Domain has been connected to DMMs, and a series of scenarios were solved in order to demonstrate the benefits of semantic modelling.

In summary, Chapter 5 explains an implementation of the PS Domain before the conceptualized constraints introduction (detailed in Part III). To this end, the chapter uses the ontology constructed in the previous chapters and links the semantic models to the DMMs without the elements of the conceptualised constraints. After presenting results obtained with the ontology modelling in PS, a new section has been developed for the conceptualisation of the DMMs. As an extra contribution, this chapter addresses the usage of the PS Domain in a software implementation to solve different problems that may appear in multi-level hierarchies.

The contribution of this chapter can be summarised as follows:

Contribution 3: For the outcome of the SECOND Methodology, the PS Domain has been used to model the PS information within a common and shared domain. Additionally, this model information has been used in the solution of multi-level functional hierarchies (see C5 in Section 3).

Part III.

Decision-Making Models and Conceptualization

Chapter 6.

Introduction to Constraint Conceptualization

The integration of decision-making process is typically assigned to bring different functional hierarchies (strategic, tactical, and operational) together in Process Systems (PS) with the aim of producing a better solution for a current issue. It requires the use of sophisticated mathematical models and great computational efforts, in addition to the need for extensive management of data and knowledge within the PS Domain. This part (Part III) investigates how Decision-Making Models (DMMs) are conceptualised and linked to the PS Domain. As a first step, this chapter introduces the conceptualisation of constraints, which is based on the generic concepts of constraint types and their connections to the PS Domain. Constraints from different DMMs are reviewed and, accordingly, general/fundamental concepts (like balances) are examined. This chapter introduces the idea of conceptualisation of constraints in a broader sense. This chapter shows how conceptualisation can be used when the structure and availability of information are changed, enabling multilevel implementations. Therefore, the primary analysis of material balances is introduced; the main pattern of the "MaterialBalance" concept is investigated.

6.1. Introduction

The Committee on Challenges for the Chemical Sciences in the 21^{st} Century (Hopf, 2004) indicates that the development of new and powerful computational methods, applicable from the atomic level to the chemical process and enterprise levels, is a crucial factor to enable multi-scale optimisation. This challenge would broaden the scope of one of the objectives attained by the Process Systems Engineering (PSE) approach, focused on decision-making systematisation through modelling and optimisation, to a new generalised paradigm. In this line, Harjunkoski et al. (2014) address the usage of standards to build models systematically and so create a master model to configure new problems without modifying the algorithmic core of mathematical models. Hooker (2012) uses meta-constraints from a pre-built library to assist model builders in a constraint-programming framework. However, although the practical implementations based on these approaches introduce significant improvements during the model building process, these meta-constraints are not conceptually connected to problems to be solved in the system. The complete model building for the integration problem is not investigated.

This thesis introduces the construction of domain ontologies in Chapter 4 - including essential concepts and relations of the PS Domain. Chapter 5 introduces the enrichment and usage of the PS Domain: a methodology that encompasses data collection, processing the data into a DMM and a DMM solution. During this complete process, files (XML, OWL), support tools (yEd graph editor, Matlab), an instance loader, and an ontology management tool are used to establish direct connections to DMMs (as illustrated in Figure 6.1).

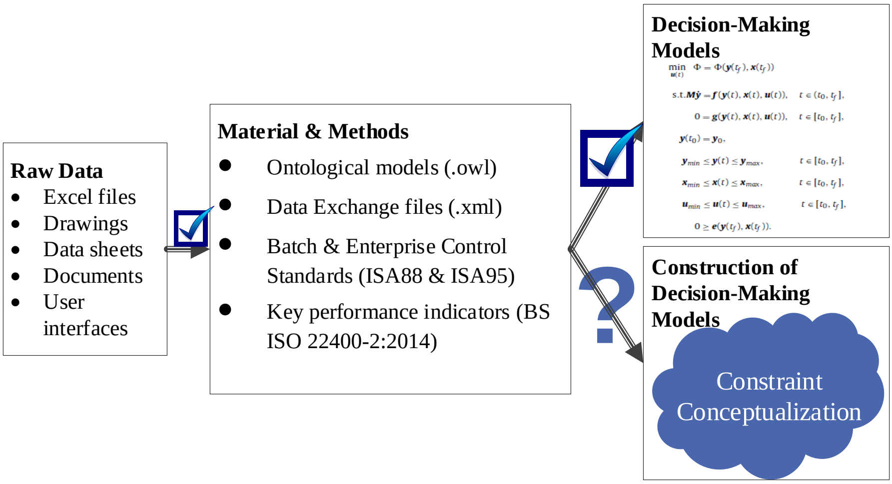


Figure 6.1.: Summary of the functional contributions

However, the procedure to construct DMMs automatically is not yet determined. It is necessary to develop a procedure for modelling elements for DMM conceptualisation and constraints/sets/variables/parameters as DMM elements. These modelling elements should address multi-level hierarchical systems and solve problems that may appear in supply chains. Additionally, the elements should produce a common language (i.e. shared, developed, and re-used) that produces more information than mathematical models.

The traditional modelling approach has the following steps: (i) a study of the process, (ii) a conceptual model of the process, (iii) a mathematical representation of the problem, and (iv) iterative model improvements (Makowski, 2005). However, how the mathematical representation of the problem is achieved and how this conceptual process model and the mathematical representation are connected is not clear. Furthermore, the mathematical representation of the problem and the connections to the conceptual model of the process are also unclear. Usually, the mathematical representation of the process is based on mathematical expressions related to fundamental laws such as balances, sizing restrictions, sequencing constraints, and allocation constraints. Other constraints, according to the details of the problem, are then added: for instance, in short-term scheduling models, time constraints can be used to describe shifts or maintenance requirements (Méndez et al., 2006). Alternatively, balances can be detailed according to features (i.e. energy balances, material balances). The mathematical representation of constraints is then built according to the model granularity (e.g. the used time representation), the given data, and other requirements. However, during this traditional modelling procedure, constraints and their connections depend on the expertise of the modeller and they are devised for a static problem. Therefore, the resulting DMMs remain static with the given data structure and cannot be reused at different levels (even within the same organisation) without additional effort.

To overcome these limitations, the aggregation of abstract information related to a common concept is proposed to conceptualise constraints at different hierarchical levels. This conceptualisation may be used to create upper-level relations and may be connected with different datasets available in the PS Domain.

6.2. Constraint Conceptualization

Ontological modelling is a conceptualisation to explicitly specify a domain (Gruber, 1993); established elements of ontological modelling have been used to conceptualise numerous applications (see Section 2.3).

Figure 6.2 shows primitive relations between concepts of conceptualised constraints connected to the PS Domain. The PS Domain has the "Identification" concept which consists of two more concepts ("ProcessInput"[1] and "ProcessOutput"). The material balance constraint is introduced as the "MaterialBalance" concept to the conceptualised constraint view. In this case, one of the elements that may appear in a material balance is the "CurrentlyAvailableMaterial" concept that has an instance, $S_{s,t}$. The goal is to then introduce connections from concepts to the following instances in the PS domain: "processInput1", "processInput2", and "processOutput1". All of these relations appear in the "MaterialBalance" concept and provide a pattern for the constraint.

---

[1] Concept names are written using CamelCase representation.

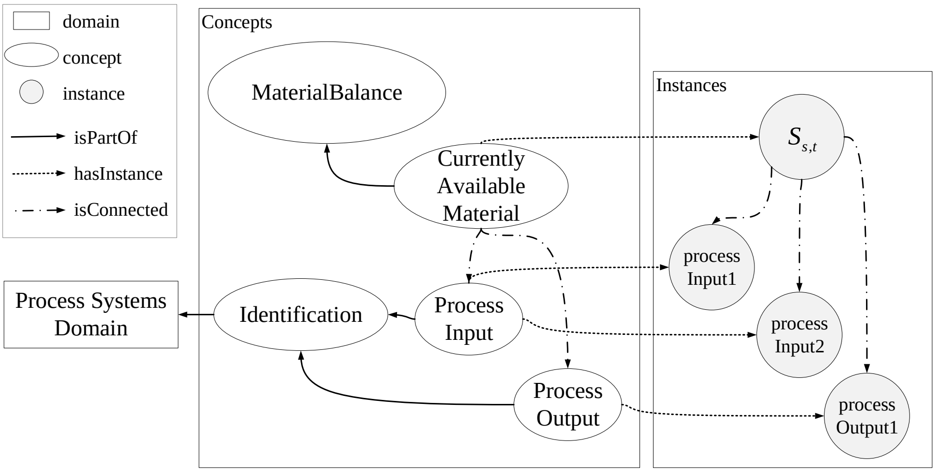


Figure 6.2.: Proposed conceptualization approach

Based on this idea, the proposed modelling approach exploits the constraint conceptualisation to formulate the problem at a higher (more generic) level, which is dynamically connected to the data in the PS Domain. Conceptualisation of constraints represents the principles of the technological system (e.g. material balances). To create the problem instance to be solved, the candidate concepts for constraint conceptualisation are then used to represent this norm (following the same example, the "CurrentlyAvailableMaterial" is connected to the "ProcessInput" and "ProcessOutput" concepts, which are part of the "Identification" concept in the PS Domain). These concepts may be gathered as the "Identification" concept since "ProcessInput" is defined as identification for materials, energy, or other resources required for a recipe (Note: Chapter 7 introduces an extension of the PS Domain, and current "identification" concepts are collected as the "StateModel" with additional concepts regarding hierarchical levels).

There are two aspects to be emphasised in this new way of modelling constraints. The first is related to how knowledge is managed to identify where system inputs are loaded in the ontological model (as in Chapter 5). The required systematic approach will typically imply the standardisation of the information; in this work, the ISA proposals (ISA88 and ISA95 Standards) have been applied, so the models include the recipe model, the procedural model, and the physical model (explained in Chapter 4). The second aspect is the constraint management associated with connections among conceptualised elements. The conceptualised elements of constraints construct the DMMs considering the PS Domain, and the proposed methodology implements the following steps: (i) ontological representation of the problem in PS Domain; (ii) selection from conceptualised constraints; (iii) model creation from conceptualised constraints and introduced data; and (iv) a solution of the model.

Furthermore, the claim is that conceptualised constraints are not only applicable to a particular hierarchical level (such as strategic vs tactical level). The same constraint may appear at different levels with different information and assumptions. Therefore, this approach uses the same generic concept that connects with the information to solve the problem at various levels. For instance, in the case of a material balance, depending on the available information, it can be constructed around a unit or a site; and the process inputs and outputs will change, accordingly.

6.3. Application: Material Balance Constraint Conceptualization

This section introduces the conceptualisation of material balance constraints using the proposed modelling approach (i.e. depicted in Figure 6.2). To explain how the conceptualisation is achieved, three material balance constraints are taken from DMMs (found in the literature) that aim to solve scheduling and long-term design problems. Therefore, the physical model is limited to units for the scheduling DMM and sites for the planning DMM.

The scheduling DMM seeks to find an optimal solution for short-term scheduling problems (Kondili et al., 1993). The material balance constraint from this DMM has been depicted in Figure 6.3; the figure contains detailed explanations related to the different elements and mathematical expressions found in the DMM.

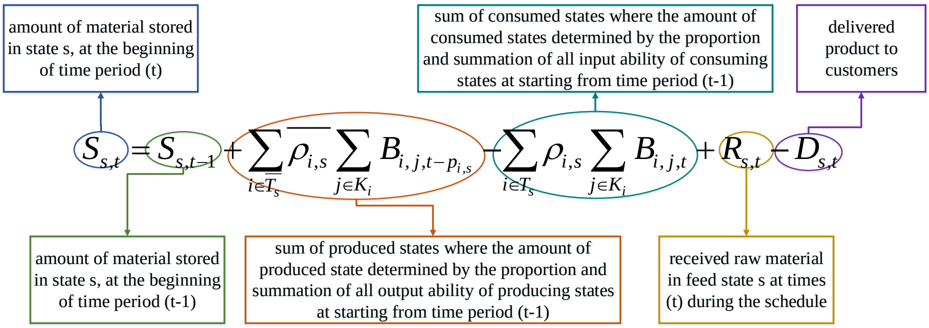


Figure 6.3.: Material balance from short-term scheduling DMM (Kondili et al., 1993)

The other two constraints are taken from a DMM; the DMM attempts to find an optimal design of chemical supply chains (Laínez et al., 2007). Two material balances are intro-

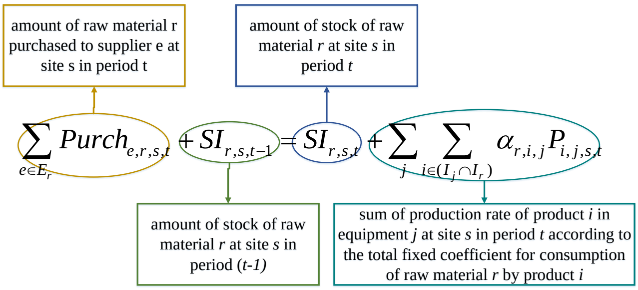


Figure 6.4.: Material balance built around raw materials from planning DMM (Laínez et al., 2007)

duced in this DMM: one is constructed around raw materials (depicted in Figure 6.4); while the second material balance constraint for this supply chain planning DMM is built around manufacturing sites (depicted in Figure 6.5). In the figures, each element of the constraints is semantically examined and described in the attached text-boxes according to the corresponding nomenclature[2].

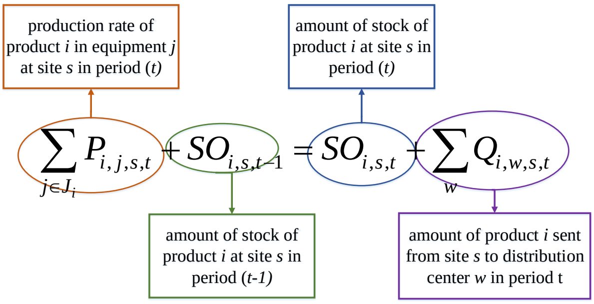


Figure 6.5.: Material balances built around manufacturing sites from planning DMM (Laínez et al., 2007)

In the planning DMM (Laínez et al., 2007), the material balance constraints for raw materials and products are separately created. The first observation for the material balances in Figures 6.4 and 6.5 is that this separation can be overcome using the recipe concept which is also known as a state-task network (STN) representation (Kondili et al., 1993). When the planning (Laínez et al., 2007) and the scheduling formulations (Kondili et al., 1993) are compared, the variable related to the production uses different physical elements: sites and units, respectively. To integrate the various levels, differentiation of the physical and procedural models is required (which is partially given in the ISA88 Standard and is applicable to other operation modes).

---

[2] Please see the source of the papers for a detailed description of the nomenclature used in these constraint and complete DMMs.

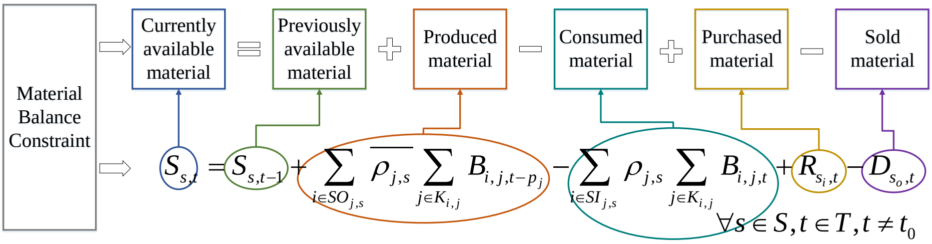


Figure 6.6.: “MaterialBalance” CC and resulted mathematical expression

Conceptually, combining the three examined constraints gives the general view of the elements in the “MaterialBalance” concept. This general view contains the elements of which constraints are composed; for instance, “CurrentlyAvaliableMaterial”, “ReceivedMaterial”, and “PurchasedMaterial”. Figure 6.6 summarises the “MaterialBalance” concept with a mathematical expression and the connections to these elements. The hierarchical conceptual relations of the constraint are presented in Table 6.1 taking into consideration the two DMMs previously studied. Relations in this chapter are restricted to the “Unit” and “Site” levels in the hierarchy. While the constraint in Figure 6.3 has instances of the “Unit” concept to be used as a set, constraints in Figures 6.4 and 6.4 have the “Site” concept connection.

Table 6.1.: Nomenclature for the “MaterialBalance” concept:

Sets	**Member Concepts**	**Subsets**	**Explanation**
s	Process, Site Process Segment Input, Process, Site Process Segment Output	$K_{i,j}$	Mapping between physical and procedural model
j	Unit Procedure, Site Procedure	$SO\,\&\,SI$	Recipe connection
i	Unit, Site	s_i	Process, Site Process Segment Input
t	Time Period	s_o	Process, Site Process Segment Output
Parameters	**Explanation**	**Variables**	**Explanation**
$\rho_{j,s}$	The proportion of input	$S_{s,t}$	Currently available material
$\overline{\rho}_{j,s}$	The proportion of output	$B_{i,j,t}$	Undertaken material for production
p_j	Processing time of the procedural model elements		

An additional example would be the “CurrentlyAvailableMaterial” concept, which is connected with an “Identification” concept to obtain the “ProcessInput” and the “ProcessOutput" for the identified level (Figure 6.2). For the planning model, the "Identification" concept, which describes materials required for recipes, includes the "SiteProcessSegmentInput" (raw materials) and "SiteProcessSegmentOutput" (products) concepts. The "CurrentlyAvailableMaterial" may then become a function of

$$CurrentlyAvailableMaterial(Identification, PhysicalModel, Time)$$

where the "Identification" refers to a set of materials that depend on the considered level. The "PhysicalModel" includes the set of "Unit" or "Site", while the "Time" concept adds the information related to the discretisation. These relations are the basis of the PS Domain and conceptualised concepts.

6.4. Concluding Remarks

This chapter has presented an introduction to constraint conceptualisation with the aim of comprehensively formulating and solving DMMs from different points of view in PS using a multi-level generic approach (or to connect a specific data set related to a problem and automatically building a DMM). As a motivating example, material balances have been selected to illustrate the conceptualisation.

The proposed conceptualisation approach is expected to apply to any system where a set of rules regulating the relations (connections) between the different sub-systems exist and the information inside these systems is modelled accordingly. In the case of multi-level hierarchies, these relations are precise since they are previously identified and even standardised. Therefore, the application of the proposed methodology and the identification of the conceptual equivalences become evident. As for other systems (e.g. interwoven systems and systems of systems), the relations may be more difficult to standardise for a generic case, although general concepts will also exist and may be exploited accordingly.

As a result, this methodology provides a basis for systematically creating of models, and even more importantly, ensuring the coherence of the results obtained by different models operating at different hierarchical levels in a multi-level system. However, it must be acknowledged that there will always be constraints that are not practical or feasible to generalise.

It must be noted that the proposed modelling approach requires more detailed study to be a model source of algorithms and functionalities. There is a need to develop concept types of constraint, in addition to special relations between these types of concepts. Furthermore, the PS Domain should be linked to this new domain. Conceptual modelling of constraints is developed in Chapter 7 in more detail with elements, principles and different types of constraints, and including linkage to the PS Domain.

The contribution of this chapter can be summarised as follows:

Contribution: This chapter has addressed the conceptual modelling of DMMs with an introduction to constraint conceptualisation, which introduces the very first steps of the CC Domain developments. A general introduction has been established in this chapter, as well as the conceptualisation of material balance constraints.

Chapter 7.

Conceptual Constraint Domain Building

Conceptualization is the basis of any systematic problem-solving procedure as well as the development of new techniques and paradigms for centuries. However, the formal conceptualisation, required for building decision-making systems automatically, has only been a subject of research in recent years. The goal of this chapter is to provide the basis for generating mathematical programming DMMs based on concepts. Fot this purpose, ontology-based modelling is applied to the Domain of CCs to generate an abstraction of currently available DMMs as explained in Chapter 6. Specifically, procedures have been developed for the identification and integration of models, as well as for recognising and classifying the features of the most common mathematically expressed constraints. The knowledge obtained from this identification can be transferred to other levels in the decision-making structure according to the information available from the PS Domain and generating other DMMs able to operate in other specific scenarios. This chapter presents this novel approach to conceptualisation.

7.1. Introduction

Conceptualization can be devised by ontologies which are defined as "explicit specifications of conceptualisation" (Gruber, 1993). Ontologies (i) support the construction of comprehensive models related to the scope of a particular activity (activity domain), (ii) rely on the use of a standard modelling language, and (iii) provide the formal representation of an activity domain by sharing the common understanding/knowledge sustained by a formal computer language (Bechhofer, 2007). These features support the integrated environment required by the experts to share and improve associated applications. Additionally, the four advantages of ontological modelling techniques for conceptualising domains are introduced as follows (Franz et al., 2007):

(i) Production: understandable models for both human and computers

(ii) Classification: the creation of classes, which represent commonalities,

(iii) Aggregation: the abstraction mechanisms and the consideration of objects as a whole, and

(iv) Generalisation: the abstraction of the commonalities of several classes.

Decision-making is one of the most significant tasks/activities/applications necessary for exploiting the knowledge of a particular domain. Thus, many domain applications use ontologies for the purpose of addressing a common understanding of different decision-making problems. This is the case of the ontologies associated with software support for process engineering (Morbach et al., 2009), batch process management (Muñoz et al., 2010), process synthesis (Kokossis et al., 2016), or more specific activities like pharmaceutical product engineering (Remolona et al., 2017). All these examples (and many others) have been broadly studied in recent years from the Process Systems Engineering (PSE[1]) point of view, based on the discovery and exploitation of concepts and models for the performance prediction of engineered systems (Grossmann and Westerberg, 2000).

Traditionally, the PSE approach aims to develop comprehensive Decision-Making Models (DMMs) to ensure that concepts and models are flexible enough to solve similar problems within the same content or input-data structure. This PSE approach produces DMMs starting from a simple first version, and picked up by model builders when they revise these DMMs. The first version of a DMM generally has fundamental formulations based on the first principles (e.g. balances, transfer laws) and current engineering practices (e.g. design rules, cost expressions). It is then further expanded or modified to achieve new targets by adding new elements, introducing additional simplifications, and/or adapting their scope to other cases/scenarios. The constraints are revised according to the problem specifications. For instance, in the short-term scheduling models and time constraints can be used to describe shifts or maintenance requirements while in planning problems. They can also be used to determine investment durations or delivery periods including the model granularity (e.g. the time representation), and other specified requirements. Therefore, DMMs are linked to the given data structure and model characteristics. Furthermore, these DMMs cannot be reused to solve various problems, even within the same organisation. On the contrary, when the characteristics of the problem change, the DMM should be updated with a new revising cycle by modifying the input-data elements (sets, parameters, variables) or adding/modifying constraints.

Many software applications (GAMS, 2016; AIMMS, 2018; IBM, 2018*a*; IMPL, 2016) are available to support the mathematical modelling through commercial or open source

---

[1] The complete list of acronyms is shown in Appendix F.

tools that enable experts to construct mathematical models for optimisation and simulation purposes from selected constraints derived from fundamental/physical/thermodynamic/kinetic equations. These modelling practices could be significantly improved by using ad hoc libraries for constraint selection (Yunes et al., 2010; Hooker, 2012), previously developed model templates (Fedorova et al., 2015), and/or linking models or templates to standard structures (Tolksdorf et al., 2017) like the CAPE-OPEN standards (CO-LaN, 2018). Still, these procedures rely on the users' expertise, given that the mathematical implementation of models requires knowledge and perspective from both the problem and modelling points of view. Additionally, the resulting models do not automatically connect the problem to be solved to the concepts behind DMMs, nor there is any method for checking the consistency of DMMs according to the knowledge of the authors.

Concurrently, the systematic generation of model equations has been the subject of process modelling studies in the specific domain of PSE. Stephanopoulos et al. (1990*a*,*b*) introduced a formal language for process model generation (MODEL.LA), which relies on a taxonomy for equation classification (i.e. reaction rates, balance equations). This modelling language has also been improved with an environment to support teaching activities in courses (Bieszczad, 2000). Another modelling language, BatchDesign-Kit, supports the design of batch processes by synthesising different process flowsheets and leading to systematic analysis to identify economic objectives (Linninger, 1995). TechTool allows constructing process models using a language based on the conceptualization of chemical and physical phenomena (Linninger et al., 2000). At present, commercial tools include procedures for generation of equations of process modelling (PSE, 2018).

However, to the authors' knowledge, all these works are limited to specific decision-making levels. Despite the extensive research conducted in recent years to formulate DMMs, the systematised model building and the improved tailor-made DMMs require the development of new methodologies to face the multi-scale application of the PSE approach. These methodologies are expected to be represented by systems using different degrees of formalities, which include natural language, ontologies, and mathematical process models (Klatt and Marquardt, 2009). Additionally, the steps that have to be followed to develop a DMM are not always clear. There is no systematic procedure for following and checking its consistency and completeness (Gani and E. Grossmann, 2007).

It is generally accepted that any decision-making procedure has to have of four phases (Simon, 1960): a set-up phase (intelligence phase), a design phase, a choice phase, and a decision implementation phase (see Figure 7.1). The set-up phase is related to the need to characterise the situation of the system, perceive the lack of a decision-making process, and set objectives, resulting in a problem statement. The design phase is related to the selection of the solution approach for identifying/designing alternatives and building DMMs.

The choice phase is related to the model(s) experimentation, which results in a decision that includes a selection of suitable solution alternative, sensitivity analysis, and adoption of a solution. The implementation phase is related to retrieving results and re-starting the cycle (if required).

This work studies the path from the problem statement to the resulting DMM in the PSE area within the framework of this decision-making cycle. The focus of the work is on linking the modelling of data along different domain perspectives. Therefore, a framework based on the integration of the Process Systems (PS) and the Operations Research (OR) perspectives (i.e. seeking a way to create the DMMs automatically) is proposed. The goal is to develop a systematic methodology to implement the model-based problem-solving approach employed in PSE studies that is able to address multi-level/multi-problem situations, and lead to specific DMMs according to the available information.

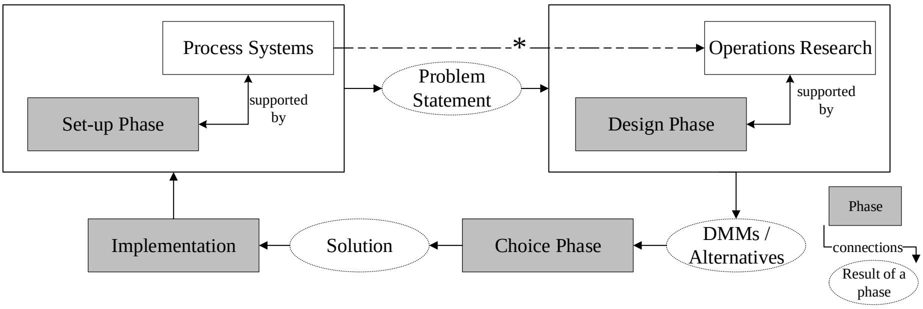


Figure 7.1.: Relations between the domains of the presented methodology and the decision-making procedure ('*': integrated domains)

The rest of the chapter is organised as follows: PS modelling and domain details are introduced in Section 7.2; this section is introduced as a continuation of Part II. A conceptual approach to DMMs and its corresponding domain are explained in Section 7.3. Section 7.4 introduces the conceptual domain of OR mathematical models. Afterwards, integration of the introduced domains, PS and OR, is explained in Section 7.5. Finally, illustrative case studies of the presented work are given in Section 7.6 and conclusions are presented in Section 7.7.

7.2. The Process Systems Domain

PS may be represented by a hierarchy that is a structure of sets and subsets so that every subset of a set is a lower rank than the set (Williams, 1994). This hierarchy can be any structure consisting in units and subunits where the subunits are lower rank than the units that are involved. In the systems view, every hierarchy has vertical interactions of subsystems that contain the entire system, in which the term (sub)-system is a transformation of the inputs into the outputs (Mesarovic et al., 1970). In the case of the appearance in PS, multi-level hierarchical systems represent this categorised system of PS, including interactions among different subsystems. Each system may occur as a part of the process model, physical model, or the procedural control model. From now on, (unless it is stated otherwise) multi-level hierarchies refer to the representation of PS in terms of structure.

Standards are one of the resources for obtaining shared views of the systems that they regulate. In the PSE area, one of the reference models for computer integrated manufacturing is the Purdue Enterprise Reference Model (Williams, 1989), which is the base for developing international batch and enterprise control standards (i.e. ISA88 (ISA, 2010) and ISA95 (ISA, 2000)). The models in the ISA88 Standard have also been implemented as XML Schema Definition (XSD) files (MESA, 2018) for enterprise control programs. On the other hand, the Purdue Enterprise Reference Model is the base of ISO 62264 Standard (ISO British Standards Institution, 2013), focusing on the third level activities (work-flow, recipe, detailed production scheduling, reliability assurance) of the multi-level functional hierarchy. Furthermore, NAMUR (Interessengemeinschaft Automatisierungstechnik der Prozessindustrie - User Association of Automation Technology in Process Industries) develops technical documents (recommendations) that apply to automation systems. The requirements for recipe-based operations and recipe concept are described in the NE-33 document (Namur, 2003).

On the other hand, the use of standards to structure problems has been suggested to support systematic model building without modifying the algorithmic core of the solution procedures (Harjunkoski et al., 2014). Furthermore, Moniz et al. (2014) suggest a knowledge-based approach based on the Purdue Enterprise Reference Model (Williams, 1989) and the ISA88 Standard (ISA, 2010)). The integration between scheduling and decision-making is established using the recipes and the corresponding mixed-integer linear programming (MILP) representation of DMMs.

The ISA88 Standard has already been used as a base for the conceptual modelling of some PS relations. The BAtch Process ONtology (BaPrOn) is built from the concepts of this batch control standard and used in order for monitoring and controlling the process scheduling in a pilot plant (Muñoz et al., 2010). The intention is not just communicating, but also supporting the integration of different software tools, as well as the exploitation of plant information (Muñoz et al., 2012). In addition, interoperability of planning and scheduling activities in batch processes have been modelled as an ontology (Vegetti and Henning, 2015).

The ISA88 standard has also been used to test a systematic approach for building domain ontologies. Dombayci et al. (2015). It includes the following two steps to this procedure: (i) an extraction procedure of concept pairs from a technical document (Farreres et al., 2014) and (ii) a systematic method of solving inconsistencies and contradictions arising from the first step. This semi-automatic procedure also produces a list of suggestions for improving technical documents by analysing the conceptual model that is semi-automatically constructed from the source (Dombayci, Farreres, Rodríguez, Espuña and Graells, 2017).

In an enterprise, daily operations are supported by transaction and analytical processing systems that provide Decision-Support Systems (DSSs). Transactional processing systems are based on managing the structured/stored data required to automate the decision-making activities. This structured data is used by the analytical processing systems that support the decisions with calculations or models. Consequently, the analytical processing systems are built to analyse the data stored in databases (Bog, 2014). It is also possible to introduce the analytical and transactional processing systems concerning the concept of tools (Grossmann et al., 2008). For instance, while the transactional tools represent database applications where significant amounts of data are stored and managed, analytical tools provide smarter solutions for DSSs taking into account the information available through transactional processing, which stores the data into the database application. In this framework, conceptual modelling can be used for representing and integrating transactional and analytical tools in a DSS. These analytical tools are designed to support the decision-making procedures through the integration of relevant data stored in databases or ontological models and analytical models (Laínez and Puigjaner, 2012). In addition, ontologies are well fitted to exploit the integration of analytical and transactional tools that can be exploited using ontologies (Muñoz, Capón-García, Laínez-Aguirre, Espuña and Puigjaner, 2017).

These aspects (e.g. standardized and ontological view of production systems, DSSs supported by both transactional and analytical tools) and many others, which apply to many kinds of chemical, physical, or biological systems (i.e. designed, built, and run to produce products, inspired by Marquardt et al. (2010)), are part of the so-called the PS Domain.

7.2.1. The Process Systems Domain Model

The PS Domain established in this work is related to manufacturing products in PS, and represents multi-level hierarchies using concepts. The management of the PS Domain requires the use of many concepts associated with the process, physical and procedural control models that can be found in the ISA88 Standard (ISA, 2010), and the appropriate extensions of the models developed through the ISA95 Standard (ISA, 2000). These standards can provide support for building the multi-level hierarchical information structures to solve problems that appear in PS. Consequently, the ISA88/95 Standards are chosen to be built as the core of the PS Domain since both standards can be comprehensively integrated to explain enterprise control systems.

7.2.2. The Process Systems Domain: Representation

The PS Domain Model is formalized with the support of an ontology (O) consisting of a set of concepts (C) and a set of instances of these different concepts (I) which are connected with a set of object properties (P). Object properties describe features of concepts, instances, or the relations between them. Additionally, there is a set of data properties (D) that describes the data associated to concepts or instances; and a set of axioms (A) which is defined over $C \cup I \cup P \cup D$ and represent rules and restrictions of the model (Franz et al., 2007).

The ontology can be generally represented following Expression 7.1 (the first time introduced in Expression (2.8)):

$$O = \{C, I, P, D, A\} \tag{7.1}$$

Generic concepts have been introduced to provide a general structure of the PS Domain e.g. “ProceduralControlModel”, “PhysicalModel”). These generic concepts are connected to specific concepts so that the concepts, which appear in the PS Domain, may be identified in one of the levels at the multi-level hierarchies. The identification of each specific concept also includes its generic concepts. For instance, *“UnitProcedure ‘partOf’*[2] *ProceduralControlModel”* while *“Unit ‘partOf’ PhysicalModel”* depends on the decision level (Figure 7.2). When these two specific concepts (i.e. “UnitProcedure”, “Unit”) are identified, they can be generalized through their generic concepts (i.e. “ ProceduralControlModel”, “ PhysicalModel”, respectively). Moreover, these specific concepts contain different relations that may be considered as vertical and horizontal relations. The vertical relations support the identification of the same level; for example, the “hasPhysicalCapability” relation between the “UnitProcedure” concept and the “Unit” concept provides the level connection between two concepts. The horizontal relations support the movements from one relation to another; for instance, the “Unit” concept is connected to the “ProcessCell” concept with the “partOf” relation as in the generic concept relation connection.

---

[2] Here, the “partOf” is a relation that connects two different concepts. It is also names as the object properties in the ontology language, owl.

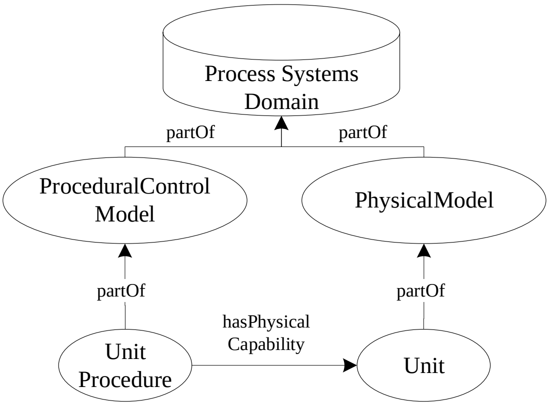


Figure 7.2.: General structure of the PS Domain

There are many ways to exploit the information and relations introduced in an ontology; for instance, reasoners (i.e. using the capabilities of the reasoners to retrieve implicit relations), semantic reasoning (i.e. using queries to retrieve knowledge), knowledge representation (i.e. sharing knowledge of a domain), and exchange files (i.e. transferring structured information through different channels).

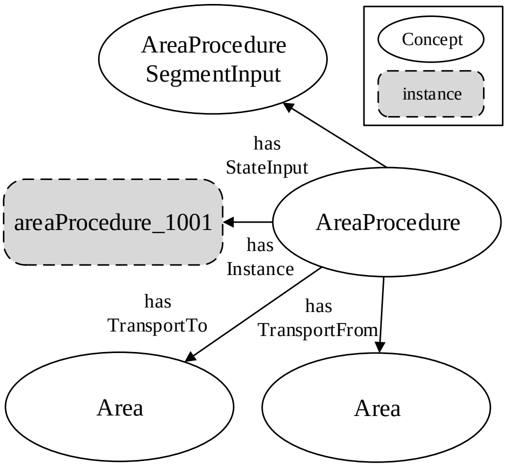


Figure 7.3.: Ontology representation example of the PS Domain (adapted from Dombayci and Espuña (2017))

This chapter focuses on using the semantic reasoning queries as a way to retrieve the information that belongs to a specific domain, in order to identify the data from the problem and feed this information to the DMMs. This functionality should be achieved in a robust way and should track the consistency of the information. In order to explain this query usage, an example from the PS Domain is presented in Figure 7.3, which shows the possible connections between the "AreaProcedure" concept and other concepts ("Area", "AreaProcedureSegmentInput", "AreaProcedureSegmentOutput") with some object properties/relations (e.g. "hasArea", "hasStateInput"). Moreover, the figure shows an instance of the "AreaProcedure" concept (areaProceure_1001), which is directly connected to the problem instance (Figure 7.4).

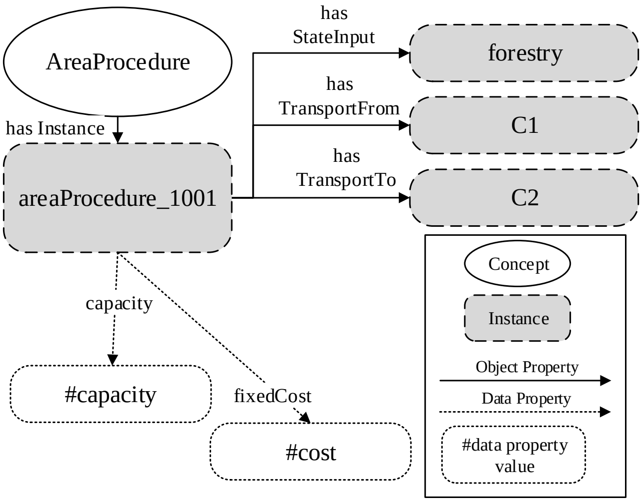


Figure 7.4.: Connections of a specific instance in the PS Domain Ontology (adapted from Dombayci and España (2017))

Two types of semantic queries me be devised in order to efficiently exploit the structure of the model associated with the ontology view of the PS Domain:

Type 1: Get instances with or without connected concepts:

- *hasInstance AreaProcedure* → gets all instances asserted to the "AreaProcedure" concept.
- *hasInstance (AreaProcedure hasStateInput StateInput)* → gets all instances asserted to the "AreaProcedure" concept which are connected to instances associated to the "StateInput" concept.

Type 2: Get values of instances with or without connected instances

- *AreaProcedure capacity* → gets all instances to the "AreaProcedure" concept and their "capacity" data property values.

- *((AreaProcedure hasStateInput StateInput) and (AreaProcedure hasTransportationFrom Area)) capacity* → get connected instances and their data property values.

These types of semantic queries have essential roles in the domain models, as well as for the integration of different levels at the multi-level hierarchies (explained in Section 7.5). The structure of the queries lets the modelled knowledge be transmitted without any effect on the concept change, and this means that the structure of the queries allows the integration to be introduced consistently once the linking rules are established.

7.2.3. The Process Systems Domain: Building

In summary, two steps were followed to construct the ontology representation of the PS Domain:

(i) Concepts and relations are retrieved from the ISA88 Standard to create a first conceptual model of the PS Domain. The resulting domain taxonomy includes around 500 concepts that are built through an ontology construction methodology (previously reported in Dombayci et al. (2018)).

(ii) Next, the resulting model is extended with the support of the ISA95 Standard as follows:

  - Identification of inputs, outputs, and intermediates corresponding to each level are introduced as the "StateModel[3]" (Figure 7.5). In the ISA88 Standard, the process input and output concepts are defined as raw materials and the resulting output of an execution, respectively, without an explicit reference to process intermediates between procedures. For this reason, a new set of concepts specifying the input-intermediate-output for each hierarchical level, is added to the model. The new "StateModel" concept includes the "StateModelInput", the "StateModelIntermediate", and the "StateModelOutput" concepts. While a specific level is used, the "StateModel" concept changes to the "ProcessStateModel", which contains the "ProcessInput", "ProcessIntermediate", and "ProcessOutput" concepts. These distinctions of concepts introduce the flexibility of using the "StateModel" as separated concepts and this usage introduces the practicability of collecting more than one concept or selecting one of them.

---

[3] In this work, concepts are written using CamelCase representation and instances start with the lower case. Also, object and data properties ("partOf", "hasComponentElement") use the CamelCase representation starting with lower cases. In the figures, the concept names are using spaces for each name to be read and understood.

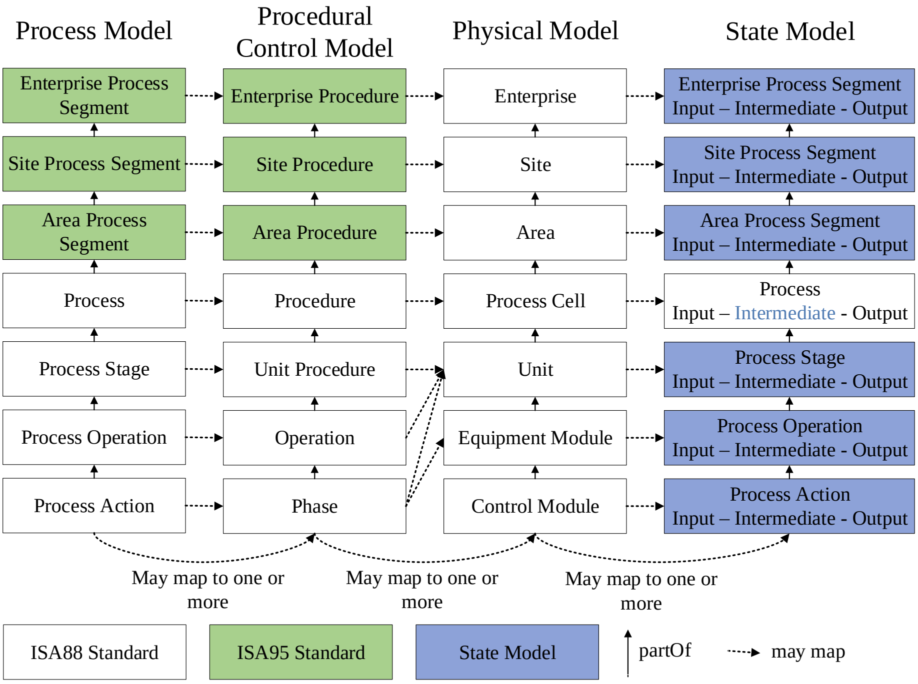


Figure 7.5.: Combination of models in ISA88 and the ISA95 Standards and the "StateModel" concept addition to the models

- The greenfilled-in part in Figure 7.5, which is not detailed in the ISA95 Standard, is also introduced. In the ISA95 Standard, there are only process segments, "which are the collection of capabilities needed for a segment of production, independently of any particular product". There are also product segments, which comprise "the shared information between a plan-of-resources and a production-rule for a specific product. It is a logical grouping of personnel resources, equipment resources, and material specifications required to carry out the production step" (Scholten, 2007*b*).

These steps can be further developed using the Plan, Do, Check, Act (PDCA) quality control cycle to support the ontology building (interested readers are referred to Muñoz et al. (2011) for PDCA application on ontology building).

7.3. Conceptualization of Decision-Making Models and Structures: The Conceptual Constraints Domain

In recent years, progressive developments of methods and tools have been based on computational capacity advancements and the cost-cutting required for solving huge problems. These developments supported by the improvements have generated considerable research interest on the development and application of generic DMMs, given that it is computationally and costly possible to model and solve them. Working in parallel, the research focus is moving from tailor-made DMMs to applying generic ones, since these generic DMMs contribute to reducing the time and effort required to build models. In this regard, generic mathematical models are expected to be able to address any problem in any domain if a mathematical representation of the system is available. That being said, the issue is still how these models should be constructed.

Traditionally, generating a mathematical representation of a problem is based on four construction steps (Makowski, 2005):

(i) problem analysis,

(ii) abstract modelling of the problem,

(iii) formal representation of the problem (usually using a mathematical-based language), and

(iv) iterative representation improvements.

In the case of mathematical programming, this representation of the problem usually consists in the following elements (Williams, 1978; Cagan et al., 1997):

- *Objective:* A mathematical expression to be minimised or maximised during the decision-making procedure
- *Variables:* Adjustable elements of models or so-called unknowns (including decision elements)
- *Parameters:* Coefficients of the model (scalars or matrices that do not change during the calculations)
- *Constraints:* Relations between parameters and variables, which have to be considered to ensure feasibility of the proposed decision
- *Sets/subsets:* Collection of objects or items gathered together indicating the size or complexity of the mathematical representation to be solved

Iterative improvements on a DMM can be achieved with these mathematical programming elements.

However, no systematic way has been developed to connect these mathematical programming elements with the conceptual understanding of problems for building the required DMMs. One way to connect these two views is to present the DMM with the conceptual understanding that produces the abstract view of the constraints, elements of the constraints, and the connections of these constraint elements by, (i) taking into account the associated mathematical programming elements and, (ii) connecting all of these conceptual elements to problem structures. From this comprehension, it is possible to connect DMMs with the conceptual understanding of the problem and also possible to support DMM building procedures.

The conceptual understanding of a DMM may be established from the constraint level. For instance, the flow variables from a balance constraint of the strategic planning mathematical model for every product (p), site (s) and time-period (t) are illustrated in Figure 7.6. The grey-filled ellipses represent the general conceptual terms in the figure, while the white filled rectangles introduce the matching pieces of the constraint. The "ProducedMaterial" term is expressed by the production amount connected to the product, site, and time-period, while the "ConsumedMaterial" term requires identifying the bill of materials (BOM) and the production amount.

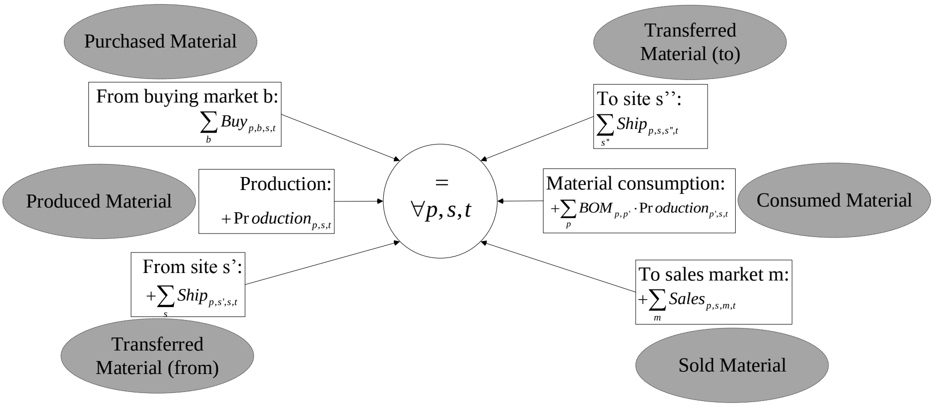
A general modelling approach is developed from the starting point expressed using these flow variables to overcome the scalability and conceptualization limitations needed to share, re-use, and build a domain (i.e. the Conceptual Constraint (CC) Domain). Again, the development of the CC Domain is supported by ontological modelling techniques, and the domain conceptualizes DMMs, which take into account constraints.

7.3.1. The Conceptual Constraint Domain: Model

The CC Domain proposed in this work is related to the OR scope and represents DMMs using concepts at three different levels: the modelling level (i.e. balances, calculations, amounts), the instance level (i.e. mathematical expressions, values), and the domain level, which includes types of constraints as well as their relations.

General features of the CC Domain elements are listed below:

(i) The Conceptual Constraints (CCs) express the type of constraints and represent the taxonomy of constraints, built through an "is-a" relation such as "BalanceCC", "ResourceAllocationCC", "TimingCC", "SizingCC", "SequencingCC", and "EconomicalCC" concepts. Once the most general types of constraints are determined as CCs,



$$\sum_b Buy_{p,b,s,t} + Production_{p,s,t} + \sum_s Ship_{p,s',s,t} = \sum_{s''} Ship_{p,s,s'',t} + \sum_p BOM_{p,p'} \cdot Production_{p',s,t} + \sum_m Sales_{p,s,m,t}, \forall p,s,t$$

Figure 7.6.: A flow balance constraint (adapted from Fleischmann and Oberstein (2008))

the CC taxonomy is deepened with sub-concepts (sub-classes). For instance, the "BalanceCC" has "MaterialBalanceCC" and "EnergyBalanceCC", which share the fundamental balancing idea as a common element.

(ii) The Conceptual Components (CComps) appear as the main terms in CCs (e.g. "Stored Amount" that is a component of a "MaterialBalanceCC"). The "hasComponent" relation connects the CCs and the CComps to express the patterns of each CC. Each of the CComps may be connected to more than one CC as a constraint element (parameter, variable), and may appear in more than one constraint. The elements in DMMs and their connections are straightforwardly represented by the CComps (e.g. the "ProducedMaterial" CComp may represent a variable). On the other hand, a CComp may represent an expression that is constructed from a variable and a parameter. For example, the "ProducedMaterial" may be a proportion (parameter) of the input material (variable).

(iii) The Conceptual Component Elements (CCompEls) are the specific concept types of the CC Domain, linked to the CComps that appear in another counterpart domain (in this work, the PS Domain detailed in Section 7.2.1) to carry out the decision-making process. For instance, the "StoredAmount" CComp may change from energy to material, depending on the CCompEls that are linked to it (the "StoredMaterial" or "StoredEnergy" concepts). Additionally, the connections between CCompEls are crucial for further details of the CC Domain as introduced in the PS Chapter.

7.3.2. The Conceptual Constraint Domain: Representation

The CC Domain takes into account the classification of constraints into CCs, the aggregation of the abstract CComps, and the generalisation of constraints for decision-making procedures using CCompEls. As in the case of the PS Domain, in order to formally represent the concepts and connections of the CC Domain, an ontology representation can be used (see Expression (7.1)). In the CC Domain, all of the developed CCs, CComps and CCompEls belong to the set of concepts (C). Instances of these constraints (I) are the real constraints. Object properties (P) are established between these concepts and specific instances of constraints and expressions. Data properties (D) play a crucial role in the integration of the CC Domain and the counterpart domain as explained in Section 7.5.

However, there are special characteristics associated to the CC Domain. The relational structure of the CC Domain illustrated in Figure 7.7 can be represented through the following expression:

$$\begin{aligned} CC[(CComp;\ Relation_{m_3};\ CCompEl_{m_4})_{m_2};\\ (\ ;\ Relation_{m_6};\ CCompEl_{m_7})_{m_5}]_{m_1} \end{aligned} \tag{7.2}$$

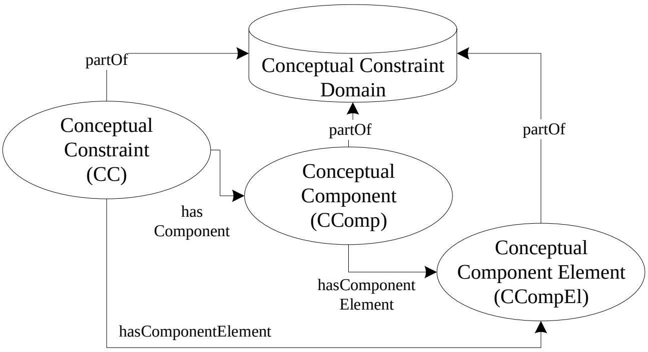


Figure 7.7.: General structure of the CC Domain (Domain Level)

This expression shows the elements playing a key role in different connections among the CC Domain: there are m_1 number of CCs, which define the constraint types including a CC taxonomy and connections of each CC that is defined as follows:

- Each CC may have one or many (m_2) CComps, and one or many (m_5) groups of CCompEls.

- Each CComp may have one or many (m_4) CCompEls (connected through m_3 relations to their corresponding CComps).
- Each one of the groups of CCompEls may have one or many (m_7) CCompEls (connected through m_6 number of relations to their corresponding CComps).

This CC Domain representation can be used to present a generic DMM expression as follows:

$$
\begin{aligned}
\min_{x} \quad & (CComp;\ Relation_{k_2};\ CCompEl_{k_3})_{k_1} && \text{(7.3a)} \\
\textbf{s.t.} \quad & CC[(CComp;\ Relation_{k_6};\ CCompEl_{k_7})_{k_5};\ (\ ; \\
& \qquad Relation_{k_9};\ CCompEl_{k_{10}})_{k_8}]_{k_4}. && \text{(7.3b)}
\end{aligned}
$$

where $k_1 = \{1, ..., n\}, ..., k_{10} = \{1, ..., n\}$ are integer numbers that may be different for each part of the expression. Expression (7.3a) represents the objective function, which can be expressed as a combination of one or more CComps aiming to be minimised (or maximised, or both), while Expression (7.3b) contains a set of CCs that represents the constraints of the DMM. Additionally, a dictionary structure that represents the specification of the CC Domain should be maintained. These relations can be formalized through a .xsd schema given in Section B.1 and illustrated in Figure B.1.

7.3.3. The Conceptual Constraint Domain: Building

A generic procedure is proposed to build the CC Domain:

(i) The first step is to construct a CC taxonomy by introducing the fundamental concepts to be managed as constraints.

(ii) Afterwards, the related CComps should be identified and connected to CCs at the same time.

(iii) Then, CCompEls should be introduced and connected to CComps and CCs.

As in the PS Domain building in Chapter 5, these steps associated to the CC Domain building can be further developed using the PDCA quality control cycle to support the ontology building.

7.3.4. The Conceptual Constraint Domain: An Illustrative Example

An introductory example is given before addressing the detailed explanations related to the development of different classes of CCs in Section 7.4.

Figure 7.8 depicts the typical sizing constraint related to the unit-task allocation procedure in a batch scheduling process. It is based on the identification of a binary variable (W) that connects the batch size (continuous variable, B) and the minimum value of the batch size ($Vmin$). This constraint applies to the different tasks (i), units (j) which can perform each task (K_i), and time intervals (t).

Two patterns are identified to represent this sizing constraint in the CC Domain. Pattern 1 defines the sizing constraint through the “BatchSizeAmount”, the “MinBatchCapacity”, and the “Allocation” CComps, which lead the constraint to the “BatchSizingCC”. On the other hand, the constraint can be mapped through the “ProcessedAmount” and the “MinProcessingCapacity” CComps as shown in pattern 2 where “Allocation”, “MinBatchCapacity” CComps are “partOf” the “ProcessedAmount” CComp; in this case, the constraint is modelled in the domain as a “SizingCC” superclass. Then, the constraint itself (constraint instance) can be placed as an instance of two CCs which actually are connected through “is-a” relations. Based on this idea, the presented modelling approach exploits the CCs to formulate the problem at a higher (more generic) level, which is used to represent a constraint in the CC Domain with different levels of detail.

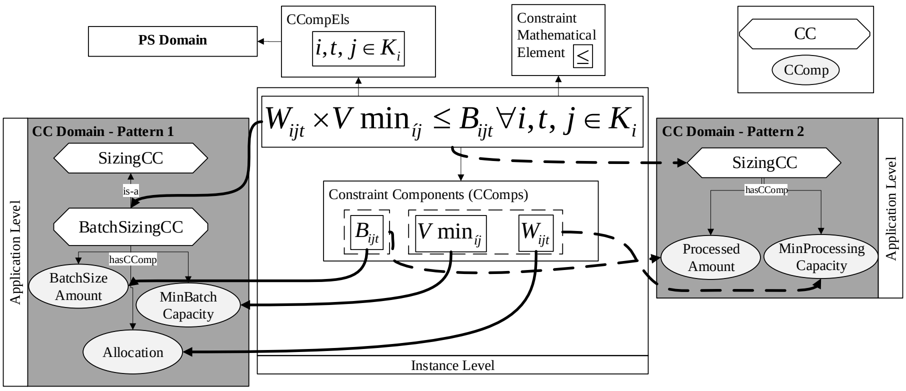


Figure 7.8.: A sizing constraint example illustrated with two patterns with the application and instance levels

Using the example in Figure 7.8 for illustrating the building procedure, the CC Domain construction steps are as follows:

(i) The “SizingCC” and the “BatchSizingCC” are introduced in the CC taxonomy,

(ii) The “BatchSize”, the “MinBatchCapacity” and the “Allocation” CComps are introduced and connected to the “BatchSizingCC” for the pattern 1,

(iii) The “ProcessedAmount” and the “MinProcessingCapacity” CComps are also introduced and connected to the “SizingCC” for the pattern 2, and

(iv) CCompEls “UnitProcedure” (i), “Unit” (j), and “TimePeriod” (t) are introduced as CCompEls and connected to each CComp.

7.4. Constraint Modelling in the Conceptual Constraint Domain

Generalised constraints can be introduced to the CC Domain from fundamental references, research articles, and review articles describing formulations at different decision-making levels. Specific features of these references are listed in Table B.1; the different classes of CCs identified during this procedure are explained from Section 7.4.1 to Section 7.4.6.

7.4.1. Balance Constraints

The conservation laws are the basis of the balance constraints that are included in the “BalanceCC” concept in the CC Domain. Balance constraints can be represented using different CComps, and sub-classes of the “BalanceCC” share similar CComps as it will be discussed during the development of this section.

The mass conservation is one of the important sub-classes of conservation laws that produce material balance constraints. The “MaterialBalanceCC” can be expressed in many ways including, in general, six main characteristics which can be identified through the following CComps: “ProducedMaterial”, “ConsumedMaterial”, “SoldMaterial”, “PurchasedMaterial”, “CurrentlyAvailableMaterial”, and “PreviouslyAvailableMaterial” (see Figure 7.6). Then, a material balance constraint can be modelled using the following equation (Kondili et al., 1993):

$$\begin{aligned} S_{s,t} = {} & S_{s,t-1} + \textstyle\sum_{i \in \overline{T}_s} \overline{\rho}_{i,s} \sum_{j \in K_i} B_{i,j,t-p_{i,s}} \\ - {} & \textstyle\sum_{i \in T_s} \rho_{i,s} \sum_{j \in K_i} B_{i,j,t} + R_{st} - D_{st}, \quad \forall s, t. \end{aligned} \tag{7.4}$$

Figure 7.9 depicts two patterns for constructing a “MaterialBalanceCC”: the first pattern is based on the “ConsumedMaterial” and “ProducedMaterial” CComps; for the second pattern, these CComps are further divided into additional CComps. Therefore, the additional CComps “MaterialBalanceInputCoeffcient”, “MaterialBalanceOutputCoeffcient”, and “BatchSize” CComps are included in this second pattern.

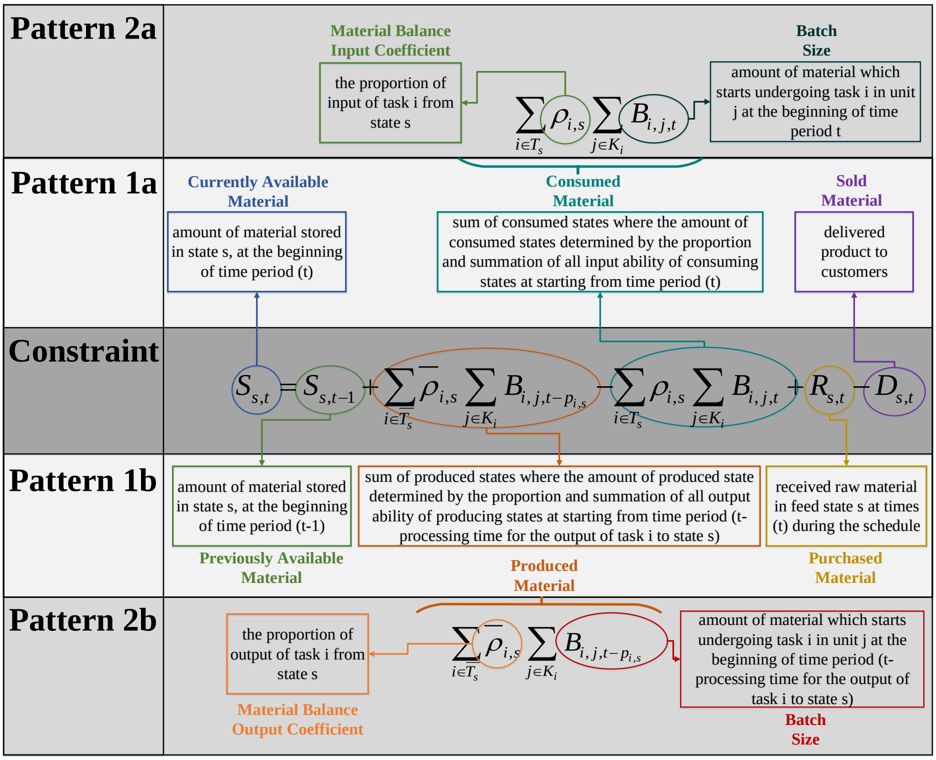


Figure 7.9.: Decomposition of material balance constraint (equation and explanations are taken from Kondili et al. (1993) and first pattern (1a and 1b) is adapted from Dombayci and Espuña (2018*b*))

Another way of connecting the “MaterialBalanceCC” to CComps is to analyse different material balance constraints found in the literature. For instance, Equations (7.5), (7.6), (7.7), (7.8) are shown subsequently with their CComps; the final derivation of “BalanceCC” is introduced as follows:

$$\begin{aligned} S_{s,t} = S_{s,t-1} \quad & + \textstyle\sum_{i\in\overline{T}_s} \overline{\rho}_{i,s} \sum_{j\in K_i} B_{i,j,t-p_{i,s}} \\ & - \textstyle\sum_{i\in T_s} \rho_{i,s} \sum_{j\in K_i} B_{i,j,t}, \forall s,t. \end{aligned} \tag{7.5}$$

Equation (7.5) (Kondili et al., 1993) is part of a general formulation of the batch plants scheduling problem using discrete time representation. The corresponding CComps and CCompEls (the ones directly connected to the CC) are identified in Table 7.1. The table also contains the corresponding mathematical expressions, showing the capabilities as a function of the CCompEls. Finally, the resulting identification of this pattern of the constraint regarding concepts and relations is shown in Figure 7.10.

Table 7.1.: CC Domain elements of Equation (7.5) (“MaterialBalanceCC”)

Mathematical Expression	**CComp**
$S_{s,t}$	StoredAmount
$S_{s,t-1}$	PreviouslyStoredAmount
$\sum_{i \in \overline{T}_s} \overline{\rho}_{i,s} \sum_{j \in K_i} B_{i,j,t-p_{i,s}}$	ProducedAmount
$\sum_{i \in T_s} \rho_{i,s} \sum_{j \in K_i} B_{i,j,t}$	ConsumedAmount

Mathematical Expression	**CCompEl**
$s \in S$	StateModel
$t \in T$	TimePeriod

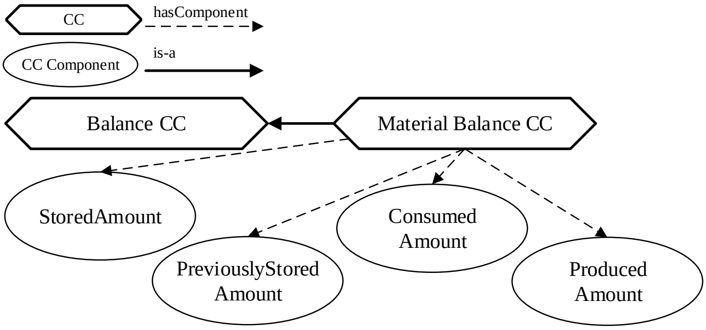


Figure 7.10.: “MaterialBalanceCC” development from Equation (7.5)

Another pattern for the “ProducedAmount” ($\sum_{i \in \overline{T}_s} \overline{\rho}_{i,s} \sum_{j \in K_i} B_{i,j,t-p_{i,s}}$) and the “ConsumedAmount” ($\sum_{i \in T_s} \rho_{i,s} \sum_{j \in K_i} B_{i,j,t}$) CComps may be identified by including the subsequent degree of granularity expressed by CComps “InputProportions”, “OutputProportions”, “PreviouslyProcessedAmount”[4], and “ProcessedAmount” CComps as shown in Table 7.2. The development of the resulting CC with this additional level of detail provides new concepts for the CC Domain and its updated relations. For instance, in the previous example, the “ConsumedMaterial” CComp is divided into two CComps: “InputProportion” and “PreviouslyProcessedAmount”, which lead to “partOf” relations among

---

[4] Since the production ends at this time period started to be processed “p” periods previously.

the affected CComps. The resulting structure of the "MaterialBalanceCC" is presented in Figure 7.11 (red and grey colours indicate the new relations and concepts).

Table 7.2.: CC Domain elements of Equation (7.5) (another level of expressions for the "ConsumedAmount" and "ProducedAmount" CComps)

Mathematical Expression	**CComp**
$\overline{\rho}_{i,s}$	InputProportion
$B_{i,j,t-p_{i,s}}$	PreviouslyProcessedAmount
$\rho_{i,s}$	OutputProportion
$B_{i,j,t}$	ProcessedAmount

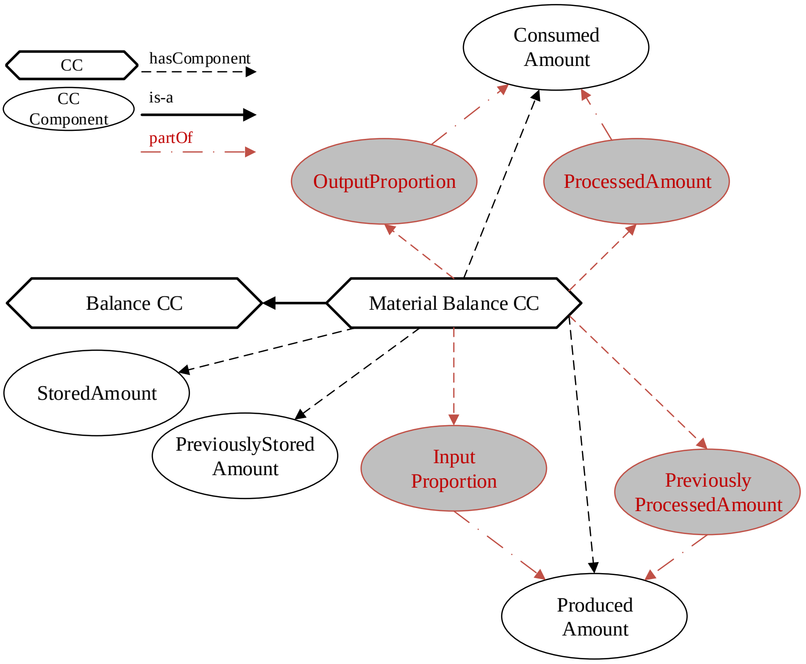


Figure 7.11.: "MaterialBalanceCC" development (another level of expressions for "ConsumedAmount" and "ProducedAmount")

The CC Domain may evolve not only with the introduction of new CComps which increase the granularity in terms of already included model description (as described in the previous step in Figure 7.11), but also with the introduction of new additionaly CComps. For example, the "MaterialBalanceCC" may include CComps associated to the "ReceivedAmount" (R_{st}) and "SentAmount" (D_{st}) CComps as shown in following equation:

$$\begin{aligned} S_{s,t} = \quad & S_{s,t-1} + \sum_{i \in \overline{T}_s} \overline{\rho}_{i,s} \sum_{j \in K_i} B_{i,j,t-p_{i,s}} \\ & - \sum_{i \in T_s} \rho_{i,s} \sum_{j \in K_i} B_{i,j,t} + R_{st} - D_{st}, \forall s, t. \end{aligned} \tag{7.6}$$

These additional CComps from Expression (7.6) are shown in Table 7.3 and Figure 7.12 depicts the updated connections of the “MaterialBalanceCC”.

Table 7.3.: CC Domain elements in the extended “MaterialBalanceCC” (Equation (7.6))

Mathematical Expression	**CComp**
R_{st}	ReceivedAmount
$D_{s,t}$	SentAmount

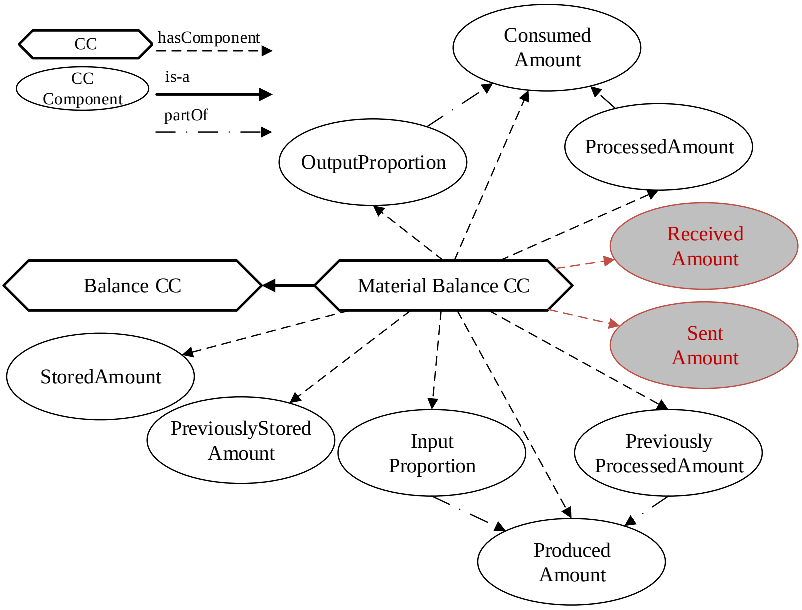


Figure 7.12.: “MaterialBalanceCC” development (adding more CComps to the constraint)

Other “MaterialBalanceCC” instances can be constructed around raw materials as in following equation (Ierapetritou and Floudas, 1998):

$$ST_0 + \sum_{i \in I_s} \rho^c_{si} \sum_{j \in J_i} B_{i,j,n} = 0, \forall\, s \in S^R. \tag{7.7}$$

Equation (7.7) is associated to the “StateModelInput” CCompEl, which has been defined in Section 7.2.1 as a part of the “StateModel”; the “StateModel” CComp includes all the input, the output, and the intermediates. The associated CC Domain elements of Equation (7.7) are illustrated in Table 7.4.

Table 7.4.: CC Domain elements of Equation (7.7)

Mathematical Expression	**CComp**
ST_0	StoredAmount
$\sum_{i \in I_s} \rho^c_{si} \sum_{j \in J_i} B_{i,j,n}$	ConsumedAmount

Mathematical Expression	**CCompEl**
$s \in S^R$	StateModelInput

The developed “MaterialBalanceCC” is also applicable to other levels at the multi-level hierarchies as proposed by (Mota et al., 2015). The corresponding CComps of the following equation are shown in Table 7.5:

$$\begin{aligned} S_{mi(t-1)} + &\quad \textstyle\sum_{\overline{m}j:(\overline{m},j,i)\in F} rp_{m\overline{m}} X_{\overline{m}jit} \\ = &\quad \textstyle\sum_{\overline{m}j:(\overline{m},j,i)\in F|F_s} rp_{m\overline{m}} X_{\overline{m}jit} \\ + &\quad S_{mit}, \forall (m,i) \in V_{nos} \wedge t \in T. \end{aligned} \tag{7.8}$$

As it can be seen, Equation (7.8) does not add any concept to the CC Domain; however, it changes the way how CCs and CComps are calculated. Therefore, the equation does not affect the application level where the elements of CC Domain appear but introduces additional calculations to the instance level.

Another used sub-class of the “BalanceCC” is the “EnergyBalanceCC” that is structurally similar to the “MaterialBalanceCC”. An example of an energy balance constraint (Silvente et al., 2015) is shown in the following equation:

$$SE_{k,t} = SE_{k,t-1} + \eta^{in}_k * Ld_{k,t} - \frac{SP_{k,t}}{\eta^{out}_k}, \forall k, t \in TRH. \tag{7.9}$$

Table 7.5.: CC Domain elements of Equation (7.8)

Mathematical Expression	**CComp**
$S_{mi(t-1)}$	PreviouslyStoredAmount
$\sum_{\overline{m}j:(\overline{m},j,i)\in F} rp_{m\overline{m}} X_{\overline{m}jit}$	ConsumedAmount
$\sum_{\overline{m}j:(\overline{m},j,i)\in F\|F_s} rp_{m\overline{m}} X_{\overline{m}jit}$	ProducedAmount
S_{mit}	CurrentlyStoredAmount

Mathematical Expression	**CCompEl**
$m, \in V_{nos}$	StateModelOutput
$i \in V_{nos}$	PhysicalModel
$t \in T$	TimePeriod

A detailed specification of the associated CC Domain components and elements of Equation (7.9) is shown in Table 7.6. In this case all these elements have been already identified in the previously described patterns. .

Table 7.6.: CC Domain elements in the “EnergyBalanceCC” (Equation (7.9))

Mathematical Expression	**CComp**
$SE_{k,t}$	StoredAmount
$SE_{k,t-1}$	PreviouslyStoredAmount
$\eta_k^{in} * Ld_{k,t}$	InputAmount
$SP_{k,t}/\eta_k^{out}$	OutputAmount

Mathematical Expression	**CCompEl**
k	PhysicalModel
$t \in TRH$	TimePeriod

7.4.2. Sequencing Constraints

Another type of the CCs is the “SequencingCC” that expresses the need to maintain a temporal path among the different procedures involved in a production process. Consequently, it combines the allocation components connected to the procedural and the physical models as well as the sequencing requirements; it includes the “Allocation” and the “TimeModel” related CComps (e.g. the “StartingTime” CComp), decision variables on sequencing allocations, and information related to sequencing (e.g. the “SequencingPolicy” CComp).

A partial presentation of the “SequencingCC” and its connections to CComps are shown in Figure 7.13. Several branches can be identified in the “SequencingCC” as being the most usual ones. Then, the “SequenceDependentChangeOverCC”, which may include “SequencingRequirement” and “SequencingPolicy” CComps in the mathematical representation of a PS DMM. The figure also allows the comparison between the “SequenceDependentChangeOverCC” and “SequenceDependentCleaningChangeoverCC” which uses the “CleaningTime” CComp in addition to the comparison between the “SequenceDependentChangeOverCC” and “SequenceDependentFormatChangeoverCC” which uses the “FormatChangingTime” CComp. These comparisons show an example related to the variety of CCs with the effect of CComp connections.

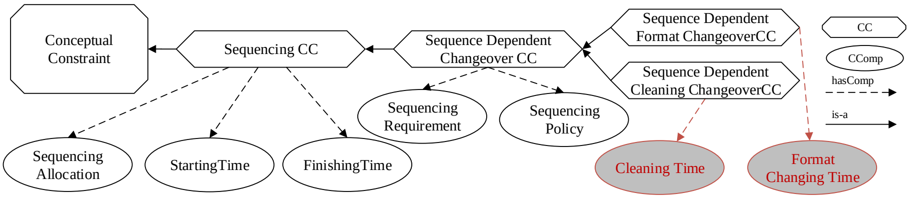


Figure 7.13.: “SequencingCC” and its CComp connections (adapted from Dombayci, Capón-García, Muñoz and Espuña (2017))

An example of the “SequencingCC” and related connections are shown in following equation (Ierapetritou and Floudas, 1998):

$$\begin{aligned} T^s(i,j,n+1) \geq \quad & T^f(i',j'n) \\ - \quad & H(2 - wv(i',n) - wv(i,n+1)) \\ & \forall i \in I_j, i' \in I_j, j,j' \in J, n \in N. \end{aligned} \tag{7.10}$$

Equation (7.10) has different tasks in different units if j operates continuous mode and j’ operates batch mode. The CComps of Equation (7.10) and the CCompEl connections of the constraint are introduced in in Table 7.7 giving the CC connections with the additional information; the first CCompEl is defined as “UnitProcedure, Unit (Capability)” which is expressed as (; Capability; UnitProcedure; Unit) using Expression (7.2). Here, the mathematical expression ($i, i' \in J_i$) requires two CCompEls (“Unit” and “UnitProcedure”) to express a “Relation” between these two CCompEls. In the case of this example, the relation is the “Capability”. In these cases, the resulting subset does not introduce all the “Unit” instances, but a subset of the “Unit” instances which have the “Capability” of executing “UnitProcedure” instances.

Table 7.7.: CC Domain elements of “SequencingCC” (Equation (7.10))

Mathematical Expression	**CComp**
$T^s(i, j, n+1)$	StartingTime
$T^f(i', j'n)$	FinishingTime
$H(2 - wv(i', n) - wv(i, n+1))$	SequencingAllocation

Mathematical Expression	**CCompEl**
$i, i' \in I_j$	UnitProcedure, Unit (Capability)
$j, j' \in J$	Unit
$n \in N$	EventPoint

7.4.3. Resource Allocation Constraints

The resource allocation constraints are connected to the allocation information of the data, which may contain PS-based data. This data may include the procedural control model, the physical model, and the time model. Decisions in the “ResourceAllocationCC” are connected to the “Allocation” CComps, and the binary comparison structure supports the decisions in this CC. This CC makes decisions over possible executions of the “ProceduralControlModel” elements on the contrary of time-related decisions as in the “SequencingCC”.

An example of the “ResourceAllocationCC” is given in the following Equation (Kondili et al., 1993) and Table 7.8 shows the CComp and CCompEl connections:

$$\sum_i W_{ijt} \leq 1, \forall j, i \in I_j, t. \tag{7.11}$$

Table 7.8.: CC Domain elements of Equation (7.11)

Mathematical Expression	**CComp**
$\sum_{i=I_j} W_{ijt}$	Allocation

Mathematical Expression	**CCompEl**
$i \in I_j$	UnitProcedure, Unit (Capability)
j	Unit
t	TimePeriod

In addition to the constraint given in Equation (7.11), another example of the "ResourceAllocationCC" is shown in the following equation (Kondili et al., 1993):

$$\sum_{i'=I_j} \sum_{t'=t}^{t+p_i-1} \left(W_{i'jt'} - 1\right) \quad \leq \quad M\left(1 - W_{ijt}\right), \forall j, i \in I_j, t. \tag{7.12}$$

Table 7.9 illustrates the CC Domain elements of Equation (7.12). This constraint has another CComp that is related to the BigM formulation, and it maintains the allocation along the time.

Table 7.9.: CC Domain elements of the "ResourceAllocationCC" (Equation (7.12))

Mathematical Expression	**CComp**
$\sum_{i'=I_j} \sum_{t'=t}^{t+p_i-1} (W_{i'jt'} - 1)$	Allocation
$M\left(1 - W_{ijt}\right)$	BigMAllocation

Mathematical Expression	**CCompEl**
$i = I_j$	UnitProcedure, Unit (Capability)
j	Unit
t	TimePeriod

These two constraints (given in Equations (7.11) and (7.12)) are different regarding mathematical expressions; while the first one has the comparison for summation of all procedures, the second one considers the allocation information including the BigM formulation, which has a sufficiently large number M. However, regarding conceptual comparison, both constraints seek decisions on the “Allocation” CComps that fulfil the expressed constraints. Therefore, they are aligned regarding CComps.

Apart from these two given constraints, the “ResourceAllocationCC” may be changed depending on their object that is allocated. In this case, one option to separate the resources as a physical model and other resources; then, these can be more detailed depending on the concept of the “PhysicalModel” element and the other type of the resource (e.g. energy, utility, water as in the “BalanceCC”).

7.4.4. Timing Constraints

Timing constraints control the event points (for the continuous time formulations) and time points (for the discrete time formulations) and manage the duration of “ProceduralControlModel” elements. The “TimingCC” is similar to the “SequencingCC” regarding allocation related binary variables. The usage of the “TimingCC” that the time duration information is aiming to ensure the consistency of starting and ending time of procedures taking into account the duration.

There are many ways to represent the “TimingCC”, for instance (Méndez and Cerdá, 2002):

$$C_i - \sum_{i \in J_i} L_{ij} \leq C_{i'} - \sum_{j \in J_{i'j}} L_{i'j} + H(1 - U_{ii'}), \forall i \in I_s^+, i' \in I_s^-, s \in S. \tag{7.13}$$

The timing constraint in Equation (7.13) ensures the consistency of the starting and ending times of the procedures. The associated CC Domain elements are introduced as in Table 7.10. Here, there is an advantage of expressing “StateModel” with input, output, and intermediates; the particular condition of this case is the “+” expression used in the CCompEl column. This condition uses both the “StateModelIntermediate” CCompEl and the “StateModelOutput” CCompEl together excluding the “StateModelInput” CCompEl.

Table 7.10.: CC Domain elements of Equation (7.13)

Mathematical Expression	**CComp**
C_i	ProducedAmountCompletionTime
$\sum_{i \in J_i} L_{ij}$	DurationOfProductionProcedure
$C_{i'}$	ConsumedAmountCompletionTime
$\sum_{j \in J_{i'j}} L_{i'j}$	DurationOfConsumtionProcedure
$H(1 - U_{ii'})$	SequencingRequirement

Mathematical Expression	**CCompEl**
$i \in I_s^+$	UnitProcedure, StateOutput (Capability)
$i' \in I_s^-$	UnitProcedure, StateInput (Capability)
$s \in S$	StateIntermediate + StateOutput

The “TimingCC” can also be expressed as following equation (Méndez et al., 2006), being the associated CC Domain elements presented in Table 7.11:

$$Ts_i \geq Tf_{i'} + \sum_{j \in J_{ii'}} cl_{i'ij} X_{i'ij} - M(1 - \sum_{j \in J_{ii'}} X_{i'ij}) \forall i, i'. \tag{7.14}$$

Table 7.11.: CC Domain elements of Equation (7.14)

Mathematical Expression	**CComp**
Ts_i	ProcedureStartTime
$Tf_{i'}$	ProcedureFinishTime
$\sum_{j \in J_{ii'}} cl_{i'ij} X_{i'ij}$	ChangeOverSequencingRequirement
$M(1 - \sum_{j \in J_{ii'}} X_{i'ij})$	Allocation

Mathematical Expression	**CCompEl**
i	UnitProcedure, StateOutput(Capability)
i'	UnitProcedure, StateInput (Capability)

7.4.5. Sizing Constraints

Sizing constraints control the upper and/or lower bounds of the different components in the DMMs. The “Sizing CC” is relatively simple structured when comparing to other CCs since it ensures the values of variables as well as the accomplishment of any requirement related to limits.

In Section 7.3.4, an example of “SizingCC” and its connections with a mathematical expression has been expressed (see Figure 7.8). The investigated “SizingCC” in that figure is also separated in constraint components which represent the intermediate steps of the constraint construction. The constraint elements are shown here, where each element is connected to the PS Domain knowledge model. Likewise, the constraint connections and the mathematical elements in the depicted constraint are also shown in the figure.

Another sizing constraint is introduced for the upper bound of the “StoredAmount” CComp (appears many times in the development of “BalanceCC”). The connections are presented in Table 7.12.

$$S_{st} \leq C_s, \forall s, t. \tag{7.15}$$

Table 7.12.: CC Domain elements of the “Sizing CC” (Equation (7.15))

Mathematical Expression	**CComp**
S_{st}	StoredAmount
C_s	UpperBoundOfStoredAmount

Mathematical Expression	**CCompEl**
s	StateModel
t	TimePeriod

7.4.6. Economic Constraints

Economic constrains are related to calculations which are related with money; for instance, the cost, profit, and net values of the procedures by including the price related amounts and processed amounts (e.g. “Price” - “PurchasedAmount”, “Price” - “SoldAmount”).

The "EconomicCC" introduced in this section is generally used in objective function calculations among other elements. The "EconomicCC" is usually depicted as following constraint (Guillén-Gosálbez et al., 2005):

$$Rev_{ts} = \sum_{pk} Sales_{pkts}\ Price_{pkt}, \forall\, t, s. \tag{7.16}$$

Equation (7.16) is illustrated with its CComps and CCompEls in Table 7.13 and the "TotalSaleAmount" CComp is calculated in another constraint given in Equation (7.17).

Table 7.13.: CC Domain elements of Equation (7.16)

Mathematical Expression	**CComp**
Rev_{ts}	TotalRevenue
$Sales_{pkts}$	TotalSaleAmount
$Price_{pkt}$	SellingPrice

Mathematical Expression	**CCompEl**
s	StateModel
t	TimePeriod

The constraint from Equation (7.16) has been added by the DMM builder in order to calculate the "TotalRevenue" CComp. However, one of the CComps, "TotalSaleAmount", has to be calculated with another constraint as follows:

$$Sales_{pkts} = \sum_{j} Y_{pjkts}, \forall\, p, k, t, s. \tag{7.17}$$

The CC Domain elements of Equation (7.17) are given in Table 7.14.

Table 7.14.: CC Domain elements of Equation (7.17)

Mathematical Expression	**CComp**
$Sales_{pkts}$	TotalSaleAmount
Y_{pjkts}	TransferedAmount

Mathematical Expression	**CCompEl**
p	OutputState
k	PhysicalModel
t	TimePeriod
s	StateModel

7.4.7. Resulting CC and CComp Taxonomies

In previous subsections, some of the relations among the classes and subclasses of CC have been identified departing from the generic relations between the CC Domain elements initially identified in Section 7.3.1 (Figure 7.7). These relations have resulted in a CC taxonomy at the application level presented in Figure 7.14. Currently, the first level of the CC taxonomy consists of six CCs ((identified through Sections 7.4.1-7.4.6). Then, these constraints are detailed according to the different occurrences of CComps.

In summary:

(i) The “BalanceCC” shows a common structure for its two subclasses (“MaterialBalanceCC” and “EnergyBalanceCC”) and includes CComps in terms of being input and output; they are connected to the CCompEls, which appear in the “StateModel”, “PhysicalModel”, “ProceduralControlModel”, “TimeModel” CComp that construct the “Balance CC” may vary from one object to another (e.g. material, energy, utility, demand) depending on the problem.

(ii) Another class of CC is the “SequencingCC” that includes the “Allocation” and the “TimeModel” related CComps (e.g. the “StartingTime” CComp), decision variables on allocation, and information related to sequencing (e.g. the “SequencingPolicy” CComp).

(iii) At this point, the “ResourceAllocationCC” represents decisions over possible executions of the “ProceduralControlModel” elements of this execution, which is the part where the “SequencingCC” is separated from this CC with the sequence related decisions. The “ResourceAllocationCC” contains the “Allocation” related CComps as variables.

(iv) The “TimingCC” is similar to the “SequencingCC” regarding time duration information and time-related variables. However, the “TimingCC” uses time-related variables to ensure the consistency of the starting and ending time of procedures.

(v) Another subclass of CC is the “Sizing CC” that has a relatively simple structure for limiting variables, as well as the accomplishment of any requirement related to the upper and lower bounds of any variable in the model.

(vi) The “EconomicCC” is related to the calculations that are related with financial resources. For instance, the cost, profit, and net values of the procedures by including the price related amounts and processed amounts (e.g. “Price” - “PurchasedAmount”, “Price” - “SoldAmount”), as well as utilities and manpower, can be managed using this CC.

Depending on the details of the constraint pieces, the “partOf” relations among the CComps (i.e. connected to the CCs) can be easily established. These CComps may contain different levels of information or appear as “partOf” more than one CComps. This is the case of the “MaterialBalanceCC” in Section 7.4.1, in which the “ConsumedAmount” CComp is composed of two CComps: “InputProportion” and “PreviouslyProcessedAmount”, which lead to the “partOf” links among these CComps as shown in Figure 7.15. The object properties established between the CComps, are transitive and support the modelling of higher level CComps. For instance, the “InputProportion” is part of the “ConsumedAmount” but it also contains the “Input” CComp. The second relation is a deduction, which uses the ancestor relations. Another essential CComp in Figure 7.15 is the “BatchSize”, which appears at both top CComps (the “Input” and the “Output” CComps) of the “MaterialBalanceCC”. These CComps are combined with other CComps (“InputCoeffcient” and “OutputCoeffcient”) to produce another CComps (“ConsumedMaterial” and “ProducedMaterial”).

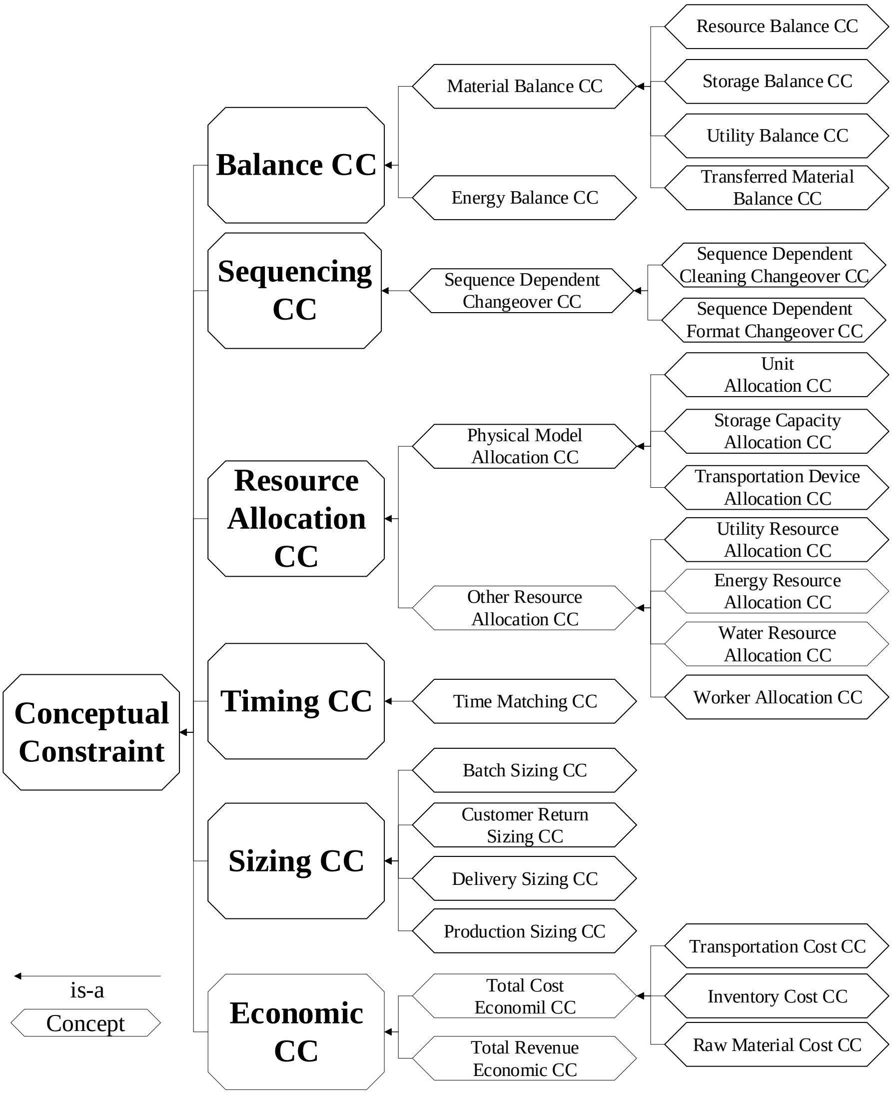


Figure 7.14.: CC taxonomy of the CC Domain

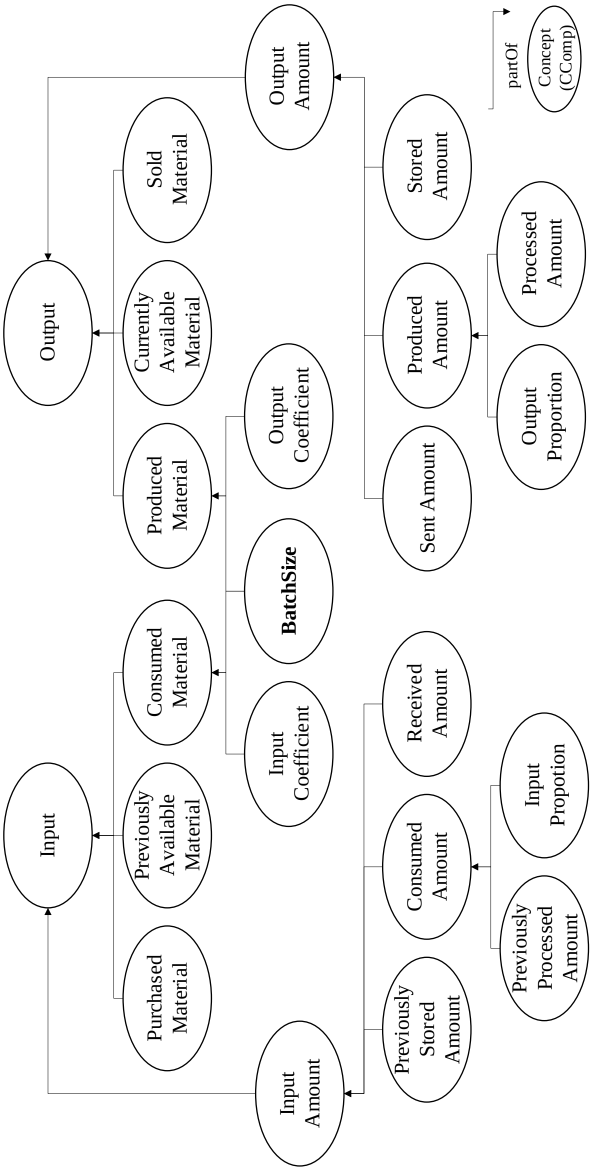


Figure 7.15.: "partOf" relations among CComp of the "MaterialBalanceCC"

7.5. Integration of the PS Domain and the CC Domain

As indicated in Section 7.2, an essential element for the proposed PS Domain and the CC Domain integration is the link between the CC Domain Elements (i.e. CComps, CCompEls) and the query types explained from the PS Domain. These links identify the DMMs elements (i.e. sets, subsets, parameters, variables, and mathematical expressions) as illustrated in Table 7.15. Sets and subsets are connected to the type 1 queries that retrieve instances or instances with relations, while the elements of DMMs that contain CComps are connected to the type 2 queries in order to retrieve data properties. These PS Domain queries (type 1 and type 2) can be identified as the integrated PS Domain queries since they use available elements from the CC Domain.

Table 7.15.: Summary of the CC and PS Domains integration

Role in a DMM	**CC Domain**	**Integrated PS Domain queries**
Set	(; ; CCompEl)	Type 1: *hasInstance* CCompEl
Subset	(; Relation; $CCompEl_1$, $CCompEl_2$)	Type 1: *hasInstance* ($CCompEl_1$ ObjectPropertyFromRelation $CCompEl_2$)
Parameter, Variable, Expression	(CComp; ; CCompEl)	Type 2: CCompEl *dataPropertyFromCComp*
Parameter, Variable, Expression	(CComp; Relation; $CComEl_1$, $CCompEl_2$)	Type 2: *hasInstance*($CCompEl_1$ *ObjectPropertyFromRelation* $CCompEl_2$) *dataPropertyFromCComp*

Figure 7.16 shows the connections required to achieve this domain integration, both at the application level and at the domain level, and including the respective generic domain structures (Figure 7.2 for the PS Domain and Figure 7.7 for the CC Domain). Additionally, different CComps and CCompEls are included on the application level, while these elements of the CC Domain are indicated using the following expression based on Expression 7.2:

$$\begin{aligned} CC \quad & [\,(CComp_1;\ ;\ CCompEl_1); \\ & (\ ;\ Relation_1;\ CCompEl_1, CCompEl_2)\,]_1 \end{aligned} \tag{7.18}$$

Expression (7.18) summarizes connections shown in Figure 7.16: a CC (CC_1) is connected to a CComp ($CComp_1$) and two CCompEls ($CCompEl_1, CCompEl_2$) while the CComp1 has a CCompEl ($CCompEl_1$) connection. The CCompEls ($CCompEl_1, CCompEl_2$) are the link to the corresponding concepts ("$Model_1$", "$Model_2$") at the PS Domain.

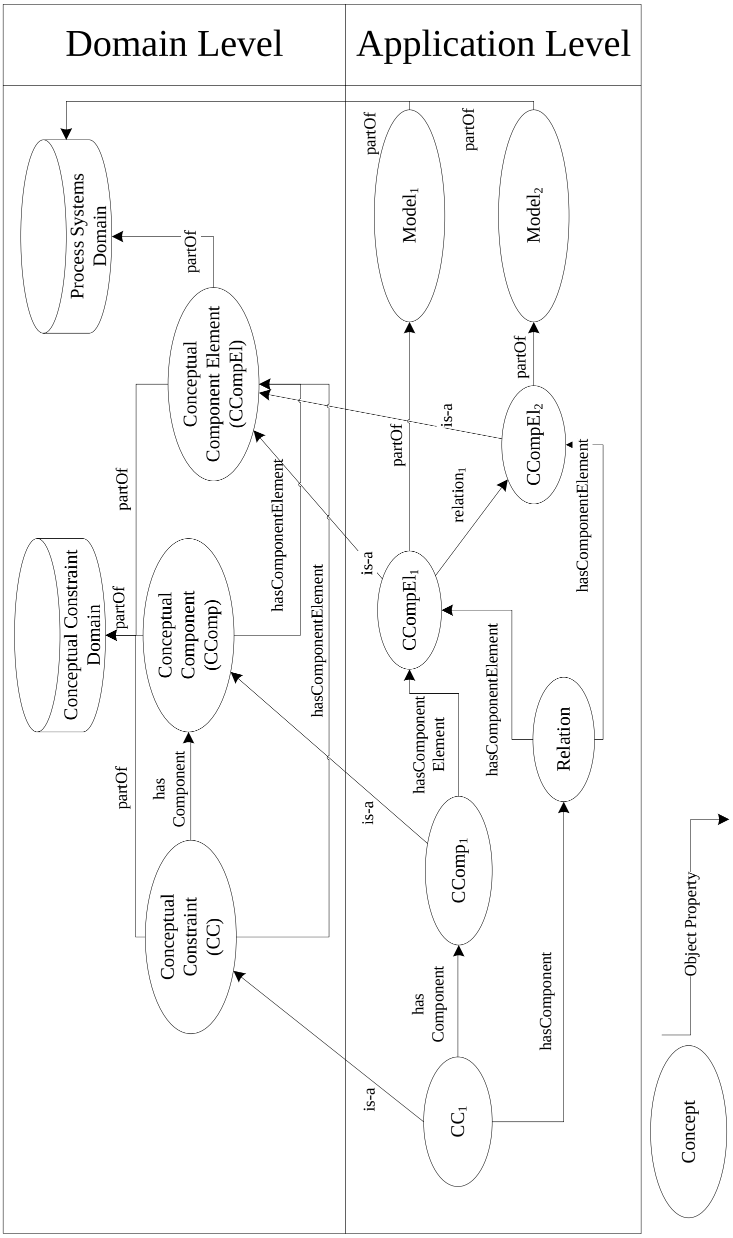


Figure 7.16.: Integration between the CC Domain and the PS Domain (belong to Expression (7.18))

7.6. Illustrative Case Studies

7.6.1. Domain Integration

With the purpose of introducing a “real” and extended version of Expression 7.18 is detailed. The usual sizing constraint that forces the production to cover the demand of the production with the demand used, the main elements of this constraint are illustrated in Table 7.16 and expressed as follows:

$$\begin{aligned} ProductionSizingCC \quad & [\,(ProducedAmount;\ Capability;\\ & \quad UnitProcedure, Unit), && \text{(7.19a)}\\ & (Demand;\ ;\ ); && \text{(7.19b)}\\ & (\ ;\ Capability;\ UnitProcedure, Unit), && \text{(7.19c)}\\ & (\ ;\ ;\ Unit)\,] && \text{(7.19d)} \end{aligned}$$

Table 7.16.: Illustrative example for the PS Domain and the CC Domain integration with the “ProductionSizingCC”

$$\boxed{\sum_{i \in I_j} \sum_j P_{i,j} \geq Dem, \forall i \in I_j, j.} \longrightarrow ProductionSizingCC$$

Mathematical Expression	CComp
$\sum_{i \in I_j} \sum_j P_{i,j}$	ProducedAmount
Dem	Demand

Mathematical Expression	CCompEl
$i \in I_j$	UnitProcedure, Unit (Capability)
j	Unit

Expressions (7.19a)-(7.19d) indicated that a “ProductionSizingCC” should include two CComps (“ProducedAmount” and “Demand”) connected to the “UnitProcedure” and “Unit” CCompEls. This “ProductionSizingCC” example enables introducing the instance level in addition to the domain level and the application level. Figure 7.17 depicts a portion (for the sake of clarity) of this “ProductionSizingCC” example. All the concepts at the application level are by with the taxonomical relations “is-a” and “partOf” to their corresponding concepts at the domain level. Additionally, concepts at the application level have similar connections as the ones at the domain level. In the instance level, all the instances and their values are connected to the application level with the “hasInstance” assertion. Finally, each instance is connected according to its concept type, and relations from the example introduced in Table 7.16.

Next, explanations from the “ProductionSizingCC” example are given as follows:

(i) Expression (7.19a) (ProducedAmount; Capability; $UnitProcedure$, $Unit$) - Type 2[5]: gets the produced amount of the connected element(s). If the connected elements of the “ProducedAmount” were only the “UnitProcedure”, then the amount of all the produced values for each reaction would have been retrieved (not drawn into the application and instance levels in Figure 7.17). However, the “ProducedAmount” CComp is connected to both “UnitProcedure” and “Unit” CCompEls and the nature of the depicted data given in the instance level forces it to only reach one value: “$\#producedAmount\ in\ reaction_1$” (red colour).

(ii) Expression (7.19b) (Demand; ;) - Type 2: gets the demand value. There will be only one value since there is no connection with any CCompEls (not drawn in Figure 7.17).

(iii) Expression (7.19c) (; Capability; $UnitProcedure$, $Unit$) - Type 1 : gets all the “UnitProcedure” and “Unit” CCompEls pairs (“UnitProcedure ‘hasCapability’ Unit”). In this example, only one relation exists between the instances “$reaction_1$ - $reactor_1$” pair (purple colour).

(iv) Expression (7.19d) (; ;$Unit$) - Type 1: gets all possible instances of the “Unit” CCompEl. The output is the instance “$reactor_1$” (blue colour).

---

[5] Type of queries are explained in Section 7.2.2.

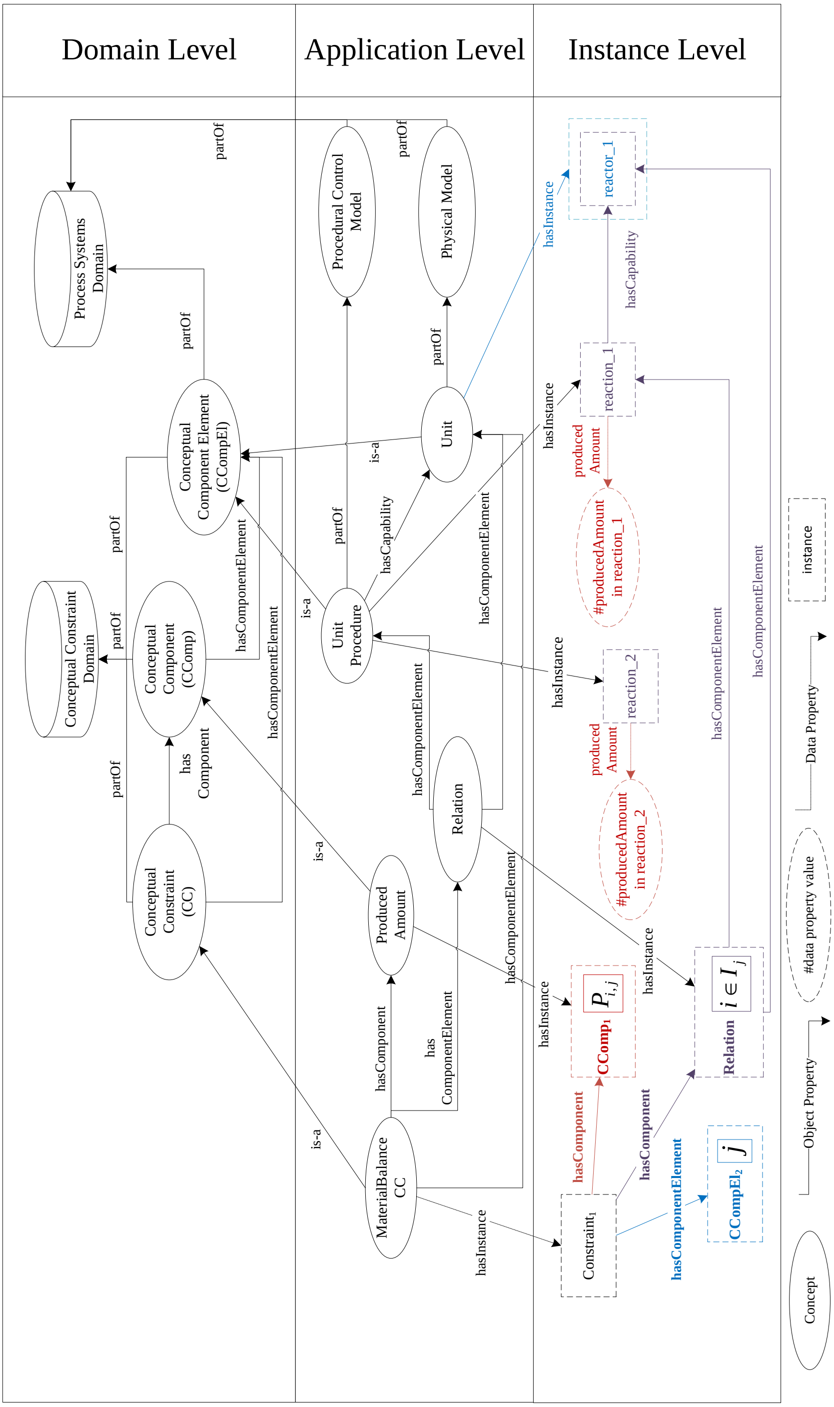


Figure 7.17.: Connections between the CC and PS Domains

7.6.2. Material Balance Constraint

This section illustrates a case where a material balance constraint is identified using the connections between the CC Domain and the PS Domain. Afterwards, the same connections are used to recreate this constraint at another level in the decision-making hierarchy.

Departing from the DMM based on a state-task network presentation, both proposed by Kondili et al. (1993), the previously defined procedures evolve as follows:

(i) A dictionary is populated according to the Comps and the CCompEls described in the CC Domain model building section (Section 7.4). The “CurrentlyAvailableMaterial” CComp is presented using the complete dictionary specifications in Table 7.17. The complete list of CComp is available in Appendix E.

(ii) The required information for identifying the material balance from the original formulation (Equation (7.20) originally given in Equation 7.4) is detailed in Tables 7.18 for sets, Table 7.19 for subsets, Table 7.20 for parameters, and Table 7.21 for variables. Tables 7.19-7.21 also present the CC Domain information of the corresponding CComps as in Expression 7.2. These tables are built using the reduced structure of dictionary specifications.

$$S_{s,t} = S_{s,t-1} + \sum_{i \in \overline{T}_s} \overline{\rho}_{i,s} \sum_{j \in K_i} B_{i,j,t-p_{i,s}} - \sum_{i \in T_s} \rho_{i,s} \sum_{j \in K_i} B_{i,j,t} + R_{s,t} - D_{s,t}, \quad \forall s \in S, \forall t \in T, t \neq t_0. \tag{7.20}$$

Table 7.17.: "CurrentlyAvailableMaterial" CComp illustration using the dictionary specifications

Tag	Explanation	Example Value
Source	Source of the information	Kondili1993
CC	Constraint information in the source	Constraint_05
CC name	Constraint information	MaterialBalanceCC
Nomenclature	Nomenclature information from the source	$S_{s,t}$ amount of material stored in state S, at the beginning of time period t
Symbol	Symbol that is used in the source	S(s,t)
Main symbol	Symbol without index connection(s)	S
Entity	Type of the entity (Set, Parameter or Variable)	Variable
CComp name	Meaning of the symbol that is connected to the CC (If there are more than one they are separated with commas and if the Component is a set this tag will be empty and next tag will define)	CurrentlyAvailableMaterial
CCompEl symbol	One of the connections of the symbol	s
CCompEl	Meaning of one of the connections in CC & PS Domains	ProcessInput, ProcessIntermediate, ProcessOutput
CCompEl symbol	One of the connections of the symbol	t
CCompEl	Meaning of one of the connections in CC & PS Domains	TimePeriod

Table 7.18.: Nomenclature details and the CC Domain expressions of Equation (7.20) (Sets)

Nomenclature	Symbol	Connection CCompEl
State	s	ProcessInput, ProcessIntermediate, ProcessOutput (or ProcessState)
Task	i	UnitProcedure
A piece of equipment	j	Unit
Time	t	TimePeriod

Table 7.19.: Nomenclature details and the CC Domain expressions of Equation (7.20) (Subsets)

Nomenclature	**Symbol**	**Relation**	**CCompEl symbol**	**Connection CCompEl**
set of unit procedures receiving material from state s	T_s	Capability	s	ProcessStateInput
				UnitProcedure
CC Domain Expression →		**(; Capability; ProcessStateInput, UnitProcedure)**		
set of unit procedures producing material from state s	$\overline{T}_s$	Capability	s	ProcessStateOutput
				UnitProcedure
CC Domain Expression →		**(; Capability; ProcessStateOutput, UnitProcedure)**		
set of units capable of performing unit procedure i	K_i	Capability	i	UnitProcedure
				Unit
CC Domain Expression →		**(; Capability; Unit, UnitProcedure)**		

Table 7.20.: Nomenclature details and the CC Domain expressions of Equation (7.20) (Parameters)

Nomenclature	Symbol	CComp	CCompEl symbol	Connection CCompEl
the proportion of input of unit procedure i to state	$\rho_{i,s}$	MaterialBalanceInputCoefficient	i	UnitProcedure
			s	ProcessState
CC Domain Expression →		**(MaterialBalanceInputCoefficient;Capability; UnitProcedure, ProcessState)**		
the proportion of output of unit procedure i to state	$\overline{\rho}_{i,s}$	MaterialBalanceOutput Coefficient	i	UnitProcedure
			s	ProcessState
CC Domain Expression →		**(MaterialBalanceOutputCoefficient; Capability; UnitProcedure, ProcessState)**		
processing time for the output of unit procedure i to state	$p_{i,s}$	ProcessingTime	i	UnitProcedure
			s	ProcessState
CC Domain Expression →		**(ProcessingTime; Capability; UnitProcedure, ProcessState)**		

Table 7.21.: Nomenclature details and the CC Domain expressions of Equation (7.20) (Variables)

Nomenclature	Symbol	CComp	CCompEl symbol	Connection CCompEl
amount of material stored in state S, at the beginning of time period t	$S_{s,t}$	CurrentlyAvailableMaterial	s	ProcessState
			t	TimePeriod
CC Domain Expression →		**(CurrentlyAvailableMaterial; Capability; ProcessState,TimePeriod)**		
amount of material which starts undergoing unit procedure i in unit j at the beginning of time period t	$B_{i,j,t}$	BatchSize	i	UnitProcedure
			j	Unit
			t	TimePeriod
CC Domain Expression →		**(BatchSize; Capability, Capability ;UnitProcedure,Unit,TimePeriod)**		
amount of material s which is purchased of time period t	$R_{s,t}$	PurchasedMaterial	s	ProcessState
			t	TimePeriod
CC Domain Expression →		**(PurchasedMaterial; (; Capability; ProcessState,TimePeriod)**		
amount of material s which is sold of time period t	$D_{s,t}$	SoldMaterial	s	ProcessState
			t	TimePeriod
CC Domain Expression →		**(SoldMaterial; Capability; ProcessState,TimePeriod)**		

Subsequently, the following changes that introduce information connections from the “Unit” level to the “Site” level are implemented in the CCompEls handling, following the links in Figure 7.5, which explains the model behind in the PS Domain:

- “Unit” ⟶ “Site”
- “UnitProcedure” ⟶ “SiteProcedure”
- “ProcessInput” ⟶ “EnterpriseProcessSegmentInput”
- “ProcessIntermediate” ⟶ “EnterpriseProcessSegmentIntermediate”
- “ProcessInput” ⟶ “EnterpriseProcessSegmentOutput”

Finally, a new constraint definition is built using the CC Domain, as illustrated in Table 7.22. The new constraint uses the same mathematical structure introduced in Equation (7.20). The table shows the new CC Domain expressions belong to the recreated constraint. According to the reconstruction, Equation (7.20) remains the same, and the connected elements of the equation are changed, resulting in a material balance for a planning problem instance.

Table 7.22.: Recreated constraint elements of “MaterialBalanceCC” in Equation (7.20)

Symbol	**CC Domain Expression**
s	(; ;EnterpriseProcessSegmentInput) + (; ;EnterpriseProcessSegmentIntermediate) + (; ; EnterpriseProcessSegmentOutput)
i	(; ; SiteProcedure)
j	(; ; Site)
t	(; ; TimePeriod)
T_s	(; Capability; EnterpriseProcessSegmentInput, SiteProcedure)
$\overline{T}_s$	(; Capability; EnterpriseProcessSegmentOutput, SiteProcedure)
K_i	(; Capability; Site, SiteProcedure)
$\rho_{i,s}$	(MaterialBalanceInputCoefficient; Capability; SiteProcedure, EnterpriseProcessSegmentInput)
$\overline{\rho}_{i,s}$	(MaterialBalanceOutputCoefficient; Capability; SiteProcedure, EnterpriseProcessSegmentOutput)
$p_{i,s}$	(ProcessingTime; Capability; SiteProcedure, EnterpriseState)
$S_{s,t}$	(CurrentlyAvailableMaterial; Capability; EnterpriseState,TimePeriod)
$B_{i,j,t}$	(BatchSize; Capability, Capability; SiteProcedure,Site,TimePeriod)
$R_{s,t}$	(PurchasedMaterial; Capability; EnterpriseState,TimePeriod)
$D_{s,t}$	(SoldMaterial; Capability; EnterpriseState,TimePeriod)

7.6.3. Results of the Cases

The CC Domain successfully identified a material balance constraint from a scheduling DMM in Section 7.6.2. Next, the constraint was regenerated to be used in another level of the multi-level hierarchy using the PS Domain concepts. As a result, the potential of using the CC Domain and the re-use of conceptualised DMMs at different hierarchical levels of the decision-making structure, have been introduced with the following benefits in the case study:

- Constraints have been identified in the CC Domain that enable sharing a common understanding of humans, as well as understandable by the computers.
- The integration of the CC and the PS Domains has provided a basis for the knowledge management and integration between the PS Domain and the CC Domain (domains integration) by using the table expressed in Section 7.5.
- The constraint integration has been introduced by giving a new identity.
- The automatic generation of DMMs has been initiated by using the CC Domain, together with algorithms and the structure of the dictionary specifications.

7.7. Concluding Remarks

The general aim of the working framework proposed in this chapter was to support the automatic Decision-Making Model (DMM) building by conceptualising constraints, in order to be able to generate new DMMs specifically adapted to solve other problem instances. Therefore, a modelling methodology for DMMs is built through the development of a Conceptual Constraint (CC) Domain that supports the design phase of decision-making procedures. After the development of the CC Domain, mathematical models are identified and can be used by other scenarios in situations such as integration of DMMs at the hierarchical levels with the developed identification and integration procedures.

This work has developed the modelling methodology that preserves the main patterns related to different constraints and dictionary structures. The patterns are constructed from the CC Domain and its elements: Conceptual Constraints (CCs), Conceptual Components (CComp), and Conceptual Component Elements (CCompEls). The CC Domain taxonomy presents the CCs, and each pattern of CCs introduces the CComps. The CC Domain elements are illustrated at the domain level, which has connections to the application level for the concepts placed under the type of the CC Domain elements. Additionally, the application level is connected to the instance level with real constraint examples. The dictionary specifications were established due to the modelling methodology, and the resulting elements of the CC Domain.

Moreover, the CCompEls define the implementation domain that the Process Systems (PS) Domain, which represents multi-level hierarchical activities and models in production systems, is used as a particular implementation of the general approach for building DMMs. In this sense, the integration of the CC Domain and the PS Domain has been explained.

Additionally, the required time and effort for accomplishing repetitive modelling tasks are aimed to be reduced. The support of non-specialists can also be foreseen during the first steps of building DMMs.

More DMMs may be introduced using the developed domain model, such as DMM instances and conceptual model improvements. However, during an extension of the CC Domain, constraints should be considered that are not practical or feasible to introduce.

Additionally, the potential of the developed domains lies for many applications areas. For instance:

(i) The CC Domain connections may support the chosen phase of any decision-making process.

(ii) The work has been conducted considering the complete constraint expressions that are considered consistent. It is important to investigate how the mathematical expressions that construct a constraint are connected to each other and how these connections can be automatically maintained. In addition, mathematical representation formats should represent constraints for storage and management purposes.

(iii) The CC Domain may be extended to assist other methodologies and applications connected to PSE (e.g. multi-objective optimisation, stochastic programming, dynamic programming, constraint programming), another type of system models (systems of systems, interwoven systems, etc.), and industries (e.g. manufacturing, information industry).

The contribution of this chapter can be summarized as follows:

Contribution 4: The DMMs conceptualisation procedure is introduced by modelling elements of the CC Domain.

Contribution 5: The domain integration between the PS Domain and the CC Domain is achieved.

Part IV.

Integrated Exploitation of the Process Systems Domain and the Conceptual Constraint Domain

Chapter 8.

Decision-Making Model Building Sequences and Algorithms

In the previous chapter, an ontology-based modelling approach has enabled the development of the Conceptual Constraint (CC) Domain and the Process Systems (PS) Domain.

In this chapter, two procedures are developed to build new Decision-Making Model (DMM) instances, through the exploitation of the information and relations formalised in the PS Domain and the CC Domain. First, an identification procedure exploits the common conceptual patterns found in the literature of a specific sub-domain (solving Process Systems Engineering (PSE) problems through mathematical programming) to generate a taxonomy with the most frequently used constraints. From this taxonomy, an abstraction of the DMM instances is generated using the CC domain (conceptual DMM) so that the knowledge obtained from such an identification can be transferred to other levels in multi-level hierarchies based on the PS Domain and generating other DMMs for other scenarios. A DMM construction algorithm using these procedures has been developed and applied to case studies. This shows a feasible path leading to the automation of ad-hoc DMM building that has multiple practical applications.

8.1. Introduction

One significant part of the decision-making process is the design phase that is associated with building Decision-Making Model (DMM) alternatives (Simon, 1960): the cognitive strength of domain experts (e.g. Process Systems Engineers) has supported this design phase for years. One of the design phase concerns is the repetitive task of building similar DMMs. In a broad sense, there are several ways to reduce the time and effort required by this repetitive task:

- *Equivalences of different formulations:* mathematical programming is a way to express DMMs; it supports the DMM construction with the set, subset, parameter, variable, and constraint elements as mentioned in Section 7.3 (Cagan et al., 1997). With the support of these elements, it is possible to provide systematic ways to connect different formulations. An earlier method of finding equivalences of different formulations using mathematical programming enabled the construction of stochastic models from their base deterministic linear programming models (Kall and Mayer, 1996). The developed tool provides an interface to choose the type of stochastic linear program model (e.g. single or multi stage chance-constrained) and reshapes the formulation according to user requirements. The management system connects all the information to GAMS and its solvers. Mathematical programming elements are not related to the problem type but can yield a general structural transformation of formulations.

- *General formulations:* another approach to overcome the repetitive work of building similar DMMs is to develop general formulations: these general formulations may apply to many cases of problems that appear to be solved in multi-level hierarchies (from supply chain design to task-unit assignment). For instance, the general formulation of the short-term scheduling model proposed by Kondili et al. (1993) has been applied to many other situations (Laínez et al., 2009; Méndez et al., 2006) - since its concern is the unit-based understanding of the problem and, in many cases, it is irrelevant which type of instances are being managed. However, the use of generic formulations is usually limited to the solution of similar problem structures and within a single decision-making level.

- *Conceptual representation of data:* similarly, it is possible to organise conceptual representation of problem data (e.g. unit operations, state/resource-task-network (STN/RTN), flow sheet modelling) defined by logical sequences (such as the physical, procedural models and recipes). The conceptualisation of problem representation has been accomplished to generalise and abstract the problem instance to a given effectiveness, and as this may be insufficient; in many cases, it is often combined with the general formulations. Conceptualisation is also related to the generation of process model equations; and there have been significant studies related to generating a set of equations from the conceptual representation of data (i.e. MODEL.LA (Stephanopoulos et al., 1990*a*,*b*), BatchDesign-Kit (Linninger, 1995), and TechTool (Linninger et al., 2000)). Moreover, commercial tools have also adopted the generation of equations for process modelling (e.g. gPROMS ModelBuilder (PSE, 2018)). Conceptual representations of problems are not directly connected to DMMs as generic formulations, but they can provide a base for information retrieval regarding parameters.

- *Automated code generation for DMMs:* as well as taking advantage of the equivalences of different formulations (generic formulation development and the conceptual representation of data) there are developments in tools and methodologies for supporting automated procedures. For code generation, the captured knowledge provides process models (Elve and Preisig, 2017) and model templates are used to provide process designs (Fedorova et al., 2015). Additionally, tools may transform process models written in generic languages (e.g. Latex, MathML) to a standard executable flow-sheeting environment (e.g. CAPE-OPEN) (Tolksdorf et al., 2017). Automated tools do not assist in building a path from problem statement (data-input) to DMMs for a broad range of scenarios - but they can provide storage and re-usability for DMMs.

These different approaches to reduce the time and effort required when building DMMs (i.e. equivalences of different formulations, general formulations, a conceptual representation of data, and automated code generation for DMMs) offer advantages in their application areas. For instance, mathematical programming elements can be automatically reorganised to find equivalences for different formulations; general formulations can solve similar problems; conceptual representation of data from problems can provide information retrieval regarding parameters, and automated tools can provide storage and flexible DMMs reusability. This work presents a framework in parallel to these approaches that reduces the time and effort needed for buildin DMMs.

Another aspect of this work is related to the integration of DMMs. The integration concept has to be introduced before DMM integration since the term integration can refer to several concepts. Kelly et al. (2013) define the integration as a process with five different interconnected usages; these interconnected usages of integration are introduced with illustrations and examples from the Process Systems Engineering (PSE) applications as follows:

- *Integrated treatment of issues* relates the issue under consideration to the other social, environmental, and economic objectives to improve decision quality by reducing adverse effects on others. A practical implementation can be found in multi-objective DMMs in PSE (Mele et al., 2011; Medina-González et al., 2017).
- *Integration with stakeholders* links the policies among different stakeholders and the effects resulting from these decisions. The application of this type of integration approach can be categorised under the topic of multi-enterprise decision-making strategies (You and Grossmann, 2009; Hjaila et al., 2017).
- *Integration of disciplines* connects the different points of view with the understanding of the system related to the same issue. A recent implementation is demonstrated

in the integration between analytical and transactional tools for Decision Support Systems (DSSs) (Muñoz, Capón-García, Laínez-Aguirre, Espuña and Puigjaner, 2017).

- *Integration of processes* combines many types of methods from the same perception to make decisions (e.g. the operation of energy storage according to market demands is one of the practical implementations (Silvente et al., 2015)).
- *Integration of scales of consideration* is most likely to aggregate interests. The scale consideration does not mean to keep these physical boundaries; it means to reach an understanding between actors and physical models and the linkage between different model components and the complexity of the problem within the defined area. For instance, the integration of scales can be implemented between the integrated design and planning DMMs (Laínez et al., 2009), scheduling and control DMMs (Harjunkoski et al., 2009; Muñoz et al., 2011; Baldea et al., 2015), and multi-time-scale integration considerations (Biondi et al., 2017).

This chapter aims to provide an integrated environment that includes the conceptualisation of DMMs in the framework of the hierarchical structure of PS. Therefore, the integrated DMMs are aimed at supporting the decisions across the multi-level hierarchies with scale considerations, as well as the multi-disciplinary understanding. Finally, the presented application methodology underlines the DMM integration and uses systematic DMM building procedures that follow the “modelling the modelling procedure” approach to build new DMM instances applicable to new situations.

8.2. Overview of the Framework

The proposed framework supports two phases: (i) conceptualisation of DMMs by breaking them down to their constraints; and (ii) adaptation of DMMs to new problem statements/scenarios (Figure 8.1). This procedure is based on abstraction, classification, and generalisation of DMMs. Therefore, conceptualised DMMs contain physical modelling, domain information, and equation systems (or constraints) and provide the flexibility to be used for the solution of new problems/scenarios.

To sustain conceptualisation and adaptation phases, a framework is needed to support the required connections between functionalities and domain (Figure 8.2). Functionalities include identification, integration, and validation blocks - as well as other possible developments related to the framework. Another main block contains the domains that carry out functionalities: operations research (i.e. connected to DMMs in this work) and PS (i.e. connected to the models in production). These two main blocks (functionalities and

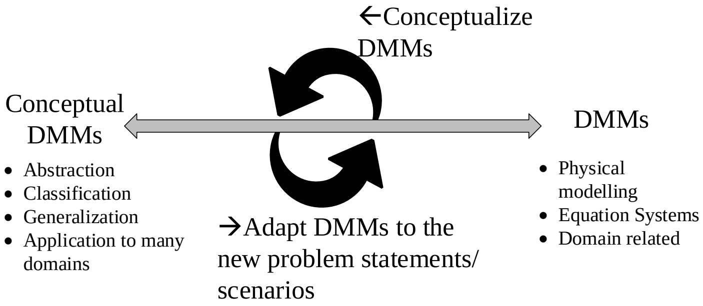


Figure 8.1.: Conceptualization and the adaptation of DMMs

domains) are connected with a dictionary structure that is used for supporting data saving during an identification procedure, or to provide the required information for a validation procedure.

Apart from the framework architectural building blocks, there are two types of human interactions that can be expressed as builder and user. The role of the builder is to introduce DMMs and interact with the constraint identification procedure, as well as the domains and required functions. The user enters problems that contain new problem statements and requests DMM integrations (adaptations) from conceptualised DMMs.

8.2.1. Domains

The work explained in this chapter of the thesis uses two domains: the CC Domain (i.e. the pattern of constraints, required elements to build these patterns, and relations) and the PS Domain (i.e. concepts from the process, physical, state, procedural control models, and their connections). The main characteristics of these two domains are given as follows (the detailed explanation of these domains are in Section 4 and Section 7):

- The PS Domain contains the model information of systems that can be built, designed, and run. The main sources of this domain are the ISA88/95 Standards (ISA, 2010, 2000). The hierarchical model information in the PS Domain supports one of the functions of the DMM integration in the multi-level hierarchies appearing in PS.
- The CC Domain is used to conceptualise DMMs; in this regard, the domain supports the conceptualisation of DMMs using the following elements:
  - The Conceptual Constraints (CCs) express the type of constraints and represent the taxonomy of constraints as built through an “is-a” relation.

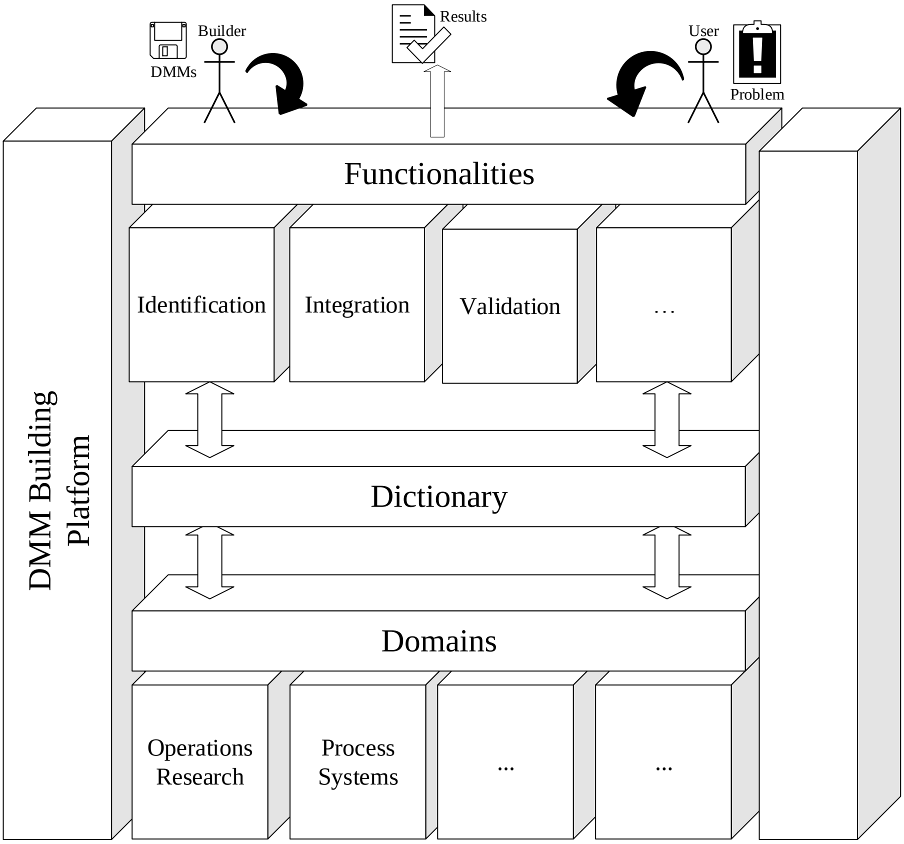


Figure 8.2.: Architecture of the framework with domains and functionalities

- The Conceptual Components (CComps) appear as the main terms in CCs. The "hasComponent" relation connects the CCs and the CComps to express the patterns of each CC.
- The Conceptual Component Elements (CCompEls) are specific concept types of the CC Domain, linked to the CComps that appear in another counterpart domain (in this work, the PS Domain) to carry out the decision-making process.

These CC Domain elements also support domains integration. The CCompEls are the elements that are directly connected to the PS Domain, and the CComps support the value-related integration.

One of the ways to use these domains is to conceptualise DMMs.

8.2.1.1. Problem statement

The mathematical programming representation contains variables, x; objective function(s), $f(x)$; equality constraints, $h(x)$; and inequality constraints, $g(x)$. It can be illustrated as follows (revisiting Section 2.1.1.1):

$$\begin{aligned} \min_{x} \quad & f(x) && && (8.1)\\ \textbf{s.t.} \quad & h(x) && = 0, && (8.2)\\ & g(x) && \leq 0. && (8.3) \end{aligned}$$

The representation given through Expression (8.1) (objective function) and Equations (8.2)-(8.3) (constraints) can be also expressed in another way using the CC Domain elements as follows:

$$\begin{aligned} min_x \quad & (CComp;\ Relation_{k_2};\ CCompEl_{k_3})_{k_1} && (8.4)\\ \textbf{s.t.} \quad & CC[(CComp;\ Relation_{k_6};\ CCompEl_{k_7})_{k_5}; \\ & (\ ;\ Relation_{k_9};\ CCompEl_{k_{10}})_{k_8}]_{k_4}. && (8.5) \end{aligned}$$

where $k_1 = \{1, ..., n\}, ..., k_{10} = \{1, ..., n\}$ are integers that represent the number of elements. Expression (8.4) represents the objective function, which can be expressed as a combination of one or more CComps aiming to be minimised (or maximised), while Expression (8.5) contains a set of CCs that represents the constraints of the DMM.

As an example, a material balance constraint that is constructed for raw materials (from Ierapetritou and Floudas (1998)) is represented using the CC Domain elements as follows:

$$ST_0 + \sum_{i \in I_s} \rho^c_{si} \sum_{j \in J_i} B_{i,j,n} = 0, \forall\, s \in S^R. \tag{8.6}$$

The constraint is presented in Equation 8.6 and its components are expressed in terms of the CC Domain in Table 8.1.

Mathematical Expression	**CComp**
ST_0	StoredAmount
$\sum_{i\in I_s} \rho^c_{si} \sum_{j\in J_i} B_{i,j,n}$	ConsumedAmount

Mathematical Expression	**CCompEl**
$s \in S^R$	StateModelInput

Table 8.1.: CC Domain elements of Equation (8.6)

This material balance constraint is rewritten and reorganized as a DMM[1]:

$$\min_{x} \qquad (Cost;\ ;\ ProcessStateInput) \tag{8.7}$$

$$\begin{aligned} \textbf{s.t.} \quad BalanceCC \quad & [\,(StoredAmount;\ ;\ ProcessStateInput),\\ & (ConsumedAmount;\ ;\ Unit, TimePoint,\\ & UnitProcedure, StateModelInput);\\ & (\ ;\ ;\ StateModelInput)] \end{aligned} \tag{8.8}$$

8.2.2. Functionalities

The functionalities block is the focus of this chapter: the part of the block related to the identification of individual constraints is introduced in Section 8.3; and DMM integration is presented in Section 8.4.

In addition to these two procedures, algorithms, which use details of these procedures, have been developed to be applied to different case studies (see Section 8.5). The details of these algorithms are given in Appendix C.2.

8.3. Constraint Identification Procedure

The constraint identification procedure aims at margining domains and instances using application concepts. The constraint identification procedure contains the CC Domain elements, dictionary support, and user supervision from the builder to produce identified

---

[1] As expected; the DMM expressed previously chooses to consume nothing since the cost would be minimum in this case. A complete DMM also requires at least "SizingCC" and consistent decision structure.

constraints in the CC Domain. This input/output structure is organised using the five steps illustrated in Figure 8.3 (a detailed sequencing diagram including file structures, inputs, and examples is presented in Figure C.1 in Appendix C.1.

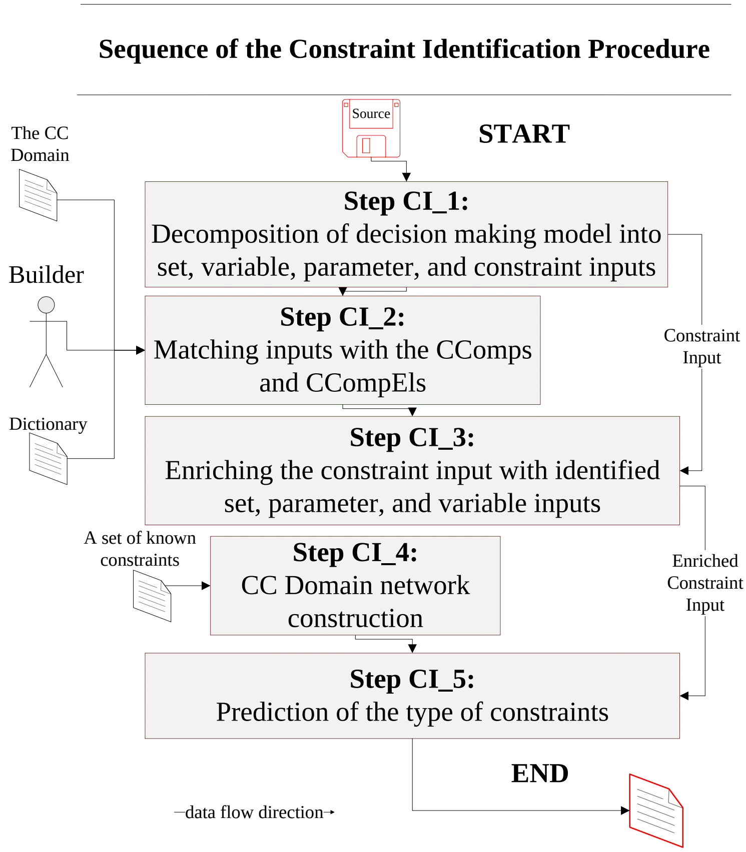


Figure 8.3.: Sequence diagram of the constraint identification procedure

8.3.1. Step CI_1: DMM Decomposition

The goal of the first step is to decompose a DMM described through a mathematical program and implemented following a dedicated language/syntax (e.g. a GAMS file, MathML file, XML file). This step parses a DMM into conceptual mathematical programming elements (i.e. sets, parameters, variables, and constraints - Section 7.3) with some explanations from nomenclature and comments (if they exist in the source file); these mathematical programming elements are the inputs for the rest of the algorithm and are named accordingly: set inputs; parameter inputs; variable inputs; and constraint inputs After this decomposition step, the resulting mathematical programming elements, except the constraint inputs (as shown in Figure 8.3 constraint inputs are directly forwarded to Step CI_3), are sent to the next steps for further identification.

8.3.2. Step CI_2: Matching Step CI_1 outputs with the CC Domain elements

The aim is to match the set, parameter, and variable inputs (the outputs of Step CI_1) to CComps and CCompEls; set inputs are matched with CCompEls, while parameter and variable inputs are matched to the CComps.

The second step has four different inputs:

- Set, parameter, and variable inputs from step CI_1,
- List of CComps and CCompEls from the CC Domain,
- Results from user interaction, and
- Dictionary structure created by previous builders (described next in Section 8.3.2.1).

The matched outputs resulting from this step are used for the enrichment of the constraint inputs and the identification of the type of constraint included in the model (step CI_3).

8.3.2.1. Dictionary

A dictionary stores previous decisions made by the builder(s). These decisions on each entity are saved to support the users through the suggestion of CComps or CCompEls. Each entity in a dictionary includes its structure that has tags and sub-tags (detailed in Dombayci and Espuña (2018*a*)). Each entity has its source name, the CC Domain elements (CC, CComps, CCompEls), source-related information (nomenclature explanation, symbol, main symbol, and entity type), and query-related information (related to the PS and CC Domains integration). The goal is to receive support from the content of the dictionary. If the builders or model developers are the same then they are expected to follow the same pattern for decisions and use similar symbols for similar concepts and nomenclature for the principal elements of DMMs (e.g. “i” for procedures, “j” for units). Additionally, support for user interactions comes from source files nomenclature; suggestions are also based on the similarity between CComps and CCompEls and input explanations (nomenclature) that are extracted from source files. Support can be further developed considering the expressions composed of more than one parameter and variable.

8.3.3. Step CI_3: Constraint Input Enrichment

All inputs (i.e. set inputs, parameter inputs, variable inputs, and constraint inputs) generated from step CI_1 (except for constraint inputs) are matched to the corresponding CC Domain elements. In this step, inputs and their corresponding CComps and CCompEls are linked to their corresponding constraint inputs to sustain the constraint enrichment step. Thus, the enriched constraints are connected to their associated CComps and CCompEls. In this step, constraints are connected to the CC Elements using the DMM constraint definition illustrated in Expression (8.5).

8.3.4. Step CI_4: Network Construction

To predict the CC types associated with constraints, a network of known constraints is created. Such a network can be used to predict the constraint types and assign them probabilities (according to each CC in the network), so that each constraint can be automatically identified as similar to one of the existing CCs.

This step uses a supervised network construction to build a network from a set of known constraints using the Naive Bayes classification technique that is supported by the Machine Learning Toolbox of MATLAB (MathWorks, 2015). Previously, step CI_1, step CI_2 and step CI_3 prepare a set of known constraints and their CC types (e.g. “BalanceCC”, “SizingCC”, “ResourceAllocationCC”). This clustered data (i.e. known constraints) is formed by the DMM references from different purposes and decision levels; this structured information is passed to the network construction step.

Each CC and its connections to the CComps and CCompEls are introduced to the training step of the Naive Bayes classification procedure, which estimates the probability distribution for the CCs according to the input features. A Naive Bayes classifier considers that each of these features (i.e. CComps and CCompEls) contributes independently to the probability of a class (i.e. CC)and this means that features do not necessarily associate with each other.

Aggregated information of the known constraints is used during the network construction step. The network training is executed considering the connections from the subclasses; the training data is introduced as the aggregated relations that CComp patterns are shared with the upper level of a CC. For instance, the “MaterialBalanceCC” CComp pattern is also defined as the “BalanceCC” CComp pattern in the network construction step.

8.3.5. Step CI_5: Prediction of the Type of Constraints

The purpose of this step is to identify the type of an unknown constraint. Therefore, this step requires two data inputs: an enriched constraint input (i.e. the output of step CI_3) and a CC Domain network (i.e. the output of step CI_4). As a result, a set of probability values are produced corresponding to the constraint. The identification procedure returns the classification as a Bayesian probability for each CC according to the actual state of the knowledge (according to the coverage of the network); the sum of similarity percentage values for all CC equals 1.

In addition to the Bayesian probability from the known constraint network, an unknown constraint can also be computed by a similarity percentage related to the use of CComp. Each CC is connected to a set of CComps so that CComps from the use of the CComps by the non-classified constraint can be compared with the use of the CComps in the already classified constraints (i.e. CC). If the CComps of the unknown constraint perfectly match with a candidate CComp set connected to a CC, the CComp similarity percentage is taken as 100%; otherwise there will be a value of between 0% and 100%.

Finally, an overall quantitative measurement can be defined (i.e. combined similarity) considering probability values from Naive Bayes Classification and CComp similarity as follows:

$$\begin{aligned} CombinedSimilarity = \\ BayesianProbability * CCompSimilarityPercentage \end{aligned} \tag{8.9}$$

where the “BayesianProbability” is obtained from the CC Domain network and “CCompSimilarityPercentage” is based on CComps comparison. The Bayesian probability refers to all the CCs with a probability value that decreases when the CC is a subclass of another type CC; however, the CComp similarity percentage may be misleading as a CComp may appear as part of different CCs at the same time. Therefore, the “CombinedSimilarity” gives a better idea of the chances of belonging to a certain class of CCs for the unknown constraint by combining both results.

8.4. DMM Integration Procedure

This section explains the procedure that integrates identified DMMs into new DMM instances using construction algorithms. These construction algorithms utilise previously identified DMMs and create two integration functionalities (i) by extending and (ii) by changing the scope of DMMs. The DMM integration procedure requires identified DMMs (i.e. collections of identified constraints through the identification procedures). To apply the integration functionalities, the procedure is composed of two steps: (i) “DMM composition” and (ii) “DMM update”. Additionally, these steps require the definition of two new elements: a connection matrix identifying the links required to compose the new DMM and calculate functions to generate any eventual missing data. The sequence diagram of the DMM integration procedure is illustrated in Figure 8.4. The general DMM integration procedure includes DMM composition from a set of identified constraints including their input elements (the complete identified set of constraints can be called identified DMM) by introducing the problem input and integration request from the user and connection matrix that is used to create a composed DMM. The created DMM may then be required to be updated with the support of the calculate functions to create DMMs without missing data and calculations.

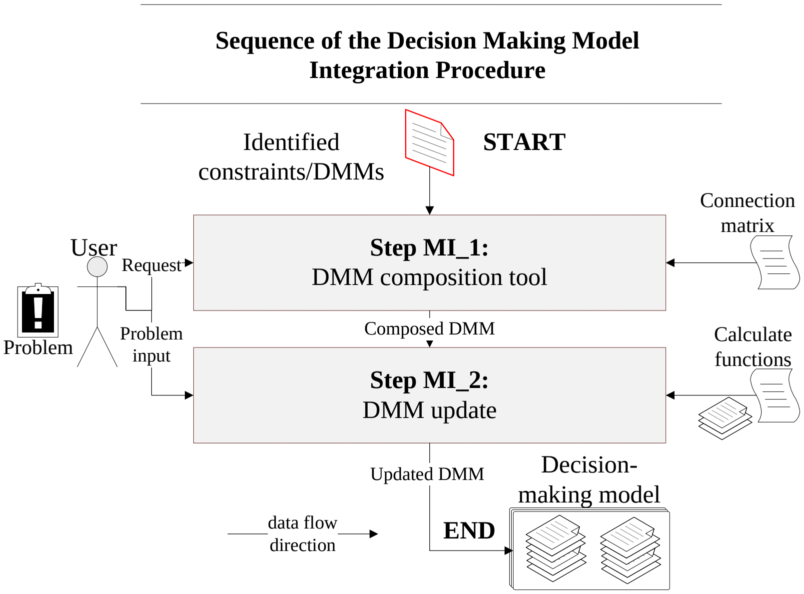


Figure 8.4.: DMM integration sequence diagram

8.4.1. Step MI_1: DMM Composition

This step requires three inputs:

(i) Fully identified DMM (or collections of identified constraints) which is the collection of constraints and all connections to the inputs (if the source file contains a complete DMM in step CI_1),

(ii) The specifications characterising the new DMM instance, and

(iii) A connection matrix according with the new specifications and characteristics of the PS Domain and the CC Domain (described next in Section 8.4.1.1).

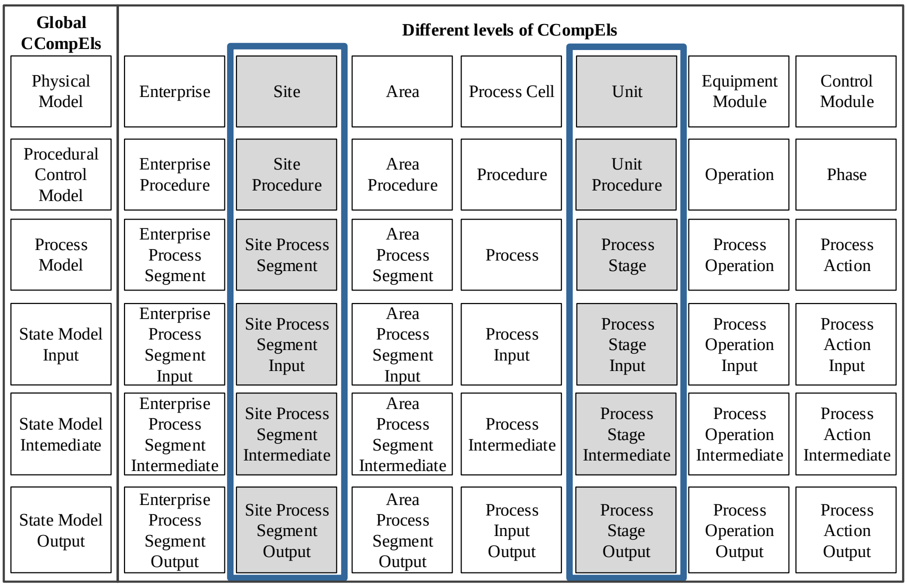
The goal is to compose a new DMM (composed DMM) with the support of a connection matrix. The connection matrix will be developed or selected according to the user purpose (request type and content). The new DMM will be based on the transformations of the relations (constraints) previously conceptualised and characterised (the CC Domain resulting from Section 7). The first step will be to identify the equivalences that can be established through the multi-level hierarchy that will usually appear in the application Domain introduced in Section 4. These equivalences can be represented as a multilayer structure (matrix) of concepts ("connection matrix"). Depending on the problem specifications, the user will determine the request type. The request may have different objects - such as to extend a DMM, change the scope of a DMM, or combine different constraint instances. Depending on the request, the algorithms use the corresponding connection matrices (described next in Section 8.4.1.1) and other information structures. To formalize the concepts and procedures explained in this chapter some algorithms are introduced. Essentially, the algorithms use the same concepts and orders explained in Section sec:ApplicationMethodolgyConstraintIdentificationProcedure and Section 8.4 and introduce a sequential written part. For instance, Algorithm 1 (constraint identification in Section 8.3) aims to identify the CC types of each constraint using the CC Domain network built from Algorithm 2 as a part of the Algorithm 1 (introduced in Section 8.3.4). A DMM extension through the multi-level hierarchies is implemented using Algorithm 3. Then a new DMM instance is built using Algorithm 4.

The output of this step is a new DMM with a set of constraints connected to set and parameter inputs.

8.4.1.1. Connection Matrix

Connection matrices have concepts and relations related to a domain that take advantage of CCompEl links to support the DMM composition step. A connection matrix can be used to model different domains (e.g. multi-level hierarchies, interwoven systems, and systems of systems) and some algorithms are developed to use these connection matrices (this section uses a connection matrix based on multi-level hierarchies). The task of DMM integration algorithms (e.g. Algorithm 3 and Algorithm 4 in case studies 2 and 3 in Section 8.5.2 and Section 8.5.3) is to find CComps and CCompEls links between the identified DMM and the associated connection matrix. These links are then used to produce new DMMs.

Figure 8.5 presents an example of a connection matrix useful for cases where the new DMM may work over the PS Domain, as a counterpart of the CC Domain. This connection matrix presents a multi-level hierarchy; the extension of the relations and concepts supports changing the scope of a DMM to new DMMs. The concepts and relations in the connection matrix are retrieved using a semi-automatic approach during conceptual domain modelling (Chapter 4) from the ISA88 Standard based on the multi-level hierarchical models in PS (ISA, 2010). These concepts and relations are further developed using the ISA95 Standard (ISA, 2000) and the state model addition.



Global CCompEls	**Different levels of CCompEls**						
Physical Model	Enterprise	Site	Area	Process Cell	Unit	Equipment Module	Control Module
Procedural Control Model	Enterprise Procedure	Site Procedure	Area Procedure	Procedure	Unit Procedure	Operation	Phase
Process Model	Enterprise Process Segment	Site Process Segment	Area Process Segment	Process	Process Stage	Process Operation	Process Action
State Model Input	Enterprise Process Segment Input	Site Process Segment Input	Area Process Segment Input	Process Input	Process Stage Input	Process Operation Input	Process Action Input
State Model Intermediate	Enterprise Process Segment Intermediate	Site Process Segment Intermediate	Area Process Segment Intermediate	Process Intermediate	Process Stage Intermediate	Process Operation Intermediate	Process Action Intermediate
State Model Output	Enterprise Process Segment Output	Site Process Segment Output	Area Process Segment Output	Process Input Output	Process Stage Output	Process Operation Output	Process Action Output

Figure 8.5.: Connection matrix compliant with the ISA 88/95 Standards, as example of the the multi-level control/management hierarchy usually found in PS applications

8.4.2. Step MI_2: DMM Update

This step aims to update the composed DMMs to fully adjust to the new situation by adding new calculations to the DMM “calculate function” of the procedure. This step requires three inputs:

(i) A composed DMM,

(ii) Inputs of the problem statement, and

(iii) Calculate functions (described next in Section 8.4.2.1).

The output of the composition step may not be complete; there may be a need for further adjustments to fulfil the requirements of a new situation (e.g. missing information to be collected of calculated, data adjustments, SI-non metric unit calculations). This step is therefore responsible for updating the composed DMMs to completely solve the problem that has arisen from the new situation. Composed DMM requirements and problem inputs that are provided by the user are compared, and calculate functions are used to reclaim the completeness of the composed DMM. The output of the DMM update step is an updated DMM - which can be used to solve a new scenario.

8.4.2.1. Calculate Functions

A set of functions is required to check the availability and consistency of the information currently at the application domain (the PS Domain). At the DMM refining stage, specific instances of such functions will be generated and run in accordence with user request details, in such a way that the availability and consistency of all the required information is validated (and eventually enforced). Therefore, additional functionalities such as retrieving missing information, data adjustments, and calculations are required. The calculate functions are used to align differences using functionalities between composed DMMs from the previous step and problem inputs. Calculate functions are connected to CComps and CCompEls that appear in the CC Domain; they may be composed of a series of queries and calculations. Before the use of the calculate functions, a query is sent to obtain specific data values from the PS Domain -and a calculate function is used if there is an empty or an ambiguous answer to the query.

An example connected to the “Capacity” CComp is illustrated in Figure 8.6 to explain two cases of the calculate function usage. The “Capacity” CComp may be linked to different CCompEls from the procedural control model (e.g. Procedure, UnitProcedure, Operation) as in the CC Domain. Usually, this data, as a parameter input to the DMM, should be retrieved through a query composed from CComps and CCompEls (e.g. “UnitProcedure #capacity”: obtains the capacity values of each unit procedure in the problem input).

Empty query: if the information is unavailable, calculate functions are responsible for providing calculation routines for the missing information. In the example, the “getUnitProcedureCapacity” calculate function is created to calculate the required values (calculate functions may be created by the builder or the user if needed); for example, it is possible to ask for the minimum and maximum capacities of a unit procedure (“getUnitProcedureMinCapacity”, “getUnitProcedureMaxCapacity”) and to use these values to calculate the average capacity.

Ambiguous answer: if more than one type of response for a query is received, some further decisions should be included in the process in addition to calculations. In this ambiguous case, “UnitProcedure #capacity” query returns more than one (and different) values for the same unit procedure. This contradiction may come from capacity differences for unit procedures in different units or areas. In this case, the calculations should be conducted using decisions on the use of quantitative expressions such as maximum, minimum, and average values.

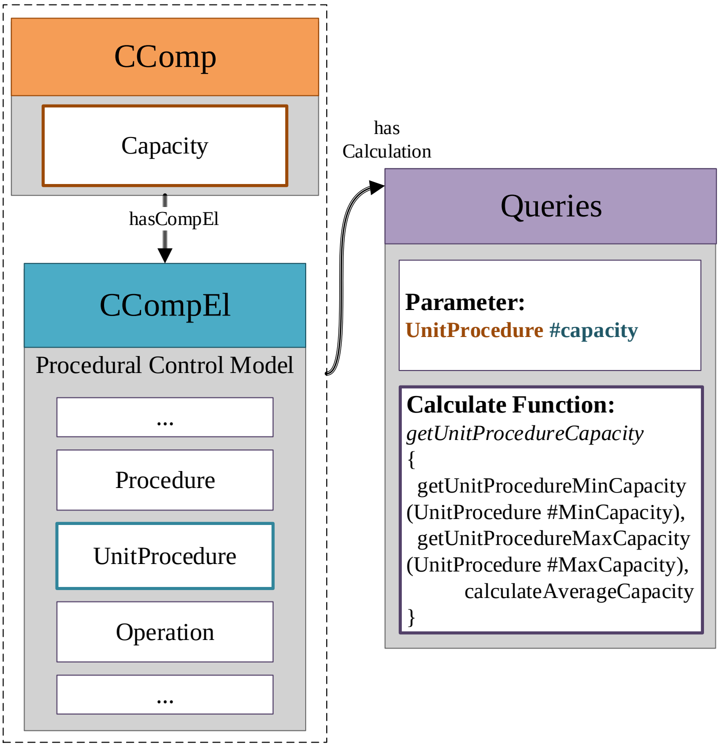


Figure 8.6.: Calculate function demonstration

8.5. Case Studies and Results

Three interconnected case studies are presented to illustrate the application of the proposed procedures and tools by demonstrating their capabilities. The algorithmic path of the application procedures for each case study are presented as an algorithm in Appendix C.2.

The summary of the case studies (i.e. algorithms, input/output relations) are depicted in Figure 8.7 and introduced as follows:

- Case Study 1 in Section 8.5.1 illustrates the identification of a DMM using Algorithm 1 aiming to identify the CC types of each constraint. In this case, Algorithm 2 previously is used to build the CC Domain network.
- The identified DMM is then used to introduce an example of DMM extension through the multi-level hierarchies in Section 8.5.2 as Case Study 2. This extension case uses Algorithm 3.
- Case Study 3 introduces another data input identification procedure and the identified DMM from Case Study 1 is used to build a new DMM instance (see Section 8.5.3) using Algorithm 4.

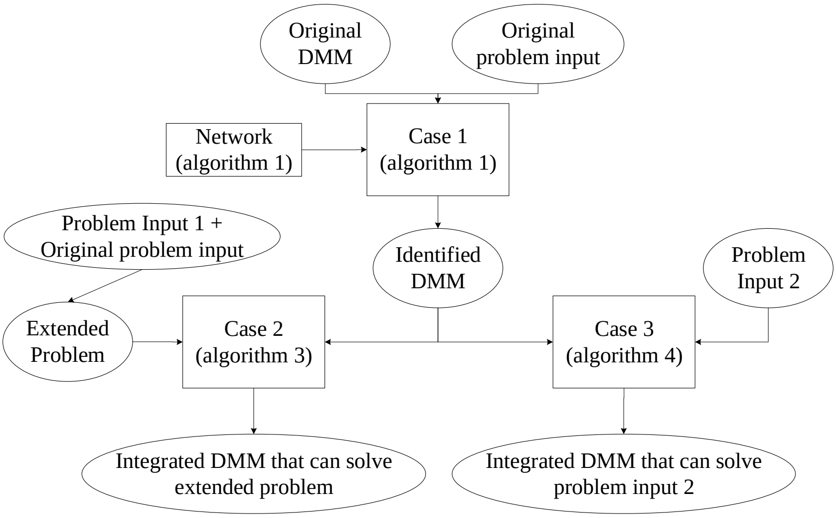


Figure 8.7.: Summary of interconnected case studies

8.5.1. Case Study 1: DMM identification

This case study follows Algorithm 1 to support the builder's activities (expressed in the framework architecture in Figure 8.2). The case study starts with a DMM that is developed for the solution of discrete-time short-term scheduling problems (Kondili et al., 1993; Shah et al., 1993; Maravelias and Grossmann, 2003*a*). All the necessary information from the CC Domain has been gathered (i.e. lists of CComps and CCompEls) and a network has been created based on Algorithm 2. After the first step (step CI_1), the summary of the DMM inputs is given as follows:

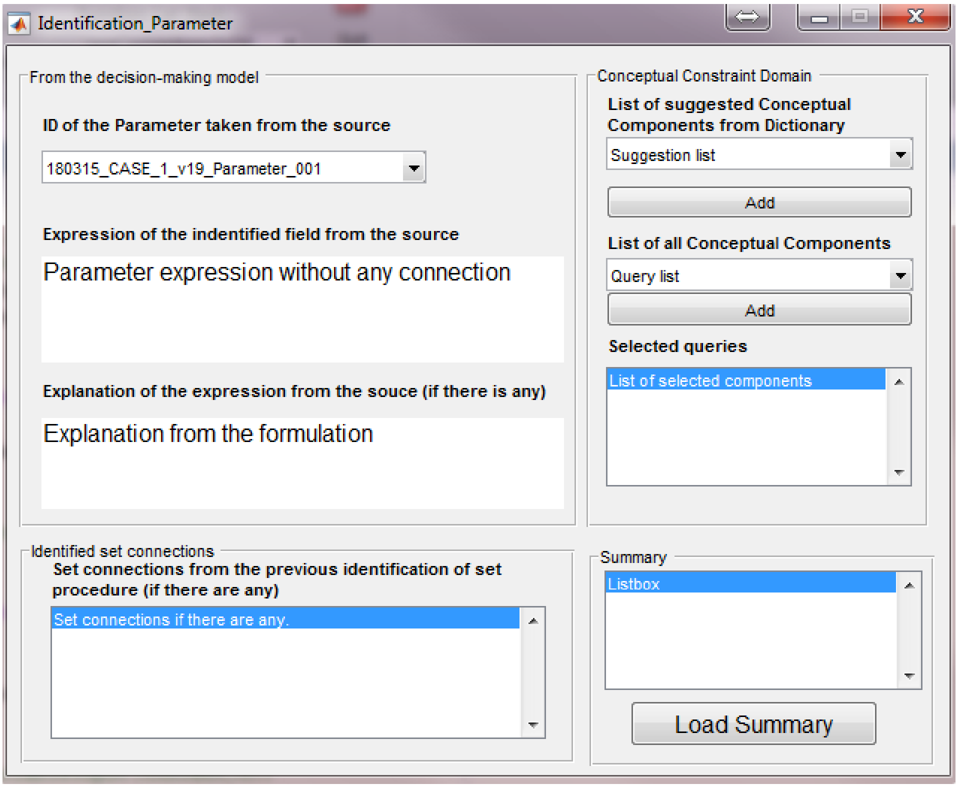
- 6 set and subset inputs,
- 3 variable inputs,
- 14 parameter inputs, and
- 12 constraint inputs (including an objective function).

For the execution of step CI_2, a user interface structure has been developed to match variables, parameters, sets, and subsets to their corresponding CC Domain elements. Figure 8.8 illustrates the user interface developed for the parameter input structure to match with the following areas:

- ID of each parameter taken from the source,
- Expression of the parameter used in the DMM without any set connections,
- Explanation of the expression from the source (if any),
- Identified set connections from the previous identification of set procedure (if any),
- List of suggested CComps from the dictionary,
- List of all CComps,
- Interaction requirement for each parameter (add buttons for the builder),
- Selected CComp, and
- Summary of builder supervision.

Step CI_3 is then implemented; each constraint is connected to CComps and CCompEls. A report is presented in Appendix C.3.

Each constraint is then processed by step CI_5 which produces the information about its CC type.


Identification_Parameter

From the decision-making model

ID of the Parameter taken from the source: 180315_CASE_1_v19_Parameter_001

Expression of the indentified field from the source: Parameter expression without any connection

Explanation of the expression from the souce (if there is any): Explanation from the formulation

Conceptual Constraint Domain

List of suggested Conceptual Components from Dictionary: Suggestion list

List of all Conceptual Components: Query list

Selected queries: List of selected components

Identified set connections

Set connections from the previous identification of set procedure (if there are any): Set connections if there are any.

Summary: Listbox

Figure 8.8.: Matching step support: parameter input example including the dictionary support

8.5.1.1. Results and Discussion of Case Study 1

In Case Study 1, 12 constraints were separately processed by Algorithm 1 based on the constraint identification procedure. The summary of the results of identification of constraints is reported in Table 8.2 and the full classification results are given in Appendix C.4. Some facts and comments considering the result of the case study are summarised as follows:

- The procedure correctly detected 10 out of 12 constraints (except the “Constraint_006”, the “StorageSizingCC”, and “Constraint_008”, the “EconomicCC”) even though the number of known constraints in the CC Domain network is limited (112 constraints). Please note that the correctness of the classification, comparison, and similarity values are introduced in true and false columns in Table 8.2. To overcome this, more constraints should be introduced in the network to obtain accurate predictions considering reported identification results. It is necessary to start testing the system with a variety of constraint types. Moreover, the dictionary should be improved to enhance the efficiency of the steps used by Algorithm 1.

- The combined similarity percentage (see Equation (8.9)) has resulted in the correct classification when this indicator is higher than 6%, leading to correct classifications for “Constraint_001” and “Constraint_0012” which are not precisely classified when just using Naive Bayesian probabilities.
- The Bayesian probability has resulted in the correct classification when the probability values are higher than 0.22[2]. The Naive Bayes classification has not produced accurate predictions when there are not many CComps connected to the CCs; for instance, the “Constraint_006” has only one connected CComp and it has not been predicted as a “SizingCC”; besides the “Constraint_007” has been correctly identified as a “SizingCC”. So there is a need to enrich classifications of constraints with more features (such as bigM formulations, summations symbols, and so on).
- The CComp Similarity has supported the correct classification when the constraints are poorly classified by the Naive Bayesian method. For instance, the comparison of CComps assisted the prediction of the “ResouceAllocationCCs”. However, the classifications obtained using just CComp Similarity as an indicator are globally the worst, including cases where wrong classifications are obtained when a 100% CComp matching is reported by this indicator. It is important to mention that this has been affected because each CComp may appear in more than one type of CC.

---

[2] PS: there are 40 types of CCs and the summation of the probabilities must be equal to 1

Table 8.2.: Prediction results for Case Study 1

	Naive Bayes Classification			**CComp Comparision**			**Combined Similarity**		
Name	**True or False**	**CC Type**	**Probability**	**True or False**	**CC Type**	**Percentage**	**True or False**	**CC Type**	**Percentage**
Constraint_001	0	SizingCC	0,219	1	ResourceAllocationCC	100	1	ResourceAllocationCC	15,696
Constraint_002	1	SizingCC	0,464	0	ResourceAllocationCC	100	1	SizingCC	30,920
Constraint_003	1	SizingCC	0,464	0	ResourceAllocationCC	100	1	SizingCC	30,920
Constraint_004	1	BalanceCC	0,259	1	BalanceCC	25	1	BalanceCC	6,481
Constraint_005	1	BalanceCC	0,259	1	BalanceCC	25	1	BalanceCC	6,481
Constraint_006	0	BalanceCC	0,213	0	BalanceCC	17	0	BalanceCC	3,553
Constraint_007	1	SizingCC	0,175	1	SizingCC	50	1	SizingCC	8,745
Constraint_008	0	BalanceCC	0,211	0	BalanceCC	25	0	BalanceCC	5,280
Constraint_009	1	EconomicCC	0,227	1	EconomicCC	100	1	EconomicCC	22,698
Constraint_010	1	EconomicCC	0,243	1	EconomicCC	50	1	EconomicCC	12,147
Constraint_011	1	BalanceCC	0,333	1	BalanceCC	50	1	BalanceCC	16,648
Constraint_012	0	SizingCC	0,219	1	ResourceAllocationCC	100	1	ResourceAllocationCC	15,696

8.5.2. Case Study 2: Extension of DMMs

This case study is intended to illustrate the proposed DMM extension function and its implementation through Algorithm 3. It is assumed that the user provides an original DMM input (problem input) and aims to have an extended DMM to manage an extended version of the original problem in order to check the feasibility/reliability of the decision made on an extended set of information. In this case study, the main purpose is to solve an extended problem using different recipes, while different product demands (due dates and amounts) must be fulfilled.

The steps in Algorithm 3 are followed to solve this problem.

- The original problem input manages the information from the case study proposed by Kondili et al. (1993) which is defined when the information appears in the "Unit" level. In this case-study, its extension is intended to manage the information at the "Site" level following the ISA 88 Standard.
- The original problem and its extension are modelled as in Figure 8.9. Added echelons introduce new transportation devices and distribution centres. While the information related to the "Production-Site" contains information in the "Unit" level, other added "Transportation-Site" and "Distribution-Site" contain the information related to the "Site" level.
- Additionally, the input of the original and the extended problem are processed by the matching step of the identification procedure (i.e. step CI_2 in Section 8.3.2). Table 8.3 and Table 8.4 illustrate both the original problem and the extended-level problem-matching information.

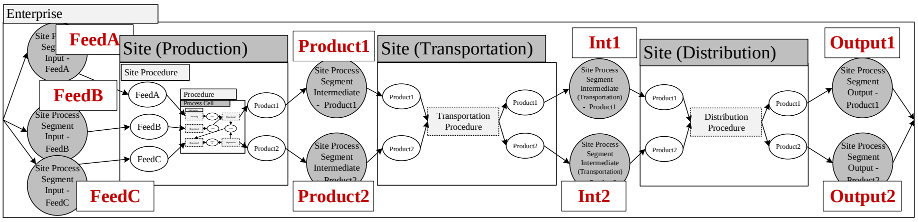


Figure 8.9.: Enterprise model of the case study (In the box-grid the state model is simplified for the clear explanation)

Table 8.3.: Summary of the identified DMM - 1

Symbol	CComp	CCompEl		
		General CCompEl	Identified Level	Extended Level
t		TimeModel	TimePeriod	TimePeriod
j		PhysicalModel	Unit	Site
i		ProceduralModel	UnitProcedure	SiteProcedure
s		StateModel	ProcessStageInput, ProcessStageIntermediate, ProcessStageOutput	SiteProcessSegmentInput, SiteProcessSegmentIntermediate, SiteProcessSegmentOutput
Rm(s)	RawMaterial	StateModel	ProcessStageInput	SiteProcessSegmentInput
FP(s)	Product	StateModel	ProcessStageOutput	SiteProcessSegmentOutput
H	Horizon			
Price(s)	Price	StateModel	ProcessStageOutput	SiteProcessSegmentOutput
Cost(s)	Cost	StateModel	ProcessStageInput	SiteProcessSegmentInput
dem(s)	Demand	StateModel	ProcessStageOutput	SiteProcessSegmentOutput
C(s)	Capacity	StateModel	ProcessStageInput, ProcessStageIntermediate, ProcessStageOutput	SiteProcessSegmentInput, SiteProcessSegmentIntermediate, SiteProcessSegmentOutput
	RawMaterial			
St_O(s)	InitialAmount RawMaterial	StateModel	ProcessStageInput	SiteProcessSegmentInput

Table 8.4.: Summary of the identified DMM - 2

Symbol	CComp	CCompEl		
		General CCompEl	Identified Level	Extended Level
$Phi\_I(i, s)$	InputCoefficient	ProceduralModel; StateModel	UnitProcedure; ProcessStageIntermediate, ProcessStageInput	SIteProcedure; SiteProcessSegmentIntermediate, SiteProcessSegmentInput
$Phi\_O(i, s)$	OutputCoefficient	ProceduralModel; StateModel	UnitProcedure; ProcessStageIntermediate, ProcessStageOutput	SIteProcedure; SiteProcessSegmentIntermediate, SiteProcessSegmentOutput
IJ(i,j)	Capability	PhysicalModel; ProceduralModel	Unit; UnitProcedure	Site; SiteProcedure
SI(i,s)	Capability	ProceduralModel; StateModel	UnitProcedure; ProcessStageIntermediate, ProcessStageInput	SIteProcedure; SiteProcessSegmentIntermediate, SiteProcessSegmentInput
SO(i,s)	Capability	ProceduralModel; StateModel	UnitProcedure; ProcessStageIntermediate, ProcessStageOutput	SIteProcedure; SiteProcessSegmentIntermediate, SiteProcessSegmentOutput
$V_{min(i,j)}$	LowerBoundOfProcedure	ProceduralModel; PhysicalModel	Unit; UnitProcedure	Site; SiteProcedure
$V_{max(i,j)}$	UpperBoundOfProcedure	ProceduralModel; PhysicalModel	Unit; UnitProcedure	Site; SiteProcedure
OperationCost(i,j)	OperationCost	ProceduralModel; PhysicalModel	Unit; UnitProcedure	Site; SiteProcedure
Instalcost(j)	InstallationCost	PhysicalModel	Unit	Site

The data inputs of the original and extended problems have so far been identified within the CC Domain elements; the differences between the original and the extended problem inputs are established. The challenge, in this case, is to control the information appearing both at the "Unit" and at the "Site" levels.

- In the next step, the DMM that can solve the original problem (i.e. identified in the first study[3]) is introduced to the system (e.g. from case study 1 in Section 8.5.1).
- The connection matrix illustrating Section 8.4.1.1 is introduced.
- Step MI_1 in Section 8.4.1 with a set of rules introduced in Algorithm 3 are applied (i.e. line 8 to line 18). A summary of the input files for the solution of the extended problem is introduced in Table 8.5. This table illustrates both the original DMM input and the extended problem input.
- At the end of the algorithm application, the problem is solved using the extended version of the DMM with the extended problem input.

---

[3] Even though another DMM can be identified and Algorithm 3 can be applied, in order to preserve the continuity of the thesis the output of Case Study 1 is chosen to be used to be extended.

Table 8.5.: List of queries

Original Problem Input Data		Integrated Problem Input Data	
1a	PysicalData_Unit	1b	PysicalData_Unit.set PhysicalData_Site
2a	ProcessStageInput, ProcessStageIntermediate, ProcessStageOutput	2b	ProcessStageInput, ProcessStageIntermediate, ProcessStageOutput, SiteProcessSegmentInput, SiteProcessSegmentIntermediate, SiteProcessSegmentOutput
3a	Identification_ProcessStageInput	3b	Identification _SiteProcessSegmentInput
4a	Identification_ProcessStageOutput	4b	Identification _SiteProcessSegmentOutput
5a	UnitProcedure_ STN_input	5b	ProceduralDefinition_STN_input
6a	sellingPrice_ProcessOutput	6b	sellingPrice_ SiteProcessSegmentOutput
6a	sellingPrice_ProcessOutput	6b	sellingPrice_ SiteProcessSegmentOutput
7a	costOfRawMaterials_ProcessInput	7b	costOfRawMaterials _SiteProcessSegmentInput

8.5.2.1. Results and Discussion of Case Study 2

The extension of the DMM is achieved according to the problem description by the user. The output of Case Study 1 (an identified DMM) has been extended and solved according to the extended problem input. A short-term scheduling DMM is extended for the solution of the multi-echelon problem. Additionally, the "Production-Site" is not considered as a single system since each "UnitProcedure" is considered for the solution of the extended DMM. The original problem input and the extended problem input differences directly affect the extension function of a DMM. Finally, Figure 8.10 shows the results obtained at the "extended-level" using the DMM identified at an "original-level" and determining the best usage of the distribution centre and transportation resources, which are also linked to the production site schedule (not in the figure).

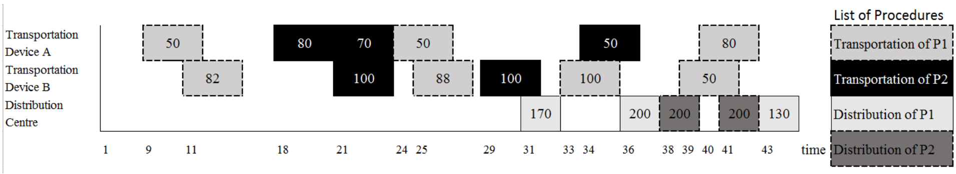


Figure 8.10.: Gantt chart of the planning period (each number in the procedure block represent the amount of material in physical entity and the x-axis presenting the time is not scaled)

The extension request of the user has all available and required information, so the calculate function has not been required in this case study.

8.5.3. Case Study 3: Changing the scope of DMMs

In this case study, the originally identified scheduling DMM based on the early work of Kondili et al. (1993), previously conceptualized (Case Study 1) and extended (Case Study 2), has been used as a departure point to address the supply chain synthesis problem associated with a natural gas (BioSNG) production and distribution network proposed by Calderón et al. (2017). For simplicity, the geographical areas to be considered in this design problem are reduced to seven.

The steps in Algorithm 4 are followed to solve this problem. An initial work related to the identification of the problem input 2 follows. The algorithm continues with the change of scope request; step MI_1 and step MI_2 are then followed to address the DMM integration. The use of a connection matrix and the calculate functions are built according to the new scope (scope changing) request. During this process, the identification of the new problem input data - supported by the procedures identified as CI_2 (Section 8.3.2) is required to connect and proceed with the procedure. In the following subsections, the three required steps associated with the composition of the new DMM are detailed.

8.5.3.1. New DMM composition

This section is straightforward as detailed in Case Study 2. The level of the identified DMM is moved from the "ProcessCell" level to the "Site" level in order to fix the CCompEl of the composed DMM. The connection matrix is the same one given in Figure 8.5.

8.5.3.2. Identification of the problem input data

The identification of the problem input 2 (the new problem input data) is investigated in the three steps related to the ontological model of the PS Domain and corresponding elements in the CC Domain: (i) decisions related to the concepts, which lead to the CCompEls, (ii) relations among these CCompEls, and (iii) available values in the PS Domain and their connections as CComps. The matching step of the identification procedure (i.e. step CI_2) is applied.

CCompEls identification: all the decisions are set at the "Site" level. The procedural model starting from the level associated with the "SiteSegmentProcedure" (SSP) is used, and different SSPs are introduced ("RawMaterial", "Cultivation", "Transportation", "Production", "Distribution", and "GridInjection"). The connection between SPPs and the state model elements are introduced as illustrated in Figure 8.11 ("RawMaterial" does not appear in this figure).

The case study related to the problem (problem input 2) contains 14 sets and subsets (detailed in Table 8.6); these input elements are identified in the PS Domain as six corresponding to CCompEls (as detailed in Table 8.7). These decisions in the PS Domain are summarised as follows:

- The transportation modes (l: truck, trailer, and rail); the set of technologies (k); and their compatibility according to the available feedstock (fk); and other relevant elements are identified in the procedural control model, "SiteProcedure".
- Information related to the physical model: location zones (z) and regions (g) are identified as the "Site" physical model elements.
- Time periods (t) are kept the same - noting that the original case study considers one time-period as five years.

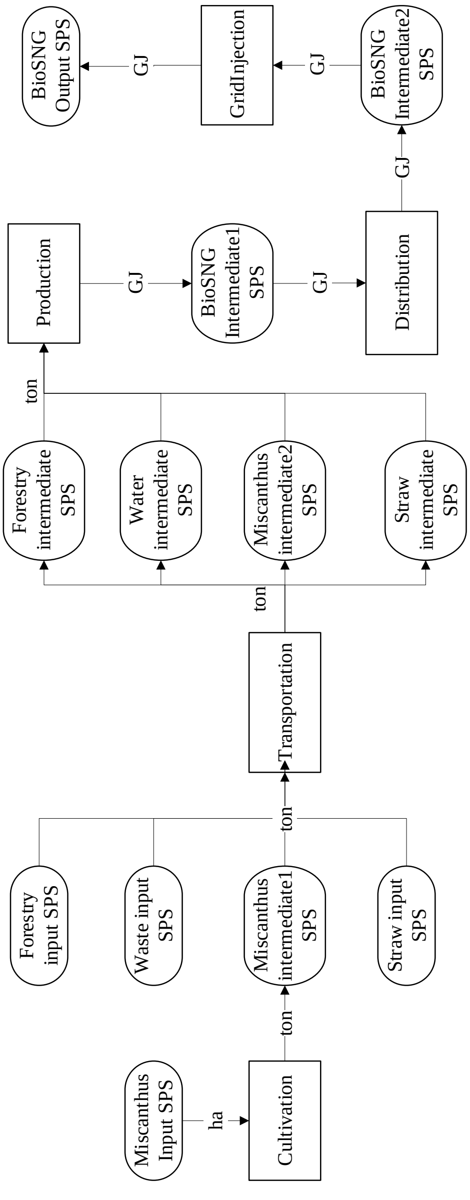


Figure 8.11.: Procedural model of Case Study 3

- All states (fa: forestry, straw, miscanthus, waste and p: BioSNG) are identified with specific state names such as “SiteProcessSegmentInput”, “SiteProcessSegmentIntermediate”, and “SiteProcessSegmentOutput” according to the topological status of the states.

Table 8.6.: Original case study sets explanations

Nomenclature	Symbol
f set of feedstocks F = Fa $\cap$ Fe	f
fa set of available feedstocks	fa
fe set of new energy crops	fe
g set of regions	g
i set of resources (feedstocks and final products)I= F $\cap$ P	i
k set of technologies for integrated facilities k	k
l transportation modes	l
p set of final products	p
s set of segments for cost linearisation	s
t set of time periods	t
z set of local distribution zones (LDZs)	z
fk set of feedstocks f that can be processed by technologies k	fk
gz set of regions g with injection points corresponding to a local distribution zone z	gz
etaiggl set of feasible transport links for each resource i between region g and g_0 via transport mode l	etaiggl

Table 8.7.: Identified sets

Original set	Identified CCompEl
k,l	SiteProcedure
z,g	Site
t	TimePeriod
i,fa,fe,f,p	SiteProcessSegmentInput, SiteProcessSegmentIntermediate, SiteProcessSegmentOutput

Relation identification : after the identification of the sets and the procedural representation of the problem, the resulting CCompEls are connected using four different relations: “hasStateInput”, “hasStateOutput”, “hasAreaFrom”, and “hasAreaTo”. These relations introduce the identity for each CCompEl to identify the structure of the model information.

CComp identification: as the last part for the problem input 2 (new problem input data) identification, CComps are matched with numerical information introduced as parameters. The previously identified SSPs are investigated as follows to determine this numerical information:

- “RawMaterial” $\rightarrow$ “AvailableRawMaterial”, “RawMaterialCost”
- “Cultivation” $\rightarrow$ “CultivationYield”, “AvailableArableLand”, “OperationCost”, “EstablishmentCost”
- “Transportation” $\rightarrow$ “FixedOperationCost”, “VariableOperationCost”

- “Production” → “FixedOperationCost”, “HeatingValue”, “VariableOperationCost”, and “OutputEfficiency” CComps.
- “Distribution” → “VariableOperationCost”, “CapacityOfTransportation”
- “GridInjection” → “Demand”, “Price”, “Incentive”

8.5.3.3. DMM Updates and Calculate Functions Usage

At this stage, there is a list of CComps and CCompEls identified and composed from the original DMM (see Section 8.5.1) and “identified problem input 2”.

As a result of the matching procedure of problem input 2, the composed DMM instance from the composition step can be checked. The DMM is updated using the calculate functions. The aim in this case is to simply compare the CComps and explore the data structures and compose proper calculate functions and queries to retrieve the required information.

- One of the calculate functions to be used in this case study is the example related to the “OperationCost” CComp which is identified with the “ProceduralModel” and the “PhysicalModel” global CCompEls. The extended connections are determined as “Site” and “SiteProcedure” CComps. In the case of the original case study related to the “Production” procedure, the connections of “OperationalCost” CComp are identified as “SiteSegmentInput”, “SiteProcedure” and “TimePeriod”. Since the original case study data input connections are modelled with five relations connected to the “ProceduralModel”, the information required for the new DMM instance information queries are composed according to the procedural data.
- In another situation, the “OutputCoefficient” CComp is identified with the “ProceduralModel” and “StateModel” global CCompEls and the extended connections are identified as “SiteProcedure”, “SiteProcessSegmentIntermediate”, and “SiteProcessSegmentInput” CComps. In the case of the original case study data input information, there is no direct information that can be used as “OutputCoefficient”. Therefore, the information should be calculated using a calculate function to provide it. In this case, the CComps “OutputEfficiency”, “HeatingValue” and “CapacityFactor” are used to provide the “OutputCoefficient” of each procedure that is detected during the identification procedure.
- The demand calculations is also another element to be explored, since the problem input source introduces the demand as a value per year in terms of capacity. However, the integrated DMM requires total demand at the end of the calculation

horizon. These specific calculate functions, which introduce the total demand for energy, introduce a starting point for the DMM revising procedure.

8.5.3.4. Results and discussion of Case Study 3

From the different CCs identified in the CC Domain in the first case study (short-term scheduling problem), and considering the available information from the problem input 2, an integrated DMM is composed and updated for the solution to problem input 2. In summary, while 5% of the total required production is satisfied by the production facilities using residual wastes (in C1), the rest of these facilities are expected to work using the miscanthus raw material (C1-49%, C2-14%, D7-32%). The miscanthus raw material is brought from the cultivated areas of many regions (C1, D1, D4, D6, D7). The total installed production capacity is about 11% less than the capacity proposed at source in the selected area, mainly because of a reduction in the considered areas leads to significant limitations in the material flows from other regions to the currently selected region. This also leads to a significant reduction in the satisfied demand (-28% since grid injection is not allowed from outside the selected areas) and to the use of only waste and miscanthus as raw materials to produce BioSNG. In the same way, since the constructed model does not include transportation costs, some production facilities have not been allocated to the same areas.

Due to the limitations of the departing DMM structure, it has not been possible to include some of the data required to solve this new problem; for instance:

- Data about maximum available total land, establishment cost, rent cost, operation cost, and available arable land data (which can be used for the “Cultivation” procedure) are included in the PS domain, but the associated model to manage this information (the required constraints) is not included in the system.
- Detailed information on all the regions has not been included in the system, so the connections between the excluded regions have not been identified.
- The original DMM does not include transportation costs so to follow the proposed procedure, these terms have not been incorporated into the optimisation problem.
- The cash and depreciation calculations cannot be compared with the original study since the constraints (including tax rates, depreciation for investments) do not appear in the integrated DMM.

In any case, it should also be noted that in addition to the unused data from the originally proposed problem, it is evident that there are many differences between the DMM proposed by Calderón et al. (2017) and the DMM generated using the framework proposed in this thesis; the type of DMM used as departure point for this automatic generation (initially suited to solve a short-term scheduling problem) is completely different and so starting from a supply chain design DMM would probably work better.

8.6. Concluding Remarks

Procedures for constraint identification and Decision-Making Model (DMM) integration, DMM construction algorithms using these procedures, and case studies showing the systematic generation of DMMs have been presented as a management methodology supported by the Process Systems (PS) Domain and the Conceptual Constraint (CC) Domain. Along with the DMM building procedure, the CC Domain and the PS Domain have supported the systematic problem-solving strategies that may appear at any level of PS because of the shareable, extendible, and flexible nature of their model constructions. In summary, the application methodology presented in this chapter on the construction of DMMs (the "modelling of the modelling process") has been developed as a contribution to the Process Systems Engineering (PSE) area.

Previous to the development of the application methodology, a modelling methodology containing the development of both domains was presented as conceptualised DMMs (the CC Domain) and conceptualised physical, procedural control, process, and state models (the PS Domain) (in Chapter 7). In addition, these domains have been semantically and structurally integrated.

This application methodology couples the PS Domain and the CC Domains to provide a range of capabilities using a set of procedures mainly grouped under identification and integration procedures. These procedures have been constructed through sequences that can be used to build different functionalities:

(i) identification of a DMM in the CC Domain to generate generic models as in Case Study 1;

(ii) integration of generic models by extending the definition in the PS Domain as in Case Study 2; and

(iii) (re)building the generic DMM to be used at another level of the multi-level hierarchies as in Case Study 3.

The expected result of the framework (including modelling and application methodologies) for the PSE community is a way to avoid the repetitive task of regenerating specific constraints in addition to the reduction time and effort when constructing the first instance of a DMM. The saving in of time and effort are expected to lead to the development of more accurate and advanced models since the human hour/energy of model builders may be transferred to further develop the details of DMMs.

The developments of the framework are not restricted to the application procedures presented in this chapter. The CC Domain may be connected to (coupled with) other domains,and to assist other methodologies and applications connected to the PS domain (e.g. multi-objective optimisation, stochastic programming, and dynamic programming) or other modelling systems (e.g. systems of systems, interwoven systems). There are potential areas of investigation related to objective functions, performance indicators, stochastic programming, and industrial symbiosis functionalities. Additionally, the functionality of (re)building generic DMMs has the potential to produce other algorithms in order to not just consider constraints as complete entities. The goal is to use the CC Domain elements and build constraints from their components by investigating mathematical expressions. Finally, a software application should be developed to share this framework with the community.

The contribution of this chapter can be summarised as follows:

Contribution 6: The algorithm and required sequences have been developed for the application of the CC and PS Domains for two general functionalities: identification of constraints and integration of DMMs; case studies considering the multi-level functional hierarchies have been solved.

Part V.

Final Remarks

Chapter 9.

Conclusions

This thesis addresses the challenges in systematic Decision-Making Models (DMMs) building, from conceptualisation to application methodologies. The focuses are DMMs that can be used in integrated multi-level management hierarchies, such as those found in industrial Process Systems (PS). Although the application areas of the methodologies are not expected to be restricted, the scope is limited to the Process Systems Engineering (PSE) perspective, which is the focus of this thesis.

The core component required to build a DMM is a conceptual description of the reality in which the DMM is expected to function (i.e. from this conceptual description, systematisation is feasible). To provide a comprehensive conceptual description, two models related to the identified domains of interest have been built: the PS Domain to conceptualise the multi-level structure that appears in the PS Domain and the Conceptual Constraint (CC) Domain for conceptualisation of Operations Research (OR) models. The PS Domain includes the concepts and relations that appear in enterprise/process control standards (i.e. the process model, the physical model, and the procedural control model). This domain supports building, designing, operating, and controlling systems to manufacture products. The CC Domain contains the required concepts and relations that represent and identify DMMs. This domain supports DMM building through its conceptual representation. Overall, the methodologies developed in this thesis support the systematic coordination and operation of activities (using DMMs) in the PS, considering the ways knowledge and decisions are managed throughout the usual hierarchical levels.

An overview of the research topics is presented in Part I. The overview establishes the basis for the PSE perspective on the thesis topics and challenges, considering the field's background (see Chapter 3). Although the main challenges are topics that are under intense study in the PSE community, this thesis presents an unconventional approach to them. The fundamental concepts and methods used throughout the thesis are introduced in the overview. Chapter 1 states the motivation behind the research relating to PSE and the

PSE perspective. The state-of-the-art of current research developments are presented in Chapter 2, including the methods and tools used to develop the thesis and considering three topics: decision-making focused on mathematical programming, PS description focused on management procedures, and conceptual modelling focused on ontological modelling. Finally, Chapter 3 concludes the thesis overview by linking the background and currently available solution strategies with the challenges of this thesis.

Part II of the thesis introduces a methodology for ontology construction and its implementation leading to an integrated management methodology for decision-making in PS.

Chapter 4 focuses on systematic procedures for ontology building. It addresses the construction of domain models that provide a basis for functional contributions in the following chapters, regarding concepts and relations using ontological modelling techniques. The approach includes a semi-automatic procedure for the construction of domain ontologies from normative documents (SECOND Methodology), which combines automated and manual procedures. The result is a methodology can detect concepts in normative documents (i.e. technical documents, standards) rapidly and accurately. The approach also detects inconsistencies that arise during domain model construction and improves the quality of standards by suggesting improvements. Due to this consistency checking, formalisation of the conceptual models and the subsequent writing of normative documents are simultaneously analysed. Moreover, new guidelines are proposed for future normative document writing. The ISA88 Standard has been used as a case study of the ontology construction/learning methodology. As a result, the semantic model of the PS Domain has been created. The outcome of the case study provides the concepts and relations that appear in the PS Domain, containing the process, physical, and procedural control models. Although the ISA88 Standard has been used, additional concepts are added according to the requirements in the following chapters.

The contributions of Chapter 4 can be summarised as follows:

Contribution 1: SECOND Methodology has been developed (see C1 [1]). The ISA88 Standard is used as a case study for the methodology and more comprehensive modelling of PS information within a common and shared domain has been built (see C4).

Contribution 2: Consistency checking to improve normative document sources has been achieved using the SECOND Methodology (see C2).

Chapter 5 presents a new implementation of the PS Domain, in which semantic models are used to communicate DMMs with the problem instances to be solved using the concepts in the PS Domain. Additionally, a modular approach has been presented to encapsulate the data flow from the problem instances to the DMMs. The concepts that appear

---

[1] Explanation of each challenge tag has been introduced in Section 3.1

in the PS Domain have been used to develop connections between a DMM and a series of scenarios. These scenarios are solved to demonstrate the benefits of conceptual modelling in decision-making. Therefore, flexible solution strategies, developed around the base problem, are applied. Alternative physical models, holistic models, and information connections are investigated and different scenarios are implemented in the case study.

Chapter 5, which addresses the usage of the PS Domain in DMM, contains the following contribution:

Contribution 3: Outcome of the SECOND Methodology, the PS Domain, has been used (see C3) to model the PS information within a common and shared domain (see C4). Additionally, this model information has been used to solve multi-level functional hierarchies (see C5).

Part III addresses the decision-making problem through conceptual modelling. Then, a new domain, the CC Domain, is introduced for systematic DMM building procedures. The CC Domain is designed to conceptualise DMMs by classifying constraints, aggregating abstract information from constraints, and generalising the input structure.

Chapter 6 presents an introduction to constraint conceptualisation, which describes the first steps in CC Domain development. A general introduction to the model is introduced in this chapter. Material balance constraints are introduced using the conceptualisation idea.

Chapter 7 introduces the proposed the CC Domain modelling structure. Furthermore, the ontological model resulting from the case study in Chapter 4 is developed and the functions have been maintained along with the CC Domain and the PS Domain integration to sustain primary domain functionalities. A conceptual modelling procedure that preserves the main patterns related to different constraints and complementary dictionary structures is developed. The pattern of each constraint is constructed from the CC Domain and its elements: Conceptual Constraints (CCs), Conceptual Components (CComp), and Conceptual Component Elements (CCompEls). The CC Domain taxonomy presents the CCs; each CC pattern introduces the CComps. In addition, the CCompEls define the implementation domain that the PS Domain uses as a specific implementation of the general approach for building DMMs. The integration methodology of the CC Domain and the PS Domain is explained based on the proposed modelling procedure and the elements to be included in the CC Domain. As a result, the dictionary specifications are established from this procedure. However, the modelling procedure resulted in the DMM specifications that can be used in many applications. These specifications have been presented in the developed dictionary structure. Many other methodologies, models, and software applications may be developed using the CC Domain, and the developed dictionary structure can also be used to develop software intended for model applications related to the PSE area.

The CC Domain modelling contributions in Chapter 7 are given as follows:

Contribution 4: the DMMs conceptualisation procedure is introduced by modelling elements of the CC Domain (see C6).

Contribution 5: domain integration between the PS Domain (see C3) and the CC Domain (see C7) is achieved.

Part IV(Chapter 8) introduces initial applications related to the CC Domain that is supported/demonstrated through the PS Domain. Procedures built through sequences, the DMM construction algorithms, and case studies showing the systematic generation of DMMs have been presented as an application methodology supported by the PS and CC Domains. The previously developed domains (see Chapter 4 and Chapter 7) provide a range of capabilities using a set of procedures that are mainly grouped under identification and integration procedures. These procedures have been constructed through sequences that can be used to build different functionalities:

(i) identification of constraints in the CC Domain to generate generic DMM, and

(ii) integration of generic DMMs by extending or integrating the CC Domain connections in the PS Domain.

The anticipated result of using the proposed framework (including modelling and application methodologies) for the PSE community is that less effort would be required for the repetitive task of regenerating specific constraints. Consequently, less time and effort would be required to construct the first instance of the DMM, leading the development of more accurate advanced models since the human hours and energy of model builders can be transferred to DMM details. Moreover, the proposed system can support non-specialists during the first steps of DMMs building. Analysis of the available information and the patterns identified from DMMs using the CC Domain lead to generic DMMs construction in a wide range of scenarios of hierarchical levels.

The final contribution of this thesis presented in Chapter 8 is as follows:

Contribution 6: algorithms have been developed to apply the CC and PS Domains (see C3-C6-C7) to two general functionalities: identification of constraints and integration of DMMs (see C8). Case studies have been solved considering the multi-level functional hierarchies (see C9).

Despite the encouraging results presented in this thesis, there is room for further improvements. Proposed topics for future research are summarised; in the next section. With these topics, the aim is to improve the approach and overcome current limitations.

Chapter 10.

Topics for Future Research

There are several promising directions for further applications and new developments related to this research line, including those listed below.

The most significant potential for future work seems to lie in expanding the use of the CC Domain by incorporating new algorithms and procedures:

- The applications in this thesis are sustained using mark-up languages such as OWL (web ontology language), XML (extensible markup language), and supporting files (e.g. .txt, .xlsx, .xsd) to transfer and store data and models. In addition, some tools such as Matlab, GAMS, and Java-based ontology management tools (OntoCep, Instance Loader) are used to connect these elements for application procedures, and more importantly to overcome scalability issues. However, it is crucial to build an integrated software environment to share the practical implementations of this thesis. The framework could be included in a complete decision support tool for systematic construction of new DMM instances.
- Apart from the developments aimed at experts, it is essential to discuss the possible developments for the cognitive progress of systems engineers and non-specialists during the first steps of building DMMs. Another way of sharing the knowledge could be to connect the study to existing PSE tools, which use different modelling approaches (e.g. Mosaic, ICAS).
- One practical suggestion is to integrate the CC Domain into formal mathematical markup language (MathML) format to recreate the real mathematical equations without considering the concept-mathematical expression connections. This would mean that reconstruction of the constraints could be linked to the CC Domain Elements. Another important future direction is the special forms of mathematical programming used by modellers (e.g. big M formulation, decomposition algorithms). The goal would be to connect these techniques to the CC Domain with algorithms.

- A database containing model information from purposes and types of DMMs is crucial for further developments. The constraint identification procedure introduced in Chapter 8 will support the identification of DMM elements and the conceptual description of connected elements. The creation of a database considering the CC type of constraints will allow clustering of constraints and movement to different levels within the same constraint using complete and consistent DMMs and by creating consistent and connected constraint packages to be selected according to the data entered into the system.

In addition, the extension and exploitation of the PS domain may lead to many interesting developments:

- It is essential to extend understanding by producing implementations that are applicable to other study areas (e.g. operations research, energy systems) to show that the work is not only applicable to the PS Domain focused on chemical supply chains.
- Further study on the ontology construction research area includes the enrichment of the ontology by extending the current SECOND Methodology to learn more about the currently used normative document. The recommendation for further research is associated with relation clustering to develop the automated methodology. The clustering may require a broader grammatical investigation of verbs that may conclude an improved ontology. Furthermore, the use of other technical documents connected to the domain could be explored, as well as automatic search algorithms from the internet to obtain implicit knowledge.

Regarding the connections between the CC domain and the application domains:

- Another related problem is the use of reasoners to take advantage of description logic studies. Some implicit links between concepts and domains should be built to achieve this connection between domains and reasoners. Links between the framework and the reasoners would facilitate the improvement and/or development of some alternative functionalities in the DMM building framework. Specifically, these reasoners may be used to identify feasible alternatives and support the DMM integration procedure. Alternatively, the reasoners may be used to control the result of the DMMs (considering the new implemented links) and the input/output data consistency. However, it is also essential to consider tailor-made reasoners for further knowledge extraction procedures and applications.

Finally, these suggestions should achieve the ultimate goal: a framework that can be applied to different areas and a wide range of situations to construct DMMs that can tackle different systems and are flexible enough to handle limitations and variety of information systems.

Part VI.

Appendices

Appendix A.

Summary of Scientific Activities Related to the Thesis

Manuscripts under preparation:

- **Dombayci C**, Espuña A. Building decision-making models: the Conceptual Constraint Domain applications.
- **Dombayci C**, Espuña A. Building decision-making models: the Conceptual Constraint Domain modelling.
- **Dombayci C**, Farreres J, Rodríguez H, Espuña A, Graells M. Systematic Construction of Domain Ontologies from Normative Documents.

Published manuscripts:

- **Dombayci C**, Farreres J, Rodríguez H, Espuña A, Graells M. Improving automation standards via semantic modelling: Application to ISA88. ISA Trans. 2017; 67:443-54. Available from: link.
- Moreno-Benito M, **Dombayci C**, Espuña A, Puigjaner L. Integrated Process and Plant Design Optimisation of Industrial Scale Batch Systems: Addressing the Inherent Dynamics through Stochastic and Hybrid Approaches. Chem Eng Trans. 2015; 45:1789-94. Available from: link.

Conference publications:

- **Dombayci C**, Espuña A. Systematic decision making models through Conceptual Constraints. Comput Aided Chem Eng. 2017. p. 1873-8. Available from: link.
- **Dombayci C**, Capón-García E, Muñoz E, Espuña A. Constraint Identification and Integration Procedures in Multi-Level Hierarchical Systems. Comput Aided Chem Eng. 2017. p. 2359-64. Available from: link.

- **Dombayci C**, Espuña A. Modelling Decision Support Systems using Conceptual Constraints - Linking Process Systems Engineering and Decision Making Models. Proc 9th Int Jt Conf Knowl Discov Knowl Eng Knowl Manag. SCITEPRESS - Science and Technology Publications; 2017a. p. 147-154. Available from: link.
- **Dombayci C**, Espuña A. Building Decision Making Models Through Conceptual Constraints: Multi-scale Process Model Implementations. In: Fink A, Fügenschuh A, Geiger M-J, editors. Oper Res Proc 2016. Springer International Publishing; 2018. p. 77-83. Available from: link.
- **Dombayci C**, Medina S, Graells M, Espuña A. Integrated management of hierarchical levels: towards a CAPE tool. In: Kravanja Z, editor. Comput Aided Chem Eng. 2016. p. 7-12. Available from: link.
- Shokry A, **Dombayci C**, Espuña A. Multiparametric Metamodels for Model Predictive Control of Chemical Processes. Comput Aided Process Eng. 2016. p. 937-42. Available from: link.
- **Dombayci C**, Farreres J, Rodriguez H, Muñoz E, Capón-Garcia E, Espuña A, et al. On the Process of Building a Process Systems Engineering Ontology Using a Semi-Automatic Construction Approach. Comput Aided Chem Eng. Elsevier B.V.; 2015. p. 941-946. Available from: link.

Participation in research projects:

- SIGERA Project supported by the Spanish Economy and Competitiveness Ministry and the European Regional Development Fund (DPI2012-37154-C02-01), 2012-2014.
- Centre d'enginyeria de processos, energia i medi ambient project supported by AGAUR (2014-SGR-1092-CEPEiMA), 2014-2016.
- AIMS Project, supported by the Spanish Economy and Competitiveness Ministry and the European Regional Development Fund (DPI2017-580 87435-R), 2017-2019.
- Centre d'enginyeria de processos, energia i medi ambient project supported by AGAUR (2017-SGR-950), 2017-2019.

Research grants:

- FI grant for the recruitment of early-stage research staff supported by AGAUR (Agencia de Gestió d'Ajuts Universitaris i de Recerca), 2014-2017.

Short stay:

- 2016 Summer (2 months), visiting Researcher in Technische Universitat Berlin, Germany
  - Working on specification and implementation of customized instantiation of equation systems in the mathematical modelling and simulation environment MOSAIC.

Conference participation:

- PSE-2015 / ESCAPE-25, 31 May-4June 2015, Copenhagen-Denmark, Process Systems Engineering Conference 2015 / 25th European Symposium on Computer-Aided Process Engineering, link.
- ECCE10, 27 September - 1 October 2015, Nice-France, 10th European Congress of Chemical Engineering along with joint conferences, link.
- DL2015, 7-10 June 2015, Athens-Greece, 28th International Workshop in Description Logics, link.
- ESCAPE-26, 12-15 June 2016, Portoroz-Slovenia, 26th European Symposium on Computer-Aided Process Engineering, link.
- OR2016, 30 August - 2 September 2016, Hamburg-Germany, Annual International Conference of the German Operations Research Society, link.
- ESCAPE-27, 1-5 October 2017, Barcelona-Spain, 27th European Symposium on Computer-Aided Process Engineering along with 10th World Congress of Chemical Engineering, link.
- KEOD 2017, 1-3 November 2017, Madeira-Portugal, 9th International Conference on Knowledge Engineering and Ontology Development, link.

Conference communication:

- **Dombayci C**, Medina S, Graells M, Espuña A. Integrated management of batch production systems using a domain ontology. 10^{th} Eur Congr Chem Eng. 2015.

Book of abstracts:

- **Dombayci C**, Farreres J, Rodríguez H, Muñoz E, Capón-García E, Espuña A, et al. On the Process of Building a Process Systems Engineering Ontology Using a Semi-Automatic Construction Approach. ESCAPE 25, B Abstr. DTU ChemicaEngineering; 2015. p. 107.

Thesis advisory:

- C. De la Roja. Advisors; M. Graells, **C. Dombayci**. Simulation and production planning in a dairy products plant, in Universitat Politecnica de Catalunya; 2016.

Workshop participation:

- SuperPro Designer and SchedulePro, 1-2 June 2017, Ghent-Belgium, Intelligen, Inc. on Process Modelling and Optimization with SuperPro Designer and SchedulePro, link.

Appendix B.

Data for Chapter 7

B.1. Dictionary Specifications

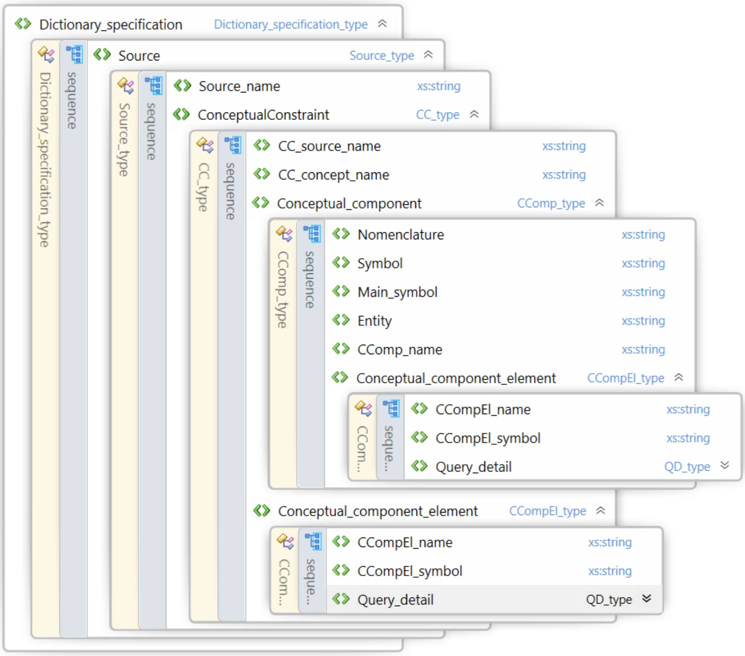


Figure B.1.: Structure of the dictionary specifications

B.1.1. Dictionary Specification in XSD File

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" version="2.1">

  <xs:complexType name="property_type">
    <xs:sequence>
      <xs:element name="name" type="xs:string" />
      <xs:element name="value" type="xs:string" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="QD_type">
    <xs:sequence>
      <xs:element name="className" type="xs:string" />
      <xs:element name="instanceName" type="xs:string" />
      <xs:element name="objectProperty" type="property_type" />
      <xs:element name="dataProperty" type="property_type" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CCompEl_type">
    <xs:sequence>
      <xs:element name="CCompEl_name" type="xs:string" />
      <xs:element name="CCompEl_symbol" type="xs:string" />
      <xs:element name="Query_detail" type="QD_type" />
    </xs:sequence>
  </xs:complexType>

        <xs:complexType name="CComp_type">
                <xs:sequence>
                        <xs:element name="Nomenclature" type="xs:string" />
                        <xs:element name="Symbol" type="xs:string" />
```

```
                      <xs:element name="Main_symbol" type="xs:string"  />
                      <xs:element name="Entity" type="xs:string"  />
                      <xs:element name="CComp_name" type="xs:string" />
                      <xs:element name="Conceptual_component_element" type="CCompEl_type" />
              </xs:sequence>
      </xs:complexType>

<xs:complexType name="CC_type">
  <xs:sequence>
    <xs:element name="CC_source_name" type="xs:string"/>
    <xs:element name="CC_concept_name" type="xs:string" />
    <xs:element name="Conceptual_component" type="CComp_type" />
    <xs:element name="Conceptual_component_element" type="CCompEl_type" />
  </xs:sequence>
</xs:complexType>

      <xs:complexType name="Source_type">
              <xs:sequence>
                      <xs:element name="Source_name" type="xs:string"/>
                      <xs:element name="ConceptualConstraint" type="CC_type"/>
              </xs:sequence>
      </xs:complexType>

<xs:complexType name="Dictionary_specification_type">
  <xs:sequence>
      <xs:element name="Source" type="Source_type"/>
  </xs:sequence>
```

```
  </xs:complexType>

        <xs:element name="Dictionary_specification" type="Dictionary_specification_type" />

        <xs:annotation>
                <xs:documentation xml:lang="en">
                        This schema defines the CC Domain data structure.
                </xs:documentation>
        </xs:annotation>
</xs:schema>
```

B.2. Some References

Table B.1.: References for constraint modelling in the CC Domain

Reference	**Decision-making level**	**Model Feature**
Kondili et al. (1993)	Scheduling	general formulation for batch plants, discrete time representation, implementation of chemical processes
Mota et al. (2015)	Strategic	closed loop supply chain, multi-objective, environmental impact consideration, considering battery producer and distributor
Guillén-Gosálbez et al. (2005)	Strategic	uncertainty consideration, multi-objective, risk management
Harjunkoski et al. (2009)	Operational	Review article
Baldea (2015)	Scheduling and Operational	integration of scheduling and control, model predictive control, implementation of chemical processes
Laínez et al. (2007)	Strategic Design	integrated model considering operational and financial constraints, implementation of chemical processes
Maravelias and Grossmann (2003*b*)	Scheduling	general formulation for multipurpose batch plants, continuous time representation, different storage policies are implemented, chemical processes
Méndez and Cerdá (2002)	Scheduling	general formulation for multipurpose continuous plants, continuous time representation, implementation of chemical processes
Cardoso et al. (2016)	Strategic Design and Planning	closed loop supply chain, multi-objective, risk analysis
Capón-García et al. (2009)	Scheduling	multi-product semi-continuous plants, continuous time representation, implementation of chemical processes
Méndez et al. (2006)	Scheduling	batch processes, review article
Ierapetritou and Floudas (1998)	Scheduling	continuous and semi-continuous processes, continuous time formulation, implementation of chemical processes
Castro et al. (2004)	Scheduling	batch and continuous processes, simple continuous time representation, MILP, resource-task network representation
Silvente et al. (2015)	Scheduling	energy supply and demand planning for energy storage systems
De Meyer et al. (2015)	Strategic and tactical	MILP, biomass based supply chain

Appendix C.

Data for Chapter 8

C.1. Sequence Diagram

This appendix illustrates the detailed sequence diagrams for the constraint identification procedure (see in Section 8.3) in Figure C.1.

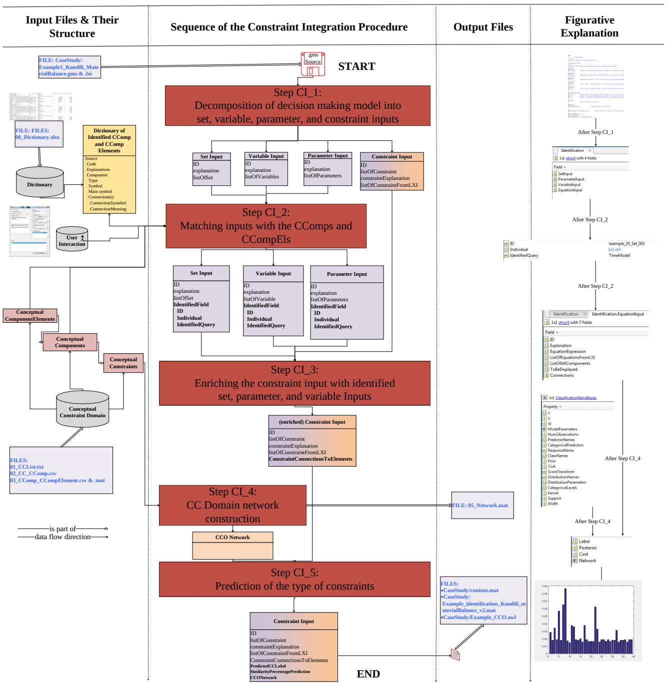


Figure C.1.: Detailed sequence diagram for the constraint identification procedure

C.2. Algorithms

Algorithm 1: Identify Constraints in a DMM (Source: complete DMM)

```
GET: Lists (CComp, CCompEl)
GET: Constraints in a DMM
GET: CC Domain network
DECOMPOSE: DMM into constraints inputs and then each constraint into
  set, variable, and parameter inputs ($StepCI$_1, Section 8.3.1)
for Each constraints ($StepCI$_3, Section 8.3.3) do
    for Each variable and parameter of a constraint $StepCI$_2, Section 8.3.2
      do
        MATCH: CComp
        for Each sets and subsets of parameters and variables do
            MATCH: CCompEl
        end
    end
end
Get Bayesian probabilities and CComp Similarity values $StepCI$_5 (Section
  8.3.5)
OUTPUT: An identified DMM
```

Algorithm 2: Construct the CC Domain Network

```
GET: Lists (CC, CComp, CCompEl)
GET: Training set of constraints
DECOMPOSE: DMM into constraints inputs and then each constraint into
  set, variable, and parameter inputs ($StepCI\_1$ (Section 8.3.1) )
for Each constraints ($StepCI\_3$, Section 8.3.3) do
    MATCH: CC
    for Each variable and parameter of a constraint ($StepCI\_2$, Section
      8.3.2) do
        MATCH CComp
        for Each sets and subsets of parameters and variables do
            MATCH CCompEl
        end
    end
end
TRAIN: Network using output of matching steps ($StepCI\_4$, Section 8.3.4)
OUTPUT CC Domain network
```

Algorithm 3: Extend a DMM

```
GET: Scope extension request from the user
DETERMINE: Requested level in the connection matrix
GET: Problem input from the user
GET: Identified DMM
GET: Constraints of a DMM
GET: Connection Matrix
PRODUCE: Composed DMM ($StepMI\_1$, Section 8.3.1)
for Each CCompEls do
    if CCompEl belongs to the procedural control model then
        Use the extended procedural description of the problem (keep the
          original problem and add procedures belong to the extension)
    else if CCompEl belongs to the physical model then
        Add the additional physical model elements to the current problem
    else if CCompEl belongs to the state model then
        Consider the extended problem input
    else
        Develop another condition
    end
end
OUTPUT 1: Composed DMM

DETERMINE: Missing information according to the problem input and the
  composed DMM (compare Ccomps of the problem input and composed
  DMM)
if There are any missing information ($StepMI\_2$, Section 8.3.2) then
    CHECK: Calculate function
    if The calculate function exist then
        USE: Calculate function
    else
        CREATE and USE: Calculate function
    end
OUTPUT 2: An updated DMM
```

Algorithm 4: Change the scope of a DMM

```
GET: Identified DMM
GET: Constraints of a DMM
GET: Connection Matrix
GET: Scope changing request from the user
DETERMINE: Requested level in the connection matrix
GET: Problem input from the user
PRODUCE: Composed DMM ($StepMI\_1$, Section 8.3.1)
for Each CCompEls do
    GET: CCompEl on the requested level
    INTRODUCE: To the DMM
end
OUTPUT 1: Composed DMM

DETERMINE: Missing information according to the problem input and the
  composed DMM (compare Ccomps of the problem input and composed
  DMM)
if There is any missing information ($StepMI\_2$, Section 8.3.2) then
    CHECK: Calculate function
    if Calculate function exist then
        USE: Calculate function
    else
        CREATE and USE: Calculate function
    end
OUTPUT 2: Updated DMM
```

C.3. Case Study 1 Matching Results

Result of the matching step:

```
Constraint;CASE_1_Constraint_001;;
;CComp (Parameter);DurationOfProceduralModel;
;;CCompEl (Parameter);ProceduralModel
;CComp (Parameter);Allocation;
;;CCompEl (Parameter);PhysicalModel
;;CCompEl (Parameter);ProceduralModel
Constraint;CASE_1_Constraint_002;;
;CComp (Parameter);Allocation;
;;CCompEl (Parameter);PhysicalModel
;;CCompEl (Parameter);ProceduralModel
;CComp (Parameter);;
;CComp (Parameter);UpperOrLowerBoundOFBatchCapacity;
;;CCompEl (Parameter);PhysicalModel
;;CCompEl (Parameter);ProceduralModel
;CComp (Variable);Allocation;
;;CCompEl (Variable);TimeModel
;;CCompEl (Variable);PhysicalModel
;;CCompEl (Variable);ProceduralModel
;CComp (Variable);BatchSize;
;;CCompEl (Variable);TimeModel
;;CCompEl (Variable);PhysicalModel
;;CCompEl (Variable);ProceduralModel
;CComp (Variable);TotalRevenue;
Constraint;CASE_1_Constraint_003;;
;CComp (Parameter);Allocation;
;;CCompEl (Parameter);PhysicalModel
;;CCompEl (Parameter);ProceduralModel
;CComp (Parameter);;
;CComp (Parameter);UpperOrLowerBoundOFBatchCapacity;
;;CCompEl (Parameter);PhysicalModel
;;CCompEl (Parameter);ProceduralModel
;CComp (Variable);Allocation;
;;CCompEl (Variable);TimeModel
;;CCompEl (Variable);PhysicalModel
;;CCompEl (Variable);ProceduralModel
```

```
;CComp (Variable);BatchSize;
;;CCompEl (Variable);TimeModel
;;CCompEl (Variable);PhysicalModel
;;CCompEl (Variable);ProceduralModel
;CComp (Variable);TotalRevenue;
Constraint;CASE_1_Constraint_004;;
;CComp (Parameter);InitialStorage;
;;CCompEl (Parameter);StateModel
;CComp (Variable);CurrentlyAvailableAmount;
;;CCompEl (Variable);TimeModel
;;CCompEl (Variable);StateModel
Constraint;CASE_1_Constraint_005;;
;CComp (Parameter);InitialStorage;
;;CCompEl (Parameter);StateModel
;CComp (Variable);CurrentlyAvailableAmount;
;;CCompEl (Variable);TimeModel
;;CCompEl (Variable);StateModel
Constraint;CASE_1_Constraint_006;;
;CComp (Variable);CurrentlyAvailableAmount;
;;CCompEl (Variable);TimeModel
;;CCompEl (Variable);StateModel
Constraint;CASE_1_Constraint_007;;
;CComp (Parameter);UpperOrLowerBoundOfStorageCapacity;
;;CCompEl (Parameter);StateModel
;CComp (Variable);CurrentlyAvailableAmount;
;;CCompEl (Variable);TimeModel
;;CCompEl (Variable);StateModel
Constraint;CASE_1_Constraint_008;;
;CComp (Parameter);;
Constraint;CASE_1_Constraint_009;;
;CComp (Parameter);Price;
;;CCompEl (Parameter);StateModel
;CComp (Variable);SoldAmount;
;;CCompEl (Variable);TimeModel
;;CCompEl (Variable);StateModel
;CComp (Variable);Cost ;
Constraint;CASE_1_Constraint_0010;;
;CComp (Variable);Cost ;
```

```
;CComp (Variable);Cost ;
;CComp (Variable);TotalRevenue;
Constraint;CASE_1_Constraint_0011;;
;CComp (Parameter);Demand;
;;CCompEl (Parameter);StateModel
;CComp (Variable);SoldAmount;
;;CCompEl (Variable);TimeModel
;;CCompEl (Variable);StateModel
Constraint;CASE_1_Constraint_0012;;
```

C.4. Case Study 1 Naive Bayes Classification Results

```
Constraint_001_CI_report

BayesProbability        BayesProbability_Name
0,218552549    SizingCC
0,156961751    ResourceAllocationCC
0,056588276    BalanceCC
0,053344973    EconomicCC
0,049084992    PhysicalModelAllocationCC
0,045957694    MaterialBalanceCC
0,032269171    SequencingCC
0,031872015    TimingCC
0,029740415    TotalCostEconomicCC
0,023225022    TotalRevenueEconomicCC
0,020889469    SequenceDependentChangeoverCC
0,017070391    UnitAllocationCC
0,016829117    ProcessingCapacityCC
0,016592921    DemandCC
0,016361664    TimesSizingCC
0,014870207    EnergyBalanceCC
0,008535196    CustomerReturnSizingCC
0,008535196    ProductionSizingCC
0,008535196    SaleBalanceCC
0,008535196    SaleSizingCC
0,008535196    StorageCapacitySizingCC
0,008535196    StorageSizingCC
0,008535196    UtilitySizingCC
0,008535196    WorkerAllocationCC
0,008414558    BatchSizingCC
0,008414558    InventoryCostCC
0,008414558    RawMaterialCostCC
0,008414558    TotaSaleCC
0,008414558    TransportSizingCC
0,008414558    TransportationCostCC
0,008296461    MaterialBalanceInitialCC
0,008180832    ManufacturingSiteMaterialBalanceCC
0,008180832    TimeMatchingCC
0,008067604    ResourceBalanceCC
0,008067604    SequenceDependentCleaningChangeoverCC
```

```
0,008067604     SequenceDependentFormatChangeoverCC
0,008067604     StorageBalanceCC
0,008067604     TransferredMaterialBalanceCC
0,008067604     UtilityBalanceCC
0,00795671      NPVCC
```

```
Constraint_002_CI_report

BayesProbability        BayesProbability_Name
0,463805726     SizingCC
0,061199768     ResourceAllocationCC
0,054605973     EconomicCC
0,035487764     TotalRevenueEconomicCC
0,029139258     BatchSizingCC
0,021385418     TotalCostEconomicCC
0,020632693     BalanceCC
0,020368316     PhysicalModelAllocationCC
0,018765264     SequencingCC
0,018193195     TimingCC
0,014569629     ProcessingCapacityCC
0,014063789     DemandCC
0,014063789     MaterialBalanceInitialCC
0,01361372      SequenceDependentChangeoverCC
0,013578877     TimesSizingCC
0,01221698      MaterialBalanceCC
0,010692709     EnergyBalanceCC
0,00754873      CustomerReturnSizingCC
0,00754873      ProductionSizingCC
0,00754873      SaleBalanceCC
0,00754873      SaleSizingCC
0,00754873      StorageCapacitySizingCC
0,00754873      StorageSizingCC
0,00754873      UnitAllocationCC
0,00754873      UtilitySizingCC
0,00754873      WorkerAllocationCC
0,007284814     InventoryCostCC
0,007284814     RawMaterialCostCC
0,007284814     TotaSaleCC
0,007284814     TransportSizingCC
0,007284814     TransportationCostCC
0,006789439     ManufacturingSiteMaterialBalanceCC
0,006789439     TimeMatchingCC
0,006556946     ResourceBalanceCC
0,006556946     SequenceDependentCleaningChangeoverCC
0,006556946     SequenceDependentFormatChangeoverCC
0,006556946     StorageBalanceCC
0,006556946     TransferredMaterialBalanceCC
0,006556946     UtilityBalanceCC
0,006333941     NPVCC
```

```
Constraint_003_CI_report

BayesProbability        BayesProbability_Name
0,463805726     SizingCC
0,061199768     ResourceAllocationCC
0,054605973     EconomicCC
0,035487764     TotalRevenueEconomicCC
0,029139258     BatchSizingCC
```

```
0,021385418     TotalCostEconomicCC
0,020632693     BalanceCC
0,020368316     PhysicalModelAllocationCC
0,018765264     SequencingCC
0,018193195     TimingCC
0,014569629     ProcessingCapacityCC
0,014063789     DemandCC
0,014063789     MaterialBalanceInitialCC
0,01361372      SequenceDependentChangeoverCC
0,013578877     TimesSizingCC
0,01221698      MaterialBalanceCC
0,010692709     EnergyBalanceCC
0,00754873      CustomerReturnSizingCC
0,00754873      ProductionSizingCC
0,00754873      SaleBalanceCC
0,00754873      SaleSizingCC
0,00754873      StorageCapacitySizingCC
0,00754873      StorageSizingCC
0,00754873      UnitAllocationCC
0,00754873      UtilitySizingCC
0,00754873      WorkerAllocationCC
0,007284814     InventoryCostCC
0,007284814     RawMaterialCostCC
0,007284814     TotaSaleCC
0,007284814     TransportSizingCC
0,007284814     TransportationCostCC
0,006789439     ManufacturingSiteMaterialBalanceCC
0,006789439     TimeMatchingCC
0,006556946     ResourceBalanceCC
0,006556946     SequenceDependentCleaningChangeoverCC
0,006556946     SequenceDependentFormatChangeoverCC
0,006556946     StorageBalanceCC
0,006556946     TransferredMaterialBalanceCC
0,006556946     UtilityBalanceCC
0,006333941     NPVCC
```

```
Constraint_004_CI_report

BayesProbability          BayesProbability_Name
0,259238542     BalanceCC
0,083434898     SizingCC
0,061095133     EconomicCC
0,052634602     MaterialBalanceCC
0,044941438     ResourceAllocationCC
0,036957358     SequencingCC
0,036502503     TimingCC
0,034061216     TotalCostEconomicCC
0,028108123     PhysicalModelAllocationCC
0,026599242     TotalRevenueEconomicCC
0,023924371     SequenceDependentChangeoverCC
0,019003604     DemandCC
0,019003604     MaterialBalanceInitialCC
0,018738749     TimesSizingCC
0,017030608     EnergyBalanceCC
0,009775221     CustomerReturnSizingCC
0,009775221     ProductionSizingCC
0,009775221     SaleBalanceCC
0,009775221     SaleSizingCC
```

```
0,009775221    StorageCapacitySizingCC
0,009775221    StorageSizingCC
0,009775221    UnitAllocationCC
0,009775221    UtilitySizingCC
0,009775221    WorkerAllocationCC
0,009637057    BatchSizingCC
0,009637057    InventoryCostCC
0,009637057    ProcessingCapacityCC
0,009637057    RawMaterialCostCC
0,009637057    TotaSaleCC
0,009637057    TransportSizingCC
0,009637057    TransportationCostCC
0,009369374    ManufacturingSiteMaterialBalanceCC
0,009369374    TimeMatchingCC
0,009239696    ResourceBalanceCC
0,009239696    SequenceDependentCleaningChangeoverCC
0,009239696    SequenceDependentFormatChangeoverCC
0,009239696    StorageBalanceCC
0,009239696    TransferredMaterialBalanceCC
0,009239696    UtilityBalanceCC
0,009112691    NPVCC
```

```
Constraint_005_CI_report

BayesProbability        BayesProbability_Name
0,259238542    BalanceCC
0,083434898    SizingCC
0,061095133    EconomicCC
0,052634602    MaterialBalanceCC
0,044941438    ResourceAllocationCC
0,036957358    SequencingCC
0,036502503    TimingCC
0,034061216    TotalCostEconomicCC
0,028108123    PhysicalModelAllocationCC
0,026599242    TotalRevenueEconomicCC
0,023924371    SequenceDependentChangeoverCC
0,019003604    DemandCC
0,019003604    MaterialBalanceInitialCC
0,018738749    TimesSizingCC
0,017030608    EnergyBalanceCC
0,009775221    CustomerReturnSizingCC
0,009775221    ProductionSizingCC
0,009775221    SaleBalanceCC
0,009775221    SaleSizingCC
0,009775221    StorageCapacitySizingCC
0,009775221    StorageSizingCC
0,009775221    UnitAllocationCC
0,009775221    UtilitySizingCC
0,009775221    WorkerAllocationCC
0,009637057    BatchSizingCC
0,009637057    InventoryCostCC
0,009637057    ProcessingCapacityCC
0,009637057    RawMaterialCostCC
0,009637057    TotaSaleCC
0,009637057    TransportSizingCC
0,009637057    TransportationCostCC
0,009369374    ManufacturingSiteMaterialBalanceCC
0,009369374    TimeMatchingCC
```

```
0,009239696      ResourceBalanceCC
0,009239696      SequenceDependentCleaningChangeoverCC
0,009239696      SequenceDependentFormatChangeoverCC
0,009239696      StorageBalanceCC
0,009239696      TransferredMaterialBalanceCC
0,009239696      UtilityBalanceCC
0,009112691      NPVCC
```

```
Constraint_006_CI_report

BayesProbability          BayesProbability_Name
0,213181886      BalanceCC
0,096114344      SizingCC
0,076339623      MaterialBalanceCC
0,071227594      EconomicCC
0,045533622      ResourceAllocationCC
0,041291359      SequencingCC
0,041036474      TimingCC
0,035455514      TotalCostEconomicCC
0,027893324      PhysicalModelAllocationCC
0,027134322      TotalRevenueEconomicCC
0,025733841      SequenceDependentChangeoverCC
0,018726504      DemandCC
0,018595549      TimesSizingCC
0,017727757      EnergyBalanceCC
0,009497013      CustomerReturnSizingCC
0,009497013      ProductionSizingCC
0,009497013      SaleBalanceCC
0,009497013      SaleSizingCC
0,009497013      StorageCapacitySizingCC
0,009497013      StorageSizingCC
0,009497013      UnitAllocationCC
0,009497013      UtilitySizingCC
0,009497013      WorkerAllocationCC
0,009429658      BatchSizingCC
0,009429658      InventoryCostCC
0,009429658      ProcessingCapacityCC
0,009429658      RawMaterialCostCC
0,009429658      TotaSaleCC
0,009429658      TransportSizingCC
0,009429658      TransportationCostCC
0,009363252      MaterialBalanceInitialCC
0,009297775      ManufacturingSiteMaterialBalanceCC
0,009297775      TimeMatchingCC
0,009233207      ResourceBalanceCC
0,009233207      SequenceDependentCleaningChangeoverCC
0,009233207      SequenceDependentFormatChangeoverCC
0,009233207      StorageBalanceCC
0,009233207      TransferredMaterialBalanceCC
0,009233207      UtilityBalanceCC
0,009169529      NPVCC
```

```
Constraint_007_CI_report

BayesProbability          BayesProbability_Name
0,174899604      SizingCC
0,135856576      BalanceCC
0,064035043      EconomicCC
```

```
0,055167389      MaterialBalanceCC
0,047104029      ResourceAllocationCC
0,038735754      SequencingCC
0,038259011      TimingCC
0,035700248      TotalCostEconomicCC
0,029460692      PhysicalModelAllocationCC
0,027879202      TotalRevenueEconomicCC
0,025075616      SequenceDependentChangeoverCC
0,020491214      StorageCapacitySizingCC
0,019918061      DemandCC
0,019640461      TimesSizingCC
0,017850124      EnergyBalanceCC
0,010245607      CustomerReturnSizingCC
0,010245607      ProductionSizingCC
0,010245607      SaleBalanceCC
0,010245607      SaleSizingCC
0,010245607      StorageSizingCC
0,010245607      UnitAllocationCC
0,010245607      UtilitySizingCC
0,010245607      WorkerAllocationCC
0,010100794      BatchSizingCC
0,010100794      InventoryCostCC
0,010100794      ProcessingCapacityCC
0,010100794      RawMaterialCostCC
0,010100794      TotaSaleCC
0,010100794      TransportSizingCC
0,010100794      TransportationCostCC
0,009959031      MaterialBalanceInitialCC
0,009820231      ManufacturingSiteMaterialBalanceCC
0,009820231      TimeMatchingCC
0,009684312      ResourceBalanceCC
0,009684312      SequenceDependentCleaningChangeoverCC
0,009684312      SequenceDependentFormatChangeoverCC
0,009684312      StorageBalanceCC
0,009684312      TransferredMaterialBalanceCC
0,009684312      UtilityBalanceCC
0,009551196      NPVCC
```

```
Constraint_008_CI_report

BayesProbability         BayesProbability_Name
0,211204326      BalanceCC
0,09522275       SizingCC
0,075631466      MaterialBalanceCC
0,070566859      EconomicCC
0,045111234      ResourceAllocationCC
0,040908324      SequencingCC
0,040655804      TimingCC
0,035126614      TotalCostEconomicCC
0,027634574      PhysicalModelAllocationCC
0,026882613      TotalRevenueEconomicCC
0,025495123      SequenceDependentChangeoverCC
0,018552789      DemandCC
0,018552789      MaterialBalanceInitialCC
0,018423049      TimesSizingCC
0,017563307      EnergyBalanceCC
0,009408915      CustomerReturnSizingCC
0,009408915      ProductionSizingCC
```

```
0,009408915      SaleBalanceCC
0,009408915      SaleSizingCC
0,009408915      StorageCapacitySizingCC
0,009408915      StorageSizingCC
0,009408915      UnitAllocationCC
0,009408915      UtilitySizingCC
0,009408915      WorkerAllocationCC
0,009342185      BatchSizingCC
0,009342185      InventoryCostCC
0,009342185      ProcessingCapacityCC
0,009342185      RawMaterialCostCC
0,009342185      TotaSaleCC
0,009342185      TransportSizingCC
0,009342185      TransportationCostCC
0,009211525      ManufacturingSiteMaterialBalanceCC
0,009211525      TimeMatchingCC
0,009147556      ResourceBalanceCC
0,009147556      SequenceDependentCleaningChangeoverCC
0,009147556      SequenceDependentFormatChangeoverCC
0,009147556      StorageBalanceCC
0,009147556      TransferredMaterialBalanceCC
0,009147556      UtilityBalanceCC
0,009084469      NPVCC
```

```
Constraint_009_CI_report

BayesProbability         BayesProbability_Name
0,226980492      EconomicCC
0,08533996       BalanceCC
0,078427949      SizingCC
0,048031408      ResourceAllocationCC
0,039296717      MaterialBalanceCC
0,035818406      SequencingCC
0,035432361      TotalCostEconomicCC
0,035159188      TimingCC
0,030670935      PhysicalModelAllocationCC
0,028234697      TotalRevenueEconomicCC
0,024084633      SequenceDependentChangeoverCC
0,020882325      DemandCC
0,02044729       TimesSizingCC
0,017716181      EnergyBalanceCC
0,010895064      CustomerReturnSizingCC
0,010895064      ProductionSizingCC
0,010895064      SaleBalanceCC
0,010895064      SaleSizingCC
0,010895064      StorageCapacitySizingCC
0,010895064      StorageSizingCC
0,010895064      UnitAllocationCC
0,010895064      UtilitySizingCC
0,010895064      WorkerAllocationCC
0,010664894      BatchSizingCC
0,010664894      InventoryCostCC
0,010664894      ProcessingCapacityCC
0,010664894      RawMaterialCostCC
0,010664894      TotaSaleCC
0,010664894      TransportSizingCC
0,010664894      TransportationCostCC
0,010441162      MaterialBalanceInitialCC
```

```
0,010223645     ManufacturingSiteMaterialBalanceCC
0,010223645     TimeMatchingCC
0,010012128     ResourceBalanceCC
0,010012128     SequenceDependentCleaningChangeoverCC
0,010012128     SequenceDependentFormatChangeoverCC
0,010012128     StorageBalanceCC
0,010012128     TransferredMaterialBalanceCC
0,010012128     UtilityBalanceCC
0,009806406     NPVCC
```

```
Constraint_0010_CI_report

BayesProbability        BayesProbability_Name
0,242934119     EconomicCC
0,082941072     SizingCC
0,064426047     BalanceCC
0,052883618     TotalRevenueEconomicCC
0,052323074     MaterialBalanceCC
0,044675443     ResourceAllocationCC
0,036738619     SequencingCC
0,036286456     TimingCC
0,033859618     TotalCostEconomicCC
0,02794176      PhysicalModelAllocationCC
0,02378277      SequenceDependentChangeoverCC
0,018891128     DemandCC
0,01862784      TimesSizingCC
0,016929809     EnergyBalanceCC
0,009717365     CustomerReturnSizingCC
0,009717365     ProductionSizingCC
0,009717365     SaleBalanceCC
0,009717365     SaleSizingCC
0,009717365     StorageCapacitySizingCC
0,009717365     StorageSizingCC
0,009717365     UnitAllocationCC
0,009717365     UtilitySizingCC
0,009717365     WorkerAllocationCC
0,009580019     BatchSizingCC
0,009580019     InventoryCostCC
0,009580019     ProcessingCapacityCC
0,009580019     RawMaterialCostCC
0,009580019     TotaSaleCC
0,009580019     TransportSizingCC
0,009580019     TransportationCostCC
0,009445564     MaterialBalanceInitialCC
0,00931392      ManufacturingSiteMaterialBalanceCC
0,00931392      TimeMatchingCC
0,009185009     ResourceBalanceCC
0,009185009     SequenceDependentCleaningChangeoverCC
0,009185009     SequenceDependentFormatChangeoverCC
0,009185009     StorageBalanceCC
0,009185009     TransferredMaterialBalanceCC
0,009185009     UtilityBalanceCC
0,009058756     NPVCC
```

```
Constraint_0011_CI_report

BayesProbability        BayesProbability_Name
0,332954745     BalanceCC
```

```
0,071440111       SizingCC
0,05231196        EconomicCC
0,048814811       DemandCC
0,045067734       MaterialBalanceCC
0,038480556       ResourceAllocationCC
0,031644286       SequencingCC
0,031254822       TimingCC
0,029164499       TotalCostEconomicCC
0,024067237       PhysicalModelAllocationCC
0,022775275       TotalRevenueEconomicCC
0,02048495        SequenceDependentChangeoverCC
0,016739827       SaleBalanceCC
0,016044824       TimesSizingCC
0,01458225        EnergyBalanceCC
0,008369914       CustomerReturnSizingCC
0,008369914       ProductionSizingCC
0,008369914       SaleSizingCC
0,008369914       StorageCapacitySizingCC
0,008369914       StorageSizingCC
0,008369914       UnitAllocationCC
0,008369914       UtilitySizingCC
0,008369914       WorkerAllocationCC
0,008251612       BatchSizingCC
0,008251612       InventoryCostCC
0,008251612       ProcessingCapacityCC
0,008251612       RawMaterialCostCC
0,008251612       TotaSaleCC
0,008251612       TransportSizingCC
0,008251612       TransportationCostCC
0,008135802       MaterialBalanceInitialCC
0,008022412       ManufacturingSiteMaterialBalanceCC
0,008022412       TimeMatchingCC
0,007911377       ResourceBalanceCC
0,007911377       SequenceDependentCleaningChangeoverCC
0,007911377       SequenceDependentFormatChangeoverCC
0,007911377       StorageBalanceCC
0,007911377       TransferredMaterialBalanceCC
0,007911377       UtilityBalanceCC
0,007802631       NPVCC
```

```
Constraint_0012_CI_report

BayesProbability          BayesProbability_Name
0,218552549       SizingCC
0,156961751       ResourceAllocationCC
0,056588276       BalanceCC
0,053344973       EconomicCC
0,049084992       PhysicalModelAllocationCC
0,045957694       MaterialBalanceCC
0,032269171       SequencingCC
0,031872015       TimingCC
0,029740415       TotalCostEconomicCC
0,023225022       TotalRevenueEconomicCC
0,020889469       SequenceDependentChangeoverCC
0,017070391       UnitAllocationCC
0,016829117       ProcessingCapacityCC
0,016592921       DemandCC
0,016361664       TimesSizingCC
```

```
0,014870207    EnergyBalanceCC
0,008535196    CustomerReturnSizingCC
0,008535196    ProductionSizingCC
0,008535196    SaleBalanceCC
0,008535196    SaleSizingCC
0,008535196    StorageCapacitySizingCC
0,008535196    StorageSizingCC
0,008535196    UtilitySizingCC
0,008535196    WorkerAllocationCC
0,008414558    BatchSizingCC
0,008414558    InventoryCostCC
0,008414558    RawMaterialCostCC
0,008414558    TotaSaleCC
0,008414558    TransportSizingCC
0,008414558    TransportationCostCC
0,008296461    MaterialBalanceInitialCC
0,008180832    ManufacturingSiteMaterialBalanceCC
0,008180832    TimeMatchingCC
0,008067604    ResourceBalanceCC
0,008067604    SequenceDependentCleaningChangeoverCC
0,008067604    SequenceDependentFormatChangeoverCC
0,008067604    StorageBalanceCC
0,008067604    TransferredMaterialBalanceCC
0,008067604    UtilityBalanceCC
0,00795671     NPVCC
```

Appendix D.

The PS Domain Concepts and Relations

Following list introduces the concepts of the PS Domain.

1 AbstractType
2 AcquireAndExecuteProceduralElement
3 Action
4 Activity
5 Actuator
6 AggregateValue
7 Alarm
8 Algorithm
9 Allocation
10 AllocationOfEquipmentToABatch
11 AllocationSymbol
12 AllowableMaterial
13 AmountOfMaterial
14 Annotation
15 ApplicationSpecificDiscreteEvent
16 Approval
17 ApprovalDate
18 Approved
19 ApprovingEntity
20 Arbitration
21 Area
22 AreaOfTheEnterprise
23 Attribute
24 AutomaticBlockValve
25 Average
26 BadProductPicture
27 BasicControl
28 BasicStageSymbol
29 Batch
30 BatchContainer
31 BatchControl
32 BatchID
33 BatchManufacturingEntreprise
34 BatchManufacturingSite
35 BatchProcess
36 BatchProductionInformation
37 BatchProductionRecord
38 BatchProductionRecordEntry
39 BatchProductionRecordReport
40 BatchProductionRecordSpecification
41 BatchReport
42 BatchSchedule
43 BatchScheduleEntry
44 BatchSize
45 BeginSymbol
46 BillOfResources
47 BOM
48 BuildingBlock
49 BusinessDriver
50 BusinessInformation
51 CalculatedValue
52 CalculatingTheMassFlowRateThroughTheFlowmeter
53 Calibration
54 Campaign
55 Capability
56 CatalyticConversion
57 ChangeManagement
58 Characteristic
59 ChargingIngredients
60 CheckedBy
61 ChemicalSynthesizing
62 Cleaning
63 CleaningVerification
64 CollectBatchAndProcessCellInformation
65 CollectBatchAndUnitInformation
66 CollectData
67 CollectionFrequency
68 CollectionTime
69 Comment
70 CommonData
71 CommonResource
72 Component
73 Condition
74 Constraints
75 ContainerObject
76 Control
77 ControlActivity
78 ControlActivityModel
79 ControlDefinition

80 ControlEquipment
81 ControlFunction
82 Controller
83 ControlModule
84 ControlOfPhase
85 ControlRecipe
86 ControlRecipeEntity
87 ControlRecipeExecution
88 ControlRecipeModel
89 ControlRecipeProcedure
90 ControlStrategy
91 ControlValve
92 Cool
93 Cooling
94 CoordinationControl
95 Copy
96 CorporateRecipe
97 Cost
98 CreationOfAControlRecipe
99 CriticalInformation
100 Crystallization
101 Data
102 Database
103 DatabaseTable
104 DataStore
105 Date
106 DeallocationSymbol
107 DefineGeneralRecipeProceduralElement
108 DefineMasterRecipeProceduralElement
109 Delay
110 Description
111 DiscreteControl
112 Documentation
113 DoneBy
114 Dosing
115 Draft
116 Effective
117 Element
118 ElementDefinition
119 ElementOfBatchProduction
120 ElementOfTheProceduralHierarchy
121 Encapsulation
122 EndOfPhase
123 EndSymbol
124 Energy
125 Enterprise
126 EnterpriseLevelRecipe
127 EnterpriseR&D
128 EnterpriseRoute
129 EnterpriseWideRecipe
130 Entity
131 Entry
132 Enumeration
133 EnumerationSet
134 EnvironmentalProtection
135 Equipment
136 EquipmentAllocation
137 EquipmentAssignment
138 EquipmentControl
139 EquipmentElement
140 EquipmentEntity
141 EquipmentEvent
142 EquipmentHistoryEvent
143 EquipmentIndependentRecipe
144 EquipmentInformation
145 EquipmentMaintenance
146 EquipmentModule
147 EquipmentOperation
148 EquipmentPhase
149 EquipmentProceduralElement
150 EquipmentProcedure
151 EquipmentRequirement
152 EquipmentRequirementElement
153 EquipmentRequirementLibrary
154 EquipmentRestrictions
155 EquipmentStatus
156 EquipmentStep
157 EquipmentUnitProcedure
158 EquipmentUtilization
159 Event
160 EventBasedRule
161 EventInformation
162 EventLog
163 ExceptionHandling
164 ExchangeTable
165 ExclusiveUseResourse
166 ExecuteBasicControl
167 ExecuteEquipmentPhase
168 ExecutionTime
169 ExpirationDate
170 ExtractionOfProductionInformation
171 Factor
172 Fermenting
173 Filename
174 FiltrationStage
175 FormalAbstractRepresentation
176 Formula
177 FormulaObject
178 FormulaValue
179 Function
180 FunctionalGroupOfEquipment
181 GasDensity
182 GasTemperature
183 GeneralizedEquipment
184 GeneralRecipe
185 GeneratedInformation
186 GeographicalGrouping
187 GoodProductPicture
188 GraphicalRepresentation
189 GraphicalSymbol
190 Grind
191 Grinding
192 Grouping
193 GroupingOfEquipment

194 Header
195 Heat
196 Heating
197 HigherLevelObject
198 HistoricalInformation
199 Holding
200 ID
201 Identification
202 IncludingTheProcessDefinition
203 IndividualApproval
204 IndustrySpecificProcessActions
205 Information
206 InProcess
207 Instance
208 Interlocking
209 InternalQualityProgram
210 InventoryStatistics
211 ISA88Definition
212 ISA88Normative
213 Issue
214 IssueDate
215 Labeling
216 LabelingInformation
217 LabTechnicianID
218 LevelBatchScheduleEntry
219 Limitations
220 Limits
221 Link
222 LinkObject
223 LogicalGrouping
224 LogicalModel
225 Lot
226 LowerLevelEntity
227 LowerLevelObject
228 LowerLevelProceduralElement
229 LowerLevelRecipeElement
230 LowerLevelScheduleEntry
231 LowestCostOfUse
232 Maintenance
233 MajorChemicalFunction
234 MajorPhysicalFunction
235 MajorPieceOfProcessingEquipment
236 MajorProcessingAction
237 MakingABatch
238 ManageBatches
239 ManageGeneralRecipe
240 ManageMasterRecipe
241 ManageProcessCellResource
242 ManageSiteRecipe
243 ManageUnitResource
244 Manner
245 Manpower
246 ManufacturingBill
247 MasterRecipe
248 MasterRecipeLevel
249 MasterRecipeParameter
250 MasterRecipeTransformComponent
251 Material
252 MaterialAllocation
253 MaterialAmount
254 MaterialConsumption
255 MaterialConsumptionSummary
256 MaterialData
257 MaterialDefinition
258 MaterialIdentification
259 MaterialInformation
260 MaterialQuality
261 MaterialQualityInformation
262 MaterialsOfConstruction
263 MaterialTransfer
264 Maximum
265 Measure
266 MeasuredVariable
267 Minimum
268 MinimumNumberOfTransfers
269 MinorProcessingActivity
270 Mix
271 Mixing
272 MixingTank
273 Mode
274 Model
275 Module
276 Monitoring
277 MultivariableControl
278 NonproceduralElement
279 Object
280 ObjectModel
281 Occurrence
282 Operation
283 OperationalRequirement
284 Operator
285 OperatorGeneratedInformation
286 OperatorID
287 OperatorSystem
288 OptimizingCriteria
289 OrderedSet
290 Organization
291 Originator
292 OtherInformationRequiredForScheduling
293 Packaging
294 PackagingInformation
295 PackagingRequirement
296 Parameter
297 ParameterObject
298 ParameterValue
299 Part
300 Path
301 PersonalAndEnvironmentalProtection
302 PersonName
303 Personnel
304 PersonnelProtection
305 PersonnelRequirement
306 Phase
307 PhaseInAControlRecipe

308 PhysicalGrouping
309 PhysicalProcessing
310 PieceOfEquipment
311 PieceOfInformation
312 PortionOfInformation
313 PredefinedString
314 PredictableEvent
315 Pressure
316 PressureRequirement
317 ProceduralControl
318 ProceduralDefinition
319 ProceduralElement
320 ProceduralElementReference
321 ProceduralElementWithAssociatedRecipeInformation
322 ProceduralLogic
323 Procedure
324 ProcedureFunctionChart
325 ProcedureInformation
326 ProcedureLogicObject
327 Process
328 ProcessAction
329 ProcessCell
330 ProcessCellEquipmentElement
331 ProcessCellResource
332 ProcessCellSpecifiCInformation
333 ProcessControl
334 ProcessCooling
335 ProcessData
336 ProcessDefinition
337 ProcessElementLibrary
338 ProcessElementLink
339 ProcessElementSpecification
340 ProcessEquipment
341 ProcessingCapability
342 ProcessingStep
343 ProcessInput
344 ProcessManagement
345 ProcessManagementEvent
346 ProcessModel
347 ProcessOperation
348 ProcessOutput
349 ProcessParameter
350 ProcessProcedure
351 ProcessRequirement
352 ProcessStage
353 ProcessVariable
354 ProducedMaterial
355 ProducedProduct
356 ProductDefinition
357 ProductDependentProcessingTime
358 ProductDisposition
359 ProductFamily
360 ProductGrade
361 ProductIdentification
362 ProductionExecutionInformation
363 ProductionHistory
364 ProductionInformation
365 ProductionInformationManagement
366 ProductionInformationRequiredForManufacturing
367 ProductionPerformanceDataStructure
368 ProductionPlanningAndScheduling
369 ProductionRecipeSpecification
370 ProductionReleaseSignoffInformation
371 ProductionRequestObject
372 ProductionResource
373 ProductionResponseObject
374 ProductionRouting
375 ProductionRule
376 ProductionVolume
377 ProductName
378 ProductSpecificInformation
379 ProductSpecificProcessingInformation
380 PropertyStateChanges
381 Quality
382 QualityInformation
383 RawMaterial
384 RawMaterialOption
385 Reaction
386 Reactor
387 Recipe
388 RecipeComponent
389 RecipeCreator
390 RecipeElement
391 RecipeElementObject
392 RecipeEntity
393 RecipeFormula
394 RecipeFormulaValue
395 RecipeHeader
396 RecipeIdentification
397 RecipeInformation
398 RecipeManagement
399 RecipeOperation
400 RecipePhase
401 RecipeProceduralElement
402 RecipeProcedure
403 RecipeUnitProcedure
404 Record
405 RecordingStartTime
406 Rectangle
407 Reference
408 ReferenceToTestDefinitions
409 RegulatingDevice
410 RegulatoryControl
411 ReportHeader
412 Representation
413 RequiredRecipeLevel
414 Requirement
415 ResearchAndDevelopment
416 Resource
417 ResourceDatabase
418 Restriction
419 RuleRule

420 Safety
421 SafetyData
422 SampleBatchProductionRecord
423 SampleTime
424 Schedule
425 ScheduledBatch
426 ScheduledCampaign
427 ScheduleInformation
428 SelectedParameterToTheProcedure
429 Seperation
430 Sequence
431 SequentialControl
432 Set
433 Setpoint
434 SharedResource
435 ShippingMaterial
436 SimilarMaterial
437 SinglePathStructure
438 Site
439 SiteLevelSchedulingActivity
440 SiteRecipe
441 SiteRecipeProcedure
442 SiteSpecificInformation
443 Source
444 SpecialProcessing
445 SpecificTypeOfBatchProductionInformation
446 Spreadsheet
447 Stage
448 StartingStandbyEquipment
449 StartOfPhase
450 StartTime
451 State
452 Status
453 StatusChange
454 Step
455 StepObject
456 StepStart
457 SterilityPeriod
458 Sterilization
459 StopTime
460 Strategy
461 Stream
462 StructuralEntity
463 SuccessRate
464 SupportedDataType
465 SystemGeneratedInformation
466 TagIdentificationObject
467 Technique
468 Temperature
469 Term
470 Test
471 TestStandard
472 TimeBasedRule
473 TimeOfEntryOfResults
474 TimeStamp
475 Train
476 TrainingData
477 TransformComponent
478 Transition
479 TransitionControl
480 TransitionObject
481 Transmitter
482 UniqueIdentifier
483 Unit
484 UnitOfMeasure
485 UnitProcedure
486 UnitRecipe
487 UnitSupervision
488 UnpredictableEvent
489 UnpredictableProcess
490 URL
491 UtilitiesConsumption
492 Value
493 Version
494 VersionNumber
495 VolumetricFlowRateFromAFlowmeter
496 WaterAnalysisResult
497 Weighing
498 Withdrawn

Appendix E.

The CC Domain Elements

Three .csv lists for the CC Domain elements are introduced in the next sections.

E.1. CC-CComp Connections

Here, the CCs and their CComp connections are introduced.

```
Conceptual Constraint ; ConceptualComponent1 ; ConceptualComponent2 ; ConceptualComponent3 ; ConceptualComponent4 ; ConceptualComponent5 ; ConceptualComponent6 ; ConceptualComponent7 ; ConceptualComponent8
BalanceConceptualConstraint ; Input_1 ; Output_1 ; ; ; ; ; ;
BalanceConceptualConstraint ; CurrentlyAvailableAmount_1 ; PreviouslyAvailableAmount_1 ; ConsumedAmount_1 ; ProcessedAmount_3 ; PurchasedAmount_1 ; SoldAmount_1 ; ;
BalanceConceptualConstraint ; SoldMaterial_1 ; Demand_1 ; ; ; ; ; ;
BalanceConceptualConstraint ; CurrentlyAvailableAmount_2 ; PreviouslyAvailableAmount_2 ; ConsumedAmount_2 ; ProcessedAmount_4 ; ChargingEfficiency_1 ; DischargingEfficiency_1 ; ;
BalanceConceptualConstraint ; StoredAmount_1 ; PreviouslyStoredAmount_1 ; InputAmount_1 ; OutputAmount_1 ; ChargingEfficiency_1 ; DischargingEfficiency_1 ; ;
BalanceConceptualConstraint ; CurrentlyAvailableMaterial_1 ; PreviouslyAvailableMaterial_1 ; PurchasedMaterial_1 ; InputProportion_1 ; OutputProportion_1 ; ; ;
BalanceConceptualConstraint ; CurrentlyAvailableMaterial_1 ; PreviouslyAvailableMaterial_1 ; ProducedMaterial_1 ; ConsumedMaterial_1 ; PurchasedMaterial_1 ; SoldMaterial_1 ; ;
BalanceConceptualConstraint ; CurrentlyAvailableMaterial_1 ; PreviouslyAvailableMaterial_1 ; ProcessedAmount_2 ; OutputProportion_1 ; PreviouslyProcessedAmount_1 ; InputProportion_1 ; PurchasedMaterial_1 ; SoldMaterial_1
BalanceConceptualConstraint ; CurrentlyAvailableMaterial_1 ; InitialStorage_1 ; SoldMaterial_1 ; ProducedMaterial_4 ; ; ; ;
BalanceConceptualConstraint ; CurrentlyAvailableMaterial_2 ; PreviouslyAvailableMaterial_2 ; ProducedMaterial_2 ; ConsumedMaterial_2 ; PurchasedMaterial_2 ; SoldMaterial_4 ; ;
BalanceConceptualConstraint ; SuppliedDemand_1 ; Demand_1 ; ; ; ; ; ;
BalanceConceptualConstraint ; CurrentlyAvailableMaterial_4 ; PreviouslyAvailableMaterial_4 ; ProducedMaterial_4 ; ConsumedMaterial_4 ; PurchasedMaterial_4 ; SoldMaterial_6 ; ;
BalanceConceptualConstraint ; CurrentlyAvailableMaterial_5 ; PreviouslyAvailableMaterial_5 ; ProducedMaterial_5 ; ConsumedMaterial_5 ; PurchasedMaterial_5 ; SoldMaterial_2 ; ;
```

```
BalanceConceptualConstraint ; CurrentlyAvailableMaterial_3 ; PreviouslyAvailableMaterial_3 ; ProducedMaterial_3 ; ConsumedMaterial_3 ; PurchasedMaterial_3 ; SoldMaterial_5 ; ;
BalanceConceptualConstraint ; Input_1 ; Output_1 ; ; ; ; ; ;
BalanceConceptualConstraint ; CurrentlyAvailableAmount_1 ; PreviouslyAvailableAmount_1 ; ConsumedAmount_1 ; ProcessedAmount_3 ; PurchasedAmount_1 ; SoldAmount_1 ; ;
DemandCC ; SoldMaterial_1 ; Demand_1 ; ; ; ; ; ;
EnergyBalanceCC ; CurrentlyAvailableAmount_2 ; PreviouslyAvailableAmount_2 ; ConsumedAmount_2 ; ProcessedAmount_4 ; ChargingEfficiency_1 ; DischargingEfficiency_1 ; ;
EnergyBalanceCC ; StoredAmount_1 ; PreviouslyStoredAmount_1 ; InputAmount_1 ; OutputAmount_1 ; ChargingEfficiency_1 ; DischargingEfficiency_1 ; ;
MaterialBalanceCC ; CurrentlyAvailableMaterial_1 ; PreviouslyAvailableMaterial_1 ; PurchasedMaterial_1 ; InputProportion_1 ; OutputProportion_1 ; ; ;
ManufacturingSiteMaterialBalanceConceptualConstraint ; CurrentlyAvailableMaterial_1 ; PreviouslyAvailableMaterial_1 ; PurchasedMaterial_1 ; InputProportion_1 ; OutputProportion_1 ; ; ;
MaterialBalanceCC ; CurrentlyAvailableMaterial_1 ; PreviouslyAvailableMaterial_1 ; ProducedMaterial_1 ; ConsumedMaterial_1 ; PurchasedMaterial_1 ; SoldMaterial_1 ; ;
MaterialBalanceCC ; CurrentlyAvailableMaterial_1 ; PreviouslyAvailableMaterial_1 ; ProcessedAmount_2 ; OutputProportion_1 ; PreviouslyProcessedAmount_1 ; InputProportion_1 ; PurchasedMaterial_1 ; SoldMaterial_1
MaterialBalanceInitialCC ; CurrentlyAvailableMaterial_1 ; InitialStorage_1 ; SoldMaterial_1 ; ProducedMaterial_4 ; ; ; ;
MaterialBalanceCC ; CurrentlyAvailableMaterial_2 ; PreviouslyAvailableMaterial_2 ; ProducedMaterial_2 ; ConsumedMaterial_2 ; PurchasedMaterial_2 ; SoldMaterial_4 ; ;
ResourceBalanceConceptualConstraint ; CurrentlyAvailableMaterial_2 ; PreviouslyAvailableMaterial_2 ; ProducedMaterial_2 ; ConsumedMaterial_2 ; PurchasedMaterial_2 ; SoldMaterial_4 ; ;
DemandCC ; SuppliedDemand_1 ; Demand_1 ; ; ; ; ; ;
SaleBalanceCC ; SuppliedDemand_1 ; Demand_1 ; ; ; ; ; ;
MaterialBalanceCC ; CurrentlyAvailableMaterial_4 ; PreviouslyAvailableMaterial_4 ; ProducedMaterial_4 ; ConsumedMaterial_4 ; PurchasedMaterial_4 ; SoldMaterial_6 ; ;
MaterialBalanceCC ; CurrentlyAvailableMaterial_5 ; PreviouslyAvailableMaterial_5 ; ProducedMaterial_5 ; ConsumedMaterial_5 ; PurchasedMaterial_5 ; SoldMaterial_2 ; ;
MaterialBalanceCC ; CurrentlyAvailableMaterial_3 ; PreviouslyAvailableMaterial_3 ; ProducedMaterial_3 ; ConsumedMaterial_3 ; PurchasedMaterial_3 ; SoldMaterial_5 ; ;
StorageBalanceConceptualConstraint ; CurrentlyAvailableMaterial_4 ; PreviouslyAvailableMaterial_4 ; ProducedMaterial_4 ; ConsumedMaterial_4 ; PurchasedMaterial_4 ; SoldMaterial_6 ; ;
TransferredMaterialBalanceConceptualConstraint ; CurrentlyAvailableMaterial_5 ; PreviouslyAvailableMaterial_5 ; ProducedMaterial_5 ; ConsumedMaterial_5 ; PurchasedMaterial_5 ; SoldMaterial_2 ; ;
UtilityBalanceConceptualConstraint ; CurrentlyAvailableMaterial_3 ; PreviouslyAvailableMaterial_3 ; ProducedMaterial_3 ; ConsumedMaterial_3 ; PurchasedMaterial_3 ; SoldMaterial_5 ; ;
EconomicalConceptualConstraint ; Cost  _1 ; Price_1 ; ; ; ; ; ;
EconomicalConceptualConstraint ; TotalInvestmentCost_1 ; Allocation_6 ; InventeryCost_1 ; ; ; ; ;
EconomicalConceptualConstraint ; SoldMaterial_2 ; PurchasedMaterial_1 ; ProducedMaterial_1 ; CurrentlyAvailableMaterial_5 ; PriceOfStateModelElement_1 ; CostOfStateModel_1 ; CostOfProcedure_1 ;
EconomicalConceptualConstraint ; TotalRawMaterialCost_1 ; Allocation_6 ; RawMaterialCost_1 ; ; ; ; ;
EconomicalConceptualConstraint ; CostOfRevenue_1 ; PurchasedMaterial_1 ; SoldMaterial_3 ; ; ; ; ;
EconomicalConceptualConstraint ; TotalRevenue_1 ; Sale_2 ; SellingPrice_1 ; ; ; ; ;
```

```
EconomicalConceptualConstraint ; PriceOfProduct_1 ; PurchasedMaterial_1 ; SoldMaterial_3
    ; ; ; ; ;
EconomicalConceptualConstraint ; TotalSale_1 ; Sale_2 ; SellingPrice_1 ; ; ; ; ;
EconomicalConceptualConstraint ; TotalTransportationCost_1 ; Allocation_6 ;
    TransportationCost_1 ; ; ; ; ;
TotalCostEconomicalCC ; TotalInvestmentCost_1 ; Allocation_6 ; InventeryCost_1 ; ; ; ; ;
InventoryCostCC ; TotalInvestmentCost_1 ; Allocation_6 ; InventeryCost_1 ; ; ; ; ;
NPVCC; SoldMaterial_2 ; PurchasedMaterial_1 ; ProducedMaterial_1 ;
    CurrentlyAvailableMaterial_5 ; PriceOfStateModelElement_1 ; CostOfStateModel_1 ;
    CostOfProcedure_1 ;
TotalCostEconomicalCC ; TotalRawMaterialCost_1 ; Allocation_6 ; RawMaterialCost_1 ; ; ; ; ;
RawMaterialCostCC ; TotalRawMaterialCost_1 ; Allocation_6 ; RawMaterialCost_1 ; ; ; ; ;
TotalCostEconomicalCC ; CostOfRevenue_1 ; PurchasedMaterial_1 ; SoldMaterial_3 ; ; ; ; ;
TotalRevenueEconomicalCC ; TotalRevenue_1 ; Sale_2 ; SellingPrice_1 ; ; ; ; ;
TotalRevenueEconomicalCC ; PriceOfProduct_1 ; PurchasedMaterial_1 ; SoldMaterial_3 ; ; ; ; ;
TotalRevenueEconomicalCC ; TotalSale_1 ; Sale_2 ; SellingPrice_1 ; ; ; ; ;
TotalCostEconomicalCC ; TotalTransportationCost_1 ; Allocation_6 ; TransportationCost_1
    ; ; ; ; ;
TotaSaleConceptualCC ; TotalSale_1 ; Sale_2 ; SellingPrice_1 ; ; ; ; ;
TransportationCostCC ; TotalTransportationCost_1 ; Allocation_6 ; TransportationCost_1 ; ; ; ; ;
ResourceAllocationConceptualConstraint ; Allocation_2 ; ; ; ; ; ; ;
PhysicalModelAllocationCC ; Allocation_2 ; ; ; ; ; ; ;
ResourceAllocationConceptualConstraint ; Allocation_1 ; ; ; ; ; ; ;
PhysicalModelAllocationCC ; Allocation_3 ; DurationOfProceduralModel_2 ; ; ; ; ; ;
UnitAllocationCC ; Allocation_3 ; DurationOfProceduralModel_2 ; ; ; ; ; ;
ResourceAllocationConceptualConstraint ; Allocation_4 ; DurationOfProceduralModel_1 ; ; ; ; ; ;
WorkerAllocationCC ; Allocation_4 ; DurationOfProceduralModel_1 ; ; ; ; ; ;
SequencingConceptualConstraint ; ProcedureStartTime_2 ; ProcedureFinishTime_2 ;
    ChangeOverTime_2 ; SequencingRequirement_1 ; Horizon_1 ; ; ;
SequencingConceptualConstraint ; ProcedureStartTime_2 ; ProcedureFinishTime_2 ;
    CleaningTime_1 ; SequencingRequirement_1 ; Horizon_1 ; ; ;
SequencingConceptualConstraint ; ProcedureStartTime_2 ; ProcedureFinishTime_2 ;
    FormatChangeTime_1 ; SequencingRequirement_1 ; Horizon_1 ; ; ;
SequenceDependentChangeoverCC ; ProcedureStartTime_2 ; ProcedureFinishTime_2 ;
    ChangeOverTime_2 ; SequencingRequirement_1 ; Horizon_1 ; ; ;
SequenceDependentChangeoverCC ; ProcedureStartTime_2 ; ProcedureFinishTime_2 ;
    CleaningTime_1 ; SequencingRequirement_1 ; Horizon_1 ; ; ;
SequenceDependentChangeoverCC ; ProcedureStartTime_2 ; ProcedureFinishTime_2 ;
    FormatChangeTime_1 ; SequencingRequirement_1 ; Horizon_1 ; ; ;
SequenceDependentCleaningChangeoverCC ; ProcedureStartTime_2 ; ProcedureFinishTime_2 ;
    CleaningTime_1 ; SequencingRequirement_1 ; Horizon_1 ; ; ;
SequenceDependentFormatChangeoverCC ; ProcedureStartTime_2 ; ProcedureFinishTime_2 ;
    FormatChangeTime_1 ; SequencingRequirement_1 ; Horizon_1 ; ; ;
SequencingConceptualConstraint ; ProcedureStartTime_3 ; ProcedureFinishTime_3 ;
    SequencingAlllocation_1 ; ; ; ; ;
SequencingConceptualConstraint ; Allocation_5 ; SequencingRequirement_1 ; Horizon_1 ; ; ; ; ;
SizingConceptualConstraint ; UpperOrLowerBoundOFBatchCapacity_1 ; BatchSize_1 ;
    Allocation_7 ; ; ; ; ;
SizingConceptualConstraint ; UpperOrLowerBoundAmountOfReturn_1 ; SuppliedDemand_2 ; ; ; ; ; ;
SizingConceptualConstraint ; UpperOrLowerBoundOfProcessingCapacity_1 ; ProcessedAmount_1 ;
    Allocation_1 ; ; ; ; ;
SizingConceptualConstraint ; UpperOrLowerBoundProductionCapacity_1 ; ProductionAmount_1
    ; ; ; ; ; ;
SizingConceptualConstraint ; UpperOrLowerBoundSale_1 ; Sale_1 ; ; ; ; ; ;
BatchSizingCC ; UpperOrLowerBoundOFBatchCapacity_1 ; BatchSize_1 ; Allocation_7 ; ; ; ; ;
CustomerReturnSizingCC ; UpperOrLowerBoundAmountOfReturn_1 ; SuppliedDemand_2 ; ; ; ; ; ;
```

```
ProcessingCapacityCC ; UpperOrLowerBoundOfProcessingCapacity_1 ; ProcessedAmount_1 ; Allocation_1 ; ; ; ; ;
ProductionSizingCC ; UpperOrLowerBoundProductionCapacity_1 ; ProductionAmount_1 ; ; ; ; ; ;
SaleSizingCC ; UpperOrLowerBoundSale_1 ; Sale_1 ; ; ; ; ; ;
SizingConceptualConstraint ; UpperOrLowerBound_1 ; Variable_1 ; ; ; ; ; ;
SizingConceptualConstraint ; UpperOrLowerBound_2 ; Variable_2 ; Allocation_1 ; ; ; ; ;
SizingConceptualConstraint ; UpperOrLowerBoundOfStorageCapacity_1 ; StoredAmount_2 ; ; ; ; ; ;
SizingConceptualConstraint ; UpperOrLowerBoundStockLevel_1 ; StoredAmount_2 ; ; ; ; ; ;
SizingConceptualConstraint ; UpperOrLowerBoundTransportSize_1 ; TransferredProduct_1 ; ; ; ; ; ;
SizingConceptualConstraint ; UpperOrLowerBoundNumberOfTrip_1 ; TransferredProduct_1 ; NumberOfTrip_1 ; ; ; ; ;
SizingConceptualConstraint ; UpperOrLowerBoundUtility_1 ; UsedUtilityAmount_1 ; ; ; ; ; ;
StorageCapacitySizingCC ; UpperOrLowerBoundOfStorageCapacity_1 ; StoredAmount_2 ; ; ; ; ; ;
StorageSizingCC ; UpperOrLowerBoundStockLevel_1 ; StoredAmount_2 ; ; ; ; ; ;
TimesSizingCC ; UpperOrLowerBoundTransportSize_1 ; TransferredProduct_1 ; ; ; ; ; ;
TimesSizingCC ; UpperOrLowerBoundNumberOfTrip_1 ; TransferredProduct_1 ; NumberOfTrip_1 ; ; ; ; ;
TransportSizingCC ; UpperOrLowerBoundNumberOfTrip_1 ; TransferredProduct_1 ; NumberOfTrip_1 ; ; ; ; ;
UtilitySizingCC ; UpperOrLowerBoundUtility_1 ; UsedUtilityAmount_1 ; ; ; ; ; ;
TimeMatchingConceptualConstraint ; ProducedAmountCompletionTime_1 ; ConsumedAmountCompletionTime_1 ; LengthOfProcedure_1 ; Allocation_8 ; Horizon_1 ; ; ;
TimingConceptaulConstraint ; ProducedAmountCompletionTime_1 ; ConsumedAmountCompletionTime_1 ; LengthOfProcedure_1 ; Allocation_8 ; Horizon_1 ; ; ;
TimingConceptaulConstraint ; ProcedureStartTime_1 ; ProcedureFinishTime_1 ; ChangeOverTime_1 ; SequencingRequirement_1 ; Allocation_5 ; ; ;
```

E.2. CComp-CCompEl Connections

Here, each different CComp given in Section E.1 has been introduced with global Component name and CCompEl connections.

```
Ccomp Code;ConstraintComponent;ConceptualComponentElement1;ConceptualComponentElement2;ConceptualComponentElement3;ConceptualComponentElement4
Allocation_1;Allocation;ProceduralModel;PhysicalModel;TimeModel;
Allocation_2;Allocation;ProceduralModel;PhysicalEntityCapability;TimeModel;
Allocation_3;Allocation;UnitProcedure;Unit;TimeModel;
Allocation_4;Allocation;ProceduralModel;Worker;TimeModel;
Allocation_5;Allocation;ProceduralModel;PhysicalModel;;
Allocation_6;Allocation;TimePeriod;PhysicalModel;;
Allocation_7;Allocation;UnitProcedure;Unit;TimePeriod;
Allocation_8;Allocation;ProceduralModel;ProceduralModel;;
BatchSize_1;BatchSize;UnitProcedure;Unit;StateModel;
ChangeOverTime_1;ChangeOverTime;ProceduralModel;;;
ChangeOverTime_2;ChangeOverTime;ProceduralModel;ProceduralModel;;
CleaningTime_1;CleaningTime;ProceduralModel;ProceduralModel;;
ConsumedAmount_1;ConsumedAmount;;;;
ConsumedAmount_2;ConsumedAmount;PhysicalModel;TimeModel;;
ConsumedAmountCompletionTime_1;ConsumedAmountCompletionTime;ProceduralModel;;;
ConsumedMaterial_1;ConsumedMaterial;StateModelInput;TimeModel;ProceduralModel;PhysicalModel
ConsumedMaterial_2;ConsumedMaterial;Resource;TimeModel;ProceduralModel;PhysicalModel
ConsumedMaterial_3;ConsumedMaterial;UtilityResource;TimeModel;ProceduralModel;PhysicalModel
ConsumedMaterial_4;ConsumedMaterial;StoredMaterial;TimeModel;ProceduralModel;PhysicalModel
ConsumedMaterial_5;ConsumedMaterial;TransferableMaterials;TimeModel;ProceduralModel;PhysicalModel
Cost _1;Cost ;StateModelInput;;;
CostOfProcedure_1;CostOfProcedure;ProceduralModel;;;
CostOfRevenue_1;CostOfRevenue;StateModelInput;;;
CostOfStateModel_1;CostOfStateModel;StateModel;;;
CurrentlyAvailableAmount_1;CurrentlyAvailableAmount;;;;
CurrentlyAvailableAmount_2;CurrentlyAvailableAmount;PhysicalModel;TimeModel;;
CurrentlyAvailableMaterial_1;CurrentlyAvailableMaterial;StateModel;TimeModel;;
CurrentlyAvailableMaterial_2;CurrentlyAvailableMaterial;Resource;TimeModel;;
CurrentlyAvailableMaterial_3;CurrentlyAvailableMaterial;UtilityResource;TimeModel;;
CurrentlyAvailableMaterial_4;CurrentlyAvailableMaterial;StoredMaterial;TimeModel;;
CurrentlyAvailableMaterial_5;CurrentlyAvailableMaterial;TransferableMaterials;TimeModel;;
Demand_1;Demand;StateModelInput;TimeModel;;
DurationOfProceduralModel_1;DurationOfProceduralModel;ProceduralModel;;;
DurationOfProceduralModel_2;DurationOfProceduralModel;UnitProcedure;;;
FormatChangeTime_1;FormatChangeTime;ProceduralModel;ProceduralModel;;
Horizon_1;Horizon;;;;
InitialStorage_1;InitialStorage;StateModel;;;
Input_1;Input;InputEntity;;;
InputAmount_1;InputAmount;PhysicalModel;TimeModel;;
InputProportion_1;InputProportion;StateModel;ProceduralModel;;
InventeryCost_1;InventeryCost;TimePeriod;PhysicalModel;;
LengthOfProcedure_1;LengthOfProcedure;ProceduralModel;PhysicalModel;;
NumberOfTrip_1;NumberOfTrip;;;;
Output_1;Output;OutputEntity;;;
OutputAmount_1;OutputAmount;PhysicalModel;StateModel;;
```

```
OutputProportion_1 ; OutputProportion ; StateModel ; ProceduralModel ; ;
PreviouslyAvailableAmount_1 ; PreviouslyAvailableAmount ; ; ; ;
PreviouslyAvailableAmount_2 ; PreviouslyAvailableAmount ; PhysicalModel ; TimeModel ; ;
PreviouslyAvailableMaterial_1 ; PreviouslyAvailableMaterial ; StateModel ; TimeModel ; ;
PreviouslyAvailableMaterial_2 ; PreviouslyAvailableMaterial ; Resource ; TimeModel ; ;
PreviouslyAvailableMaterial_3 ; PreviouslyAvailableMaterial ; UtilityResource ; TimeModel ; ;
PreviouslyAvailableMaterial_4 ; PreviouslyAvailableMaterial ; StoredMaterial ; TimeModel ; ;
PreviouslyAvailableMaterial_5 ; PreviouslyAvailableMaterial ; TransferableMaterials ;
        TimeModel ; ;
PreviouslyProcessedAmount_1 ; PreviouslyProcessedAmount ; StateModelOutput ; TimeModel ;
        ProceduralModel ;
PreviouslyStoredAmount_1 ; PreviouslyStoredAmount ; PhysicalModel ; TimeModel ; ;
Price_1 ; Price ; StateModelOutput ; ; ;
PriceOfProduct_1 ; PriceOfProduct ; StateModelOutput ; ; ;
PriceOfStateModelElement_1 ; PriceOfStateModelElement ; StateModel ; ; ;
ProcedureFinishTime_1 ; ProcedureFinishTime ; ProceduralModel ; ; ;
ProcedureFinishTime_2 ; ProcedureFinishTime ; ProceduralModel ; TimeModel ; ;
ProcedureFinishTime_3 ; ProcedureFinishTime ; ProceduralModel ; TimeModel ; PhysicalModel ;
ProcedureStartTime_1 ; ProcedureStartTime ; ProceduralModel ; ; ;
ProcedureStartTime_2 ; ProcedureStartTime ; ProceduralModel ; TimeModel ; ;
ProcedureStartTime_3 ; ProcedureStartTime ; ProceduralModel ; TimeModel ; PhysicalModel ;
ProcessedAmount_1 ; ProcessedAmount ; PhysicalModel ; ProceduralModel ; StateModel ;
ProcessedAmount_2 ; ProcessedAmount ; StateModelOutput ; TimeModel ; ProceduralModel ;
ProcessedAmount_3 ; ProducedAmount ; ; ; ;
ProcessedAmount_4 ; ProducedAmount ; PhysicalModel ; StateModel ; ;
ProducedAmountCompletionTime_1 ; ProducedAmountCompletionTime ; ProceduralModel ; ; ;
ProducedMaterial_1 ; ProducedMaterial ; StateModelOutput ; TimeModel ; ProceduralModel ;
        PhysicalModel
ProducedMaterial_2 ; ProducedMaterial ; Resource ; TimeModel ; ProceduralModel ; PhysicalModel
ProducedMaterial_3 ; ProducedMaterial ; UtilityResource ; TimeModel ; ProceduralModel ;
        PhysicalModel
ProducedMaterial_4 ; ProducedMaterial ; StoredMaterial ; TimeModel ; ProceduralModel ;
        PhysicalModel
ProducedMaterial_5 ; ProducedMaterial ; TransferableMaterials ; TimeModel ; ProceduralModel ;
        PhysicalModel
ProducedMaterial_6 ; ProducedMaterial ; StateModel ; TimeModel ; ProceduralModel ;
        PhysicalModel
ProductionAmount_1 ; ProductionAmount ; StateModel ; PhysicalModel ; ;
PurchasedAmount_1 ; PurchasedAmount ; ; ; ;
PurchasedMaterial_1 ; PurchasedMaterial ; StateModel ; TimeModel ; ;
PurchasedMaterial_2 ; PurchasedMaterial ; Resource ; TimeModel ; ;
PurchasedMaterial_3 ; PurchasedMaterial ; UtilityResource ; TimeModel ; ;
PurchasedMaterial_4 ; PurchasedMaterial ; StoredMaterial ; TimeModel ; ;
PurchasedMaterial_5 ; PurchasedMaterial ; TransferableMaterials ; TimeModel ; ;
PurchasedMaterial_6 ; PurchasedMaterial ; StateModel ; TimeModel ; SoldMaterial ; TimeModel
RawMaterialCost_1 ; RawMaterialCost ; TimePeriod ; PhysicalModel ; ;
Sale_1 ; Sale ; StateModel ; PhysicalModel ; ;
Sale_2 ; Sale ; TimeModel ; StateModel ; ProceduralModel ;
SellingPrice_1 ; SellingPrice ; TimeModel ; StateModel ; ProceduralModel ;
SequencingAlllocation_1 ; SequencingAlllocation ; ProceduralModel ; TimeModel ; ;
SequencingRequirement_1 ; SequencingRequirement ; ProceduralModel ; PhysicalModel ; ;
SoldAmount_1 ; SoldAmount ; ; ; ;
SoldMaterial_1 ; SoldMaterial ; StateModel ; TimeModel ; ;
SoldMaterial_2 ; SoldMaterial ; TransferableMaterials ; TimeModel ; ;
SoldMaterial_3 ; SoldMaterial ; TransferableMaterials ; ; ;
SoldMaterial_4 ; SoldMaterial ; Resource ; TimeModel ; ;
SoldMaterial_5 ; SoldMaterial ; UtilityResource ; TimeModel ; ;
```

```
SoldMaterial_6 ; SoldMaterial ; StoredMaterial ; TimeModel ; ;
StoredAmount_1 ; StoredAmount ; PhysicalModel ; TimeModel ; ;
StoredAmount_2 ; StoredAmount ; StateModel ; PhysicalModel ; ;
SuppliedDemand_1 ; SuppliedDemand ; StateModel ; TimeModel ; ;
SuppliedDemand_2 ; SuppliedDemand ; StateModel ; PhysicalModel ; ;
TotalInvestmentCost_1 ; TotalInvestmentCost ; TimeModel ; PhysicalModel ; ;
TotalRawMaterialCost_1 ; TotalRawMaterialCost ; TimeModel ; PhysicalModel ; ;
TotalRevenue_1 ; TotalRevenue ; TimePeriod ; StateModel ; ;
TotalSale_1 ; TotalSale ; TimePeriod ; StateModel ; PhysicalModel ;
TotalTransportationCost_1 ; TotalTransportationCost ; TimeModel ; PhysicalModel ; ;
TransferredProduct_1 ; TransferredProduct ; StateModel ; PhysicalModel ; ;
TransportationCost_1 ; TransportationCost ; TimePeriod ; PhysicalModel ; ;
UpperOrLowerBound_1 ; UpperOrLowerBound ; ; ; ;
UpperOrLowerBound_2 ; UpperOrLowerBound ; ProceduralModel ; PhysicalModel ; ;
UpperOrLowerBoundAmountOfReturn_1 ; UpperOrLowerBoundAmountOfReturn ; StateModel ; PhysicalModel ; ;
UpperOrLowerBoundNumberOfTrip_1 ; UpperOrLowerBoundNumberOfTrip ; StateModel ; PhysicalModel ; ;
UpperOrLowerBoundOFBatchCapacity_1 ; UpperOrLowerBoundOFBatchCapacity ; UnitProcedure ; Unit ; ;
UpperOrLowerBoundOfProcessingCapacity_1 ; UpperOrLowerBoundOfProcessingCapacity ; ProceduralModel ; PhysicalModel ; ;
UpperOrLowerBoundOfStorageCapacity_1 ; UpperOrLowerBoundOfStorageCapacity ; StateModel ; PhysicalModel ; ;
UpperOrLowerBoundProductionCapacity_1 ; UpperOrLowerBoundProductionCapacity ; StateModel ; PhysicalModel ; ;
UpperOrLowerBoundSale_1 ; UpperOrLowerBoundSale ; StateModel ; PhysicalModel ; ;
UpperOrLowerBoundStockLevel_1 ; UpperOrLowerBoundStockLevel ; StateModel ; PhysicalModel ; ;
UpperOrLowerBoundTransportSize_1 ; UpperOrLowerBoundTransportSize ; StateModel ; PhysicalModel ; ;
UpperOrLowerBoundUtility_1 ; UpperOrLowerBoundUtility ; StateModel ; PhysicalModel ; ;
UsedUtilityAmount_1 ; UsedUtilityAmount ; StateModel ; PhysicalModel ; ;
Variable_1 ; Variable ; ; ; ;
Variable_2 ; Variable ; ProceduralModel ; PhysicalModel ; TimeModel ;
ChargingEfficiency_1 ; ChargingEfficiency ; ProceduralModel ; ; ;
DischargingEfficiency_1 ; DischargingEfficiency ; ProceduralModel ; ; ;
```

E.3. CCompEl - Connection Matrix

Each CCompEl from the unique level in the connection matrix is introduced with global component element, query type, and CCompEl levels.

```
GlobalComponentElement;Type-Query;CCompElement_Level 1;CCompElement_Level 2;
    CCompElement_Level 3;CCompElement_Level 4;CCompElement_Level 5;
    CCompElement_Level 6;CCompElement_Level 7
InputEntity;Concept;ControlModule;EquipmentModule;Unit;ProcessCell;Area;Site;
    Enterprise
OutputEntity;Concept hasPhysicalEntity Concept;Phase;Operation;UnitProcedure;
    Procedure;AreaProcedure;SiteProcedure;EnterpriseProcedure
PhysicalEntityCapability;;ProcessActionInput;ProcessOperationInput;ProcessStageInput;
    ProcessCellInput;AreaProcessSegmentInput;SiteProcessSegmentInput;
    EnterpriseProcessSegmentInput
PhysicalModel;;ProcessActionOutput;ProcessOperationOutput;ProcessStageOutput;
    ProcessCellOutput;AreaProcessSegmentOutput;SiteProcessSegmentOutput;
    EnterpriseProcessSegmentOutput
ProceduralModel;;ProcessActionState;ProcessOperationState;ProcessStageState;
    ProcessCellState;AreaProcessSegmentState;SiteProcessSegmentState;
    EnterpriseProcessSegmentState
StateModelInput;;;;;;;;
StateModel;;;;;;;;
StateModelOutput;;;;;;;;
TimeModel;;;;;;;;
Worker;;;;;;;;
```

Appendix F.

Notation

Acronyms

ABox	Assertion Box
AI	Artificial Intelligence
AIMMS	Advanced Interactive Multidimensional Modeling System
AMPL	A Mathematical Programming Language
BaPrOn	Batch Process Ontology
CC	Conceptual Constraint
DL	Descriptions Logic
DMM	Decision-Making Model
DMMs	Decision-Making Models
DSS	Decision Support System
GAMS	General Algebraic Modeling System
ISA	International Society of Automation
ISA88	Standard S88
ISA882010	The ISA88 Standard Part1 (Batch Control Part1: Models and Terminology) published in 2010
ISA88PART2	The ISA88 Standard Part2 (BatchControl Part2: Data Structures and Guidelines for Languages)
ISA88PART3	The ISA88 Standard Part3 (Batch Control Part3: General and Site Recipe Models and Representation)
ISA88PART4	The ISA88 Standard Part4 (Batch Control Part4: Batch Production Records)
ISA88R2006	The ISA88 Standard Part1 (Batch Control Part1: Models and Terminology) published in 2006 as a revision of first version
ISA95	Standard S95
ISO	International Organization for Standardization
LCA	Life Cycle Analysis

LP	Linear programming
MILP	Mixed-integer linear programming
MINLP	Mixed-integer non-linear programming
NAMUR	Interessengemeinschaft Automatisierungstechnik der Prozessindustrie - User Association of Automation Technology in Process Industries
NL	Natural language
NLP	Non-linear programming
OR	Operations Research
OWL	Ontology web language
PS	Process Systems
PSE	Process Systems Engineering
PSM	Process System Management
RFD	Resource Description Framework
RFD-S	Resource Description Framework Schema
RTN	Resource task network
SCM	Supply chaing management
SECOND	SEmi-automatic CONstruction of Domain Ontologies
STN	State task network
TBox	Terminological box
XML	Extensible Markup Language
XSD	XML Schema Definition

Part VII.

Bibliography

Bibliography

ABB (2018), System 800xA Batch Management, Technical report, ABB. **URL:** *https://search-ext.abb.com/library/Download.aspx?DocumentID=3BUA000145-600&LanguageCode=en&DocumentPartId=&Action=Launch (last access May 2018)*

AI Foundations of Computational Agents (2018), 'Ontologies and Knowledge Sharing'. **URL:** *http://artint.info/html/ArtInt_316.html (last acces January 2018)*

AIMMS (2018), 'Advanced Interactive Multidimensional Modeling System'. **URL:** *http://aimms.com/ (last access May 2018)*

Álvarez, R. and Ruiz-Puente, C. (2016), 'Development of the Tool SymbioSyS to Support the Transition Towards a Circular Economy Based on Industrial Symbiosis Strategies', *Waste and Biomass Valorization* pp. 1–10. **URL:** *http://link.springer.com/10.1007/s12649-016-9748-1*

Amar, F. B. B., Gargouri, B. and Hamadou, A. B. (2016), 'Generating core domain ontologies from normalized dictionaries', *Engineering Applications of Artificial Intelligence* **5**1, 230–241. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S095219761600018X*

AMPL (2018), 'A Mathematical Programming Language'. **URL:** *http://ampl.com/ (last access May 2018)*

Aspen (2018), 'AspenTech'. **URL:** *http://home.aspentech.com/ (last access May 2018)*

Baldea, M. (2015), 'From process integration to process intensification', *Computers & Chemical Engineering* **8**1, 104–114. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135415000824*

Baldea, M., Du, J. , Park, J. and Harjunkoski, I. (2015), 'Integrated production scheduling and model predictive control of continuous processes', *AIChE Journal* **6**1(12), 4179–4190. **URL:** *http://doi.wiley.com/10.1002/aic.14951*

Baldea, M. and Harjunkoski, I. (2014), 'Integrated production scheduling and process control: A systematic review', *Computers & Chemical Engineering* **7**1, 377–390. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135414002543*

Banek, M., Jurić, D. and Skočir, Z. (2010), Learning Semantic N-Ary Relations from Wikipedia, *in* 'Database and Expert Systems Applications', Springer, pp. 470–477. **URL:** *http://link.springer.com/10.1007/978-3-642-15364-8_39*

Bauer, N., Huuck, R. , Lukoschus, B. and Engell, S. (2004), A Unifying Semantics for Sequential Function Charts, *in* 'Integration of Software Specification Techniques for Applications in Engineering', Vol. 3147, Springer Berlin Heidelberg, pp. 400–418. **URL:** *http://link.springer.com/10.1007/978-3-540-27863-4_22*

Bechhofer, S. (2007), 'Programming to the OWL API : Introduction Overview Vision Need to Add 'Semantics' Ontology in Computer Science Why (Formal) Semantics ? A Brief History of OWL'. **URL:** *http://owlapi.sourceforge.net/SKB-SemTech-OWLAPI-6up.pdf (last access May 2018)*

Behdani, B. and Srinivasan, R. (2017), Managing supply chain disruptions: an integrated agent-oriented approach, *in* 'Computer Aided Chemical Engineering', Elsevier, pp. 595–600. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B978044463965350101X*

Benders, J. F. (1962), 'Partitioning procedures for solving mixed-variables programming problems', *Numerische Mathematik* **4**(1), 238–252. **URL:** *http://link.springer.com/10.1007/BF01386316*

Bertran, M.-O., Frauzem, R. , Sanchez-Arcilla, A.-S. , Zhang, L. , Woodley, J. M. and Gani, R. (2017), 'A generic methodology for processing route synthesis and design based on superstructure optimization', *Computers & Chemical Engineering* **1**06, 892–910. **URL:** *http://dx.doi.org/10.1016/j.compchemeng.2017.01.030 http://linkinghub.elsevier.com/retrieve/pii/S0098135417300303*

Bieszczad, J. (2000), A Framework for the Language and Logic of Computer-Aided Phenomena-Based Process Modeling by, PhD thesis, Massachusetts Institute of Technology. **URL:** *https://dspace.mit.edu/bitstream/handle/1721.1/16735/45145856-MIT.pdf;sequence=2 (last access May 2018)*

Biondi, M., Sand, G. and Harjunkoski, I. (2017), 'Optimization of multipurpose process plant operations: A multi-time-scale maintenance and production scheduling approach', *Computers & Chemical Engineering* **9**9, 325–339. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S009813541730008X*

Bog, A. (2014), *Benchmarking Transaction and Analytical Processing Systems*, In-Memory Data Management Research, Springer Berlin Heidelberg, Berlin, Heidelberg. **URL:** *http://link.springer.com/10.1007/978-3-642-38070-9*

Brandt, S. C., Morbach, J. , Miatidis, M. , Theißen, M. , Jarke, M. and Marquardt, W. (2008), 'An ontology-based approach to knowledge management in design processes', *Computers & Chemical Engineering* **3**2(1-2), 320–342. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135407001032*

Brayton, R., Carloni, L. P. , Sangiovanni-Vincentelli, A. L. and Villa, T. (2015), 'Design Automation of Electronic Systems: Past Accomplishments and Challenges Ahead [Scanning the Issue]', *Proceedings of the IEEE* **1**03(11), 1952–1957. **URL:** *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7302635*

BS-ISO/IEC 25010:2011 (2011), BSI Standards Publication Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models, Technical report, British Standard. **URL:** *https://www.iso.org/standard/35733.html*

CACE (n.d.), 'Computer Aided Chemical Engineering'. **URL:** *https://www.sciencedirect.com/bookseries/computer-aided-chemical-engineering*

Cagan, J., Grossmann, I. E. and Hooker, J. (1997), 'A conceptual framework for combining artificial intelligence and optimization in engineering design', *Research in Engineering Design* **9**(1), 20–34. **URL:** *http://www.springerlink.com/index/10.1007/BF01607055 http://link.springer.com/10.1007/BF01607055*

Calderón, A. J., Agnolucci, P. and Papageorgiou, L. G. (2017), 'An optimisation framework for the strategic design of synthetic natural gas (BioSNG) supply chains', *Applied Energy* **1**87, 929–955. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0306261916315185*

Cameron, I. and Gani, R. (2011), *Product and Process Modelling*, Elsevier. **URL:** *https://www.elsevier.com/books/product-and-process-modelling/cameron/978-0-444-53161-2*

Capón-García, E., Ferrer-Nadal, S. , Graells, M. and Puigjaner, L. (2009), 'An Extended Formulation for the Flexible Short-Term Scheduling of Multiproduct Semicontinuous Plants', *Industrial & Engineering Chemistry Research* **4**8(4), 2009–2019. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie800539f*

Capón-García, E., Muñoz, E. , Espuña, A. and Puigjaner, L. (2017), Recipe Management based on ISA-88 using Semantic Technologies, *in* 'Computer Aided Chemical Engineering', Elsevier, pp. 2293–2298. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444639653503846*

Capón-García, E., Muñoz, E. , Laínez-Aguirre, J. M. , Espuña, A. and Puigjaner, L. (2015), Enterprise-Wide Scheduling Framework Supported by Knowledge Management, *in* 'Computer Aided Chemical Engineering', Vol. 37, pp. 983–988. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444635778500097*

Cardoso, S. R., Barbosa-Póvoa, A. P. and Relvas, S. (2016), 'Integrating financial risk measures into the design and planning of closed-loop supply chains', *Computers & Chemical Engineering* **8**5, 105–123. **URL:** *http://dx.doi.org/10.1016/j.compchemeng.2015.10.012 http://linkinghub.elsevier.com/retrieve/pii/S0098135415003415*

Carreras, X., Chao, I. , Padró, L. and Padró, M. (2004), FreeLing: An Open-Source Suite of Language Analyzers, *in* 'Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)', pp. 239–242. **URL:** *http://nlp.lsi.upc.edu/papers/carreras04.pdf (last access May 2018)*

Carroll, S. O., Wallace, J. , Pineo, C. , Basson, L. , Woodcock, J. , Daniel, R. and Mouton, C. (2014), 'The Western Cape Industrial Symbiosis Programme (WISP): An Innovative Approach to Resource Efficiency and Waste Minimisation for South African Businesses', *Proceedings of the 20th WasteCon Conference 6-10 October 2014* pp. 232–241. **URL:** *http://www.infrastructurene.ws/wp-content/uploads/sites/4/2016/01/O'Carroll-S.-et-al-80.pdf (last access May 2018)*

Castro, P. M., Barbosa-Póvoa, A. P. , Matos, H. A. and Novais, A. Q. (2004), 'Simple Continuous-Time Formulation for Short-Term Scheduling of Batch and Continuous Processes', *Industrial & Engineering Chemistry Research* **4**3(1), 105–118. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie0302995*

Cea, G. A. and Montiel-Ponsoda, E. (2007), Benefits of Ontologies to Multilingual Needs, *in* M.Kuteeva and H. F.Martins, eds, '6th International AELFE Conference', pp. 519–531. **URL:** *http://oa.upm.es/5196/ (last access May 2018)*

Cecelja, F., Trokanas, N. , Raafat, T. and Yu, M. (2015), 'Semantic algorithm for Industrial Symbiosis network synthesis', *Computers & Chemical Engineering* **8**3, 248–266. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135415001325*

Chen, C.-L., Wang, B.-W. and Lee, W.-C. (2003), 'Multiobjective Optimization for a Multienterprise Supply Chain Network', *Industrial & Engineering Chemistry Research* **4**2(9), 1879–1889. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie0206148*

Chu, Y. and You, F. (2012), 'Integration of scheduling and control with online closed-loop implementation: Fast computational strategy and large-scale global optimization algorithm', *Computers & Chemical Engineering* **4**7, 248–268. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135412002190*

Cimiano, P., Staab, S. and Tane, J. (2003), Automatic acquisition of taxonomies from text: FCA meets NLP, *in* 'ECML-PKDD Workshop on Adaptive Text Extraction and Mining (ATEM 2003) at 14th European Conference on Machine Learning (ECML 2003) and the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003)', pp. 10–17. **URL:** *https://pub.uni-bielefeld.de/download/2497914/2546336 (last access May 2018)*

Cimiano, P. and Völker, J. (2005), Text2Onto, *in* 'Natural Language Processing and Information Systems', Springer, pp. 227–238. **URL:** *http://link.springer.com/10.1007/11428817_21*

CO-LaN (2018), 'CAPE-OPEN Standard'. **URL:** *http://www.colan.org/ (last access March 2018)*

ControlDraw (2015), The Visual Automation Design Database, Technical report, ControlDraw.

Corcho, O., Fernández-López, M. and Gómez-Pérez, A. (2003), ‘Methodologies, tools and languages for building ontologies. Where is their meeting point?’, *Data & Knowledge Engineering* **4**6(1), 41–64. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0169023X02001957*

Council of Supply Chain Management Professionals (2013), Supply Chain Management Terms and Glossary, Technical report, CSCMP. **URL:** *https://goo.gl/y5ASu5 (last access May 2018))*

Dantzig, G. B., Orden, A. and Wolfe, P. (1953), ‘The generalized simplex method for minimizing a linear form under linear unequlity restraints’, *Pacific Journal of Mathematics* **5**(2). **URL:** *https://msp.org/pjm/1955/5-2/pjm-v5-n2-p04-s.pdf*

Daoutidis, P., Lee, J. H. , Harjunkoski, I. , Skogestad, S. , Baldea, M. and Georgakis, C. (2018), ‘Integrating operations and control: A perspective and roadmap for future research’, *Computers & Chemical Engineering* **1**15, 179–184. **URL:** *https://doi.org/10.1016/j.compchemeng.2018.04.011 https://linkinghub.elsevier.com/retrieve/pii/S0098135418303132*

De Meyer, A., Cattrysse, D. and Van Orshoven, J. (2015), ‘A generic mathematical model to optimise strategic and tactical decisions in biomass-based supply chains (OPTIMASS)’, *European Journal of Operational Research* **2**45(1), 247–264. **URL:** *http://dx.doi.org/10.1016/j.ejor.2015.02.045 http://linkinghub.elsevier.com/retrieve/pii/S0377221715001654*

Dombayci, C., Capón-García, E. , Muñoz, E. and Espuña, A. (2017), Constraint Identification and Integration Procedures in Multi-Level Hierarchical Systems, *in* ‘Computer Aided Chemical Engineering’, Vol. 40, pp. 2359–2364. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444639653503950*

Dombayci, C. and Espuña, A. (2017), Systematic decision making models through Conceptual Constraints, *in* ‘Computer Aided Chemical Engineering’, Vol. 40, pp. 1873–1878. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444639653503147*

Dombayci, C. and Espuña, A. (2018*a*), ‘Building decision-making models: the Conceptual Constraint Domain modelling’, *Manuscript submitted* .

Dombayci, C. and Espuña, A. (2018*b*), Building Decision Making Models Through Conceptual Constraints: Multi-scale Process Model Implementations, *in* A.Fink, A.Fügenschuh and M.-J.Geiger, eds, ‘Operations Research Proceedings 2016’, Springer International Publishing, pp. 77–83. **URL:** *http://www.springer.com.recursos.biblioteca.upc.edu/gp/book/9783319557014#aboutAuthors http://link.springer.com/10.1007/978-3-319-55702-1_12*

Dombayci, C., Farreres, J. , Rodríguez, H. , Espuña, A. and Graells, M. (2017), ‘Improving automation standards via semantic modelling: Application to ISA88’, *ISA Transactions* **6**7, 443–454. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0019057817300691*

Dombayci, C., Farreres, J. , Rodríguez, H. , Espuña, A. and Graells, M. (2018), ‘Systematic Construction of Domain Ontologies from Normative Documents’, *Submitted to Artificial Intelligence* .

Dombayci, C., Farreres, J. , Rodríguez, H. , Muñoz, E. , Capón-García, E. , Espuña, A. and Graells, M. (2015), On the Process of Building a Process Systems Engineering Ontology Using a Semi-Automatic Construction Approach, *in* ‘Computer Aided Chemical Engineering’, Vol. 37, pp. 941–946. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444635778500024*

Douglas, J. M. (1988), *Conceptual design of chemical processes*, McGraw-Hill Book Company.

Doyle, F. J., Pekny, J. F. , Reklaitis, G. and Venkatasubramanian, V. (1997), ‘A graduate course in computer integrated process operations’, *Computers & Chemical Engineering* **2**1(S), S255–S259. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135497875114*

Edgar, T. F., Dixon, D. A. and Reklaitis, G. V. (1999), Vision 2020: Computational Needs of the Chemical Industry, *in* 'Impact of Advances in Computing and Communications Technologies on Chemical Science and Technology: Report of a Workshop', National Academies Press (US), chapter 6, pp. 74–90. **URL:** *https://www.ncbi.nlm.nih.gov/books/NBK44988 (last access May 2018)*

Edgar, T. F., Himmelblau, D. M. and Lasdon, L. (1988), *Optimization of Chemical Processes*, McGraw-Hill.

El-Halwagi, M. M. (2017), 'A Shortcut Approach to the Multi-scale Atomic Targeting and Design of C-H-O Symbiosis Networks', *Process Integration and Optimization for Sustainability* pp. 1–11. **URL:** *http://link.springer.com/10.1007/s41660-016-0001-y*

Elve, A. T. and Preisig, H. A. (2017), From Ontology to Executable Program Code, *in* 'Computer Aided Chemical Engineering', Vol. 40, Elsevier, pp. 2317–2322. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444639653503883*

Engell, S. and Harjunkoski, I. (2012), 'Optimal operation: Scheduling, advanced control and their integration', *Computers & Chemical Engineering* **4**7, 121–133. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135412002232*

ETSI and European Telecommunications Standards Institute (2015), Why we need standards, Technical report, European Telecommunications Standards Institute. **URL:** *http://www.etsi.org/standards/why-we-need-standards (last access May 2018)*

Farreres, J., Gibert, K. , Rodríguez, H. and Pluempitiwiriyawej, C. (2010), 'Inference of lexical ontologies. The LeOnI methodology', *Artificial Intelligence* **1**74(1), 1–19. **URL:** *http://dx.doi.org/10.1016/j.artint.2009.09.004 http://linkinghub.elsevier.com/retrieve/pii/S0004370209001064*

Farreres, J., Graells, M. , Rodríguez, H. and Espuña, A. (2014), Towards Automatic Construction of Domain Ontologies: Application to ISA88, *in* J. J.Klemeš, P. S.Varbanov and P. Y.Liew, eds, 'Proceedings of the 24th European Symposium on Computer Aided Process Engineering', Elsevier, pp. 871–876. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444634566501460*

Fedorova, M., Sin, G. and Gani, R. (2015), 'Computer-aided modelling template: Concept and application', *Computers & Chemical Engineering* **8**3, 232–247. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135415000551*

Figueira, G., Amorim, P. , Guimarães, L. , Amorim-Lopes, M. , Neves-Moreira, F. and Almada-Lobo, B. (2015), 'A decision support system for the operational production planning and scheduling of an integrated pulp and paper mill', *Computers & Chemical Engineering* **7**7, 85–104. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135415000885*

Fischer Bandy, J. and Emerson, D. (1996), 'Common resources in SP88 style applications presented at the World Batch Forum', *ISA Transactions* **3**5(3), 267–273. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0019057896000365*

Fleischmann, B. and Oberstein, A. (2008), Strategic Network Design, *in* H.Stadtler and C.Kilger, eds, 'Supply Chain Management and Advanced Planning', Springer Berlin Heidelberg, Berlin, Heidelberg, chapter Strategic. **URL:** *http://link.springer.com/10.1007/978-3-540-74512-9*

Francisco, M., Revollar, S. , Vega, P. and Lamanna, R. (2005), 'A Comparative Study of Deterministic and Stochastic Optimization Methods for Integrated Design of Processses', *IFAC Proceedings Volumes* **3**8(1), 335–340. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S1474667016369294*

Franz, B., Diego, C. , Deborah L., M. , Daniele, N. and Patel-Scheider, P. F. (2007), *The Description Logic Handbook: Theory, implementation, and applications*, Vol. 1, 2 edn, Cambridge University Press.

GAMS (2016), 'General Algebraic Modeling System'. **URL:** *http://www.gams.com/ (last access May 2018)*

Gani, R. and E. Grossmann, I. (2007), Process systems engineering and CAPE - what next?, *in* 'Proceedings of the 17th European Symposium on Computer Aided Process Engineering', pp. 1–5. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S157079460780023X*

Georgiadis, M. C. and Papageorgiou, L. G. (2008), *Process Systems Engineering, Volume 3: Supply Chain Optimization, Part I*, Vol. 3, Wiley-VCH.

Girju, R., Badulescu, A. and Moldovan, D. (2006), 'Automatic Discovery of Part-Whole Relations', *Computational Linguistics* **3**2(1), 83–135. **URL:** *http://www.mitpressjournals.org/doi/abs/10.1162/coli.2006.32.1.83*

Glavič, P. (2012), 'Thirty Years of International Symposia on Process Systems Engineering', *Current Opinion in Chemical Engineering* **1**(4), 421–429. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S221133981200072X*

Godena, G. (2009), 'A new proposal for the behaviour model of batch phases', *ISA Transactions* **4**8(1), 3–9. **URL:** *10.1016/j.isatra.2008.08.002 http://linkinghub.elsevier.com/retrieve/pii/S0019057808000475*

Godena, G., Lukman, T. , Steiner, I. , Bergant, F. and Strmčnik, S. (2015), 'A new object model of batch equipment and procedural control for better recipe reuse', *Computers in Industry* **7**0, 46–55. **URL:** *http://www.sciencedirect.com/science/article/pii/S016636151500024X http://linkinghub.elsevier.com/retrieve/pii/S016636151500024X*

Goedkoop, M., Heijungs, R. , Huijbregts, M. , Schryver, A. D. , Struijs, J. and Zelm, R. V. (2009), ReCiPe, Technical report, Ministry of Housing, Spatial Planning and Environment. **URL:** *http://www.pre-sustainability.com/download/misc/ReCiPe_main_report_final_27-02-2009_web.pdf (last access May 2018)*

Goetschalckx, M. (2011), *Supply Chain Engineering*, Vol. 161 of *International Series in Operations Research & Management Science*, Springer.

Gonzalez-Perez, C., Henderson-Sellers, B. , McBride, T. , Low, G. and Larrucea, X. (2016), 'An Ontology for ISO software engineering standards: 2) Proof of concept and application', *Computer Standards & Interfaces* **4**8, 112–123. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0920548916300344*

Graells, M. (2016), 'Lecture Notes: Production Planning of Chemical Processes'. **URL:** *ftp://ftp-urgell.upc.es/quimica/M.Graells/Planning_and_Scheduling_of_Chemical_Processes/ (last access May 2018)*

GreenCape (2017), Waste Economoy: 2017 Market Intelligence Report, Technical report, GreenCape. **URL:** *http://www.green-cape.co.za/assets/Uploads/GreenCape-Waste-MIR-2017-electronic-FINAL-v2.pdf (last access May 2018)*

Grossmann, I. (2005), 'Enterprise-wide optimization: A new frontier in process systems engineering', *AIChE Journal* **5**1(7), 1846–1857. **URL:** *http://doi.wiley.com/10.1002/aic.10617*

Grossmann, I. E., Erdirik-Dogan, M. and Karuppiah, R. (2008), 'Overview of Planning and Scheduling for Enterprise-wide Optimization of Process Industries (Übersicht über Planungs- und Feinplanungssysteme für die unternehmensweite Optimierung in der verfahrenstechnischen Industrie', *Automatisierungstechnik* **5**6(2), 64–79. **URL:** *http://www.degruyter.com/view/j/auto.2008.56.issue-2/auto.2008.0690/auto.2008.0690.xml*

Grossmann, I. E. and Westerberg, A. W. (2000), 'Research challenges in process systems engineering', *AIChE Journal* **4**6(9), 1700–1703. **URL:** *http://doi.wiley.com/10.1002/aic.690460902*

Gruber, T. R. (1993), ‘A translation approach to portable ontology specifications’, *Knowledge Acquisition* **5**(2), 199–220. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S1042814383710083*

Guillén, G., Badell, M. , Espuña, A. and Puigjaner, L. (2006), ‘Simultaneous optimization of process operations and financial decisions to enhance the integrated planning/scheduling of chemical supply chains’, *Computers & Chemical Engineering* **3**0(3), 421–436. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135405002516*

Guillén-Gosálbez, G., Mele, F. , Bagajewicz, M. , Espuña, A. and Puigjaner, L. (2005), ‘Multiobjective supply chain design under uncertainty’, *Chemical Engineering Science* **6**0(6), 1535–1553. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0009250904008449*

Hai, R., Theißen, M. and Marquardt, W. (2011), ‘An ontology based approach for operational process modeling’, *Advanced Engineering Informatics* **2**5(4), 748–759. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S1474034611000322*

Halim, I., Carvalho, A. , Srinivasan, R. , Matos, H. A. and Gani, R. (2011), ‘A combined heuristic and indicator-based methodology for design of sustainable chemical process plants’, *Computers & Chemical Engineering* **3**5(8), 1343–1358. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135411001050*

Harjunkoski, I. (2014), Integration of Scheduling and ISA-95, *in* ‘Proceedings of the 24th European Symposium on Computer Aided Process Engineering’, Vol. 36, pp. 427–432. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444634566500727*

Harjunkoski, I., Maravelias, C. T. , Bongers, P. , Castro, P. M. , Engell, S. , Grossmann, I. E. , Hooker, J. N. , Méndez, C. , Sand, G. and Wassick, J. (2014), ‘Scope for industrial applications of production scheduling models and solution methods’, *Computers and Chemical Engineering* **6**2, 161–193. **URL:** *http://dx.doi.org/10.1016/j.compchemeng.2013.12.001*

Harjunkoski, I., Nyström, R. and Horch, A. (2009), ‘Integration of scheduling and control - Theory or practice?’, *Computers & Chemical Engineering* **3**3(12), 1909–1918. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135409001690*

Hearst, M. A. (1992), ‘Automatic acquisition of hyponyms from large text corpora’, *Proceedings of the 14th conference on Computational linguistics -* **2**, 539. **URL:** *http://portal.acm.org/citation.cfm?doid=992133.992154*

Henderson-Sellers, B., Gonzalez-Perez, C. , McBride, T. and Low, G. (2014), ‘An ontology for ISO software engineering standards: 1) Creating the infrastructure’, *Computer Standards & Interfaces* **3**6(3), 563–576. **URL:** *http://dx.doi.org/10.1016/j.csi.2013.11.001 http://linkinghub.elsevier.com/retrieve/pii/S0920548913001335*

Henning, G. (2017*a*), ‘Process Operations : A Roadmap from the Present to a Hyper-connected Future, 2040 Visions of Process Systems Engineering’. **URL:** *https://www.youtube.com/watch?v=cIeIkB9-p_8&list=PL1Z8XsQQiN4MUNGsCZSeoT_kr9J-loncC&index=4 http://stephanopoulos-symposium.mit.edu/ (last access May 2018)*

Henning, G. (2017*b*), Towards a software engineering approach to the deployment and implementation of scheduling systems, *in* ‘Computer Aided Chemical Engineering’, Vol. 40, Elsevier B.V., pp. 1405–1410. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444639653502361*

Herbrich, R. and Graepel, T. , eds (2012), *A First Course in Machine Learning*, 2nd edn, Chapman & Hall/CRC.

Hjaila, K., Puigjaner, L. , Laínez, J. M. and Espuña, A. (2017), ‘Integrated game-theory modelling for multi enterprise-wide coordination and collaboration under uncertain competitive environment’, *Computers & Chemical Engineering* **9**8, 209–235. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135416303933*

Holland, J. H. (1975), *Adaptation in Natural and Artificial Systems*, MIT Press.

Hooker, J. N. (2012), *Integrated Methods for Optimization*, Vol. 170 of *International Series in Operations Research & Management Science*, Springer US, Boston, MA. **URL:** *http://link.springer.com/10.1007/978-1-4614-1900-6*

Hopf, H. (2004), *Beyond the Molecular Frontier Challenges for Chemistry and Chemical Engineering. Edited by the Committee on Challenges for the Chemical Sciences in the 21st Century. Board on Chemical Sciences and Technologies.*, Vol. 43, The National Academies Press. **URL:** *http://doi.wiley.com/10.1002/anie.200385074*

IBM (2018*a*), 'IBM ILOG CPLEX Optimization Studio V12.6.3'. **URL:** *http://www-01.ibm.com/software/ (last access May 2018) https://goo.gl/CVqW3t*

IBM (2018*b*), 'Mathematical Programming vs. Constraint Programming'. **URL:** *http://www-01.ibm.com/software/integration/optimization/cplex-cp-optimizer/mp-cp/ (last access May 2018)*

IEC and International Electrotecnical Commision (2015), Why do we need standards?, Technical report, International Electrotechnical Commission. **URL:** *http://www.iec.ch/perspectives/general_public/why_do_we_need_standards_1.htm (last access May 2018)*

IEEE and Institute of Electrical and Electronics Engineers (2014), 2014 IEEE-SA Standard Style Manual, Technical report, IEEE Standard Association, New York, USA. **URL:** *https://development.standards.ieee.org/myprojct/Public/mytools/draft/styleman.pdf (last access May 2018)*

Ierapetritou, M. and Floudas, C. A. (1998), 'Effective Continuous-Time Formulation for Short-Term Scheduling. 2. Continuous and Semicontinuous Processes', *Industrial & Engineering Chemistry Research* **3**7(11), 4360–4374. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie9709289*

IMPL (2016), 'Industrial Modeling and Programming Language'. **URL:** *www.induapps.club/mission.html (last access December, 2016)*

Intelligen, I. (2018), 'SuperPro'. **URL:** *http://www.intelligen.com/ (last access May 2018)*

ISA (2000), *Enterprise-Control System Integration, Part 1: Models and Terminology, ANSI/ISA-95.00.01-2000*, ISA Committe.

ISA (2001), *Batch Control Part 2: Data Structures and Guidelines for Languages, ANSI/ISA-88.00.02-2001*, ISA Committee.

ISA (2003), *Batch Control Part 3: General and Site Recipe Models and Representation, ANSI/ISA-88.00.03-2003*, ISA Committee.

ISA (2006*a*), Batch Control, Part 1: Models and Terminology, ISA-88.01-1995, Technical report, ISA standard.

ISA (2006*b*), *Batch Control, Part 1: Models and Terminology, ISA-88.01-1995 (R2006)*, ISA Committe.

ISA (2006*c*), *Batch Control Part 4: Batch Production Records, ANSI/ISA-88.00.04-2006*, ISA Committee.

ISA (2010), *Batch Control, Part 1: Models and Terminology, ANSI/ISA-88.01-2010*, ISA Committe.

ISO British Standards Institution (2013), *Enterprise-control system integration, Part 1 : Models and terminology (BS EN 62264-1:2013)*, BSI Standards Publication.

ISO British Standards Institution (2014*a*), *Automation systems and integration - Key performance indicators (KPIs) for manufacturing operations management Part 1 : Overview , concepts and terminology (BS-ISO 22400-1:2014)*, ISO.

ISO British Standards Institution (2014*b*), *Automation systems and integration - Key performance indicators (KPIs) for manufacturing operations management Part 2: Definitions and descriptions (BS ISO 22400-2:2014)*, British Standard.

ISO and International Organization for Standardization (2014), How to write standard: Tips for standards writers, Technical report, ISO Standards, Geneve, Switzerland. **URL:** *http://www.iso.org/iso/how-to-write-standards.pdf (last access May 2018)*

ISO and International Organization for Standardization (2015), Benefits of International Standards, Technical report, ISO. **URL:** *http://www.iso.org/iso/home/standards/benefitsofstandards.htm (last access May 2018)*

ISO/IEC GUIDE 17:2016 (2016), Guide for writing standards taking into account the needs of micro, small and medium-sized enterprises, Technical report, British Standard. **URL:** *https://www.iso.org/standard/67340.html (last access May 2018)*

Jung, K., Morris, K. , Lyons, K. W. , Leong, S. and Cho, H. (2016), Performance Challenges Identification Method for Smart Manufacturing Systems, Technical report, National Institute of Standards and Technology, Gaithersburg, MD. **URL:** *http://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8108.pdf*

Jython (2018), 'Jython'. **URL:** *http://www.jython.org/ (last access May 2018)*

Kadam, J., Schlegel, M. , Marquardt, W. , Tousain, R. , van Hessem, D. , van den Berg, J. and Bosgra, O. (2002), *A Two-Level Strategy of Integrated Dynamic Optimization and Control of Industrial Processes-a Case Study*, Springer, chapter Integratio, pp. 511–516. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S1570794602801134*

Kall, P. and Mayer, J. (1996), 'SLP-IOR: An interactive model management system for stochastic linear programs', *Mathematical Programming, Series B* **7**5(2). **URL:** *https://link.springer.com/article/10.1007/BF02592153*

Kang, Y.-B., Delir Haghighi, P. and Burstein, F. (2014), 'CFinder: An intelligent key concept finder from text for ontology development', *Expert Systems with Applications* **4**1(9), 4494–4504. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0957417414000189*

Kelly, R. A., Jakeman, A. J. , Barreteau, O. , Borsuk, M. E. , ElSawah, S. , Hamilton, S. H. , Henriksen, H. J. , Kuikka, S. , Maier, H. R. , Rizzoli, A. E. , van Delden, H. and Voinov, A. A. (2013), 'Selecting among five common modelling approaches for integrated environmental assessment and management', *Environmental Modelling & Software* **4**7, 159–181. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S1364815213001151*

Klatt, K.-U. and Marquardt, W. (2009), 'Perspectives for process systems engineering-Personal views from academia and industry', *Computers & Chemical Engineering* **3**3(3), 536–550. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135408001737*

Kocis, G. R. and Grossmann, I. E. (1988), Global optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis, Technical Report 8, Carnegie Mellon University Engineering Design Research Center. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie00080a013*

Kokossis, A., Labrador-Darder, C. and Cecelja, F. (2016), 'Semantically enabled process synthesis and optimisation', *Computers & Chemical Engineering* **9**3, 64–86. **URL:** *https://goo.gl/OWtZWj*

Kondili, E., Pantelides, C. and Sargent, R. (1993), 'A general algorithm for short-term scheduling of batch operations - I. MILP formulation', *Computers & Chemical Engineering* **1**7(2), 211–227. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/009813549380015F*

Kraus, R., Fillinger, S. , Tolksdorf, G. , Minh, D. H. , Merchan-Restrepo, V. A. and Wozny, G. (2014), 'Improving Model and Data Integration Using MOSAIC as Central Data Management Platform', *Chemie Ingenieur Technik* **8**6(7), 1130–1136. **URL:** *http://doi.wiley.com/10.1002/cite.201400007*

Küçük, D. and Arslan, Y. (2014), 'Semi-automatic construction of a domain ontology for wind energy using Wikipedia articles', *Renewable Energy* **6**2, 484–489. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0960148113004035*

Kuntsche, S., Barz, T. , Kraus, R. , Arellano-Garcia, H. and Wozny, G. (2011), 'MOSAIC a web-based modeling environment for code generation', *Computers & Chemical Engineering* **3**5(11), 2257–2273. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135411001128*

Laínez, J. M., Guillén-Gosálbez, G. , Badell, M. , Espuña, A. and Puigjaner, L. (2007), 'Enhancing Corporate Value in the Optimal Design of Chemical Supply Chains', *Industrial & Engineering Chemistry Research* **4**6(23), 7739–7757. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie070181e*

Laínez, J. M., Kopanos, G. M. , Espuña, A. and Puigjaner, L. (2009), 'Flexible design-planning of supply chain networks', *AIChE Journal* **5**5(7), 1736–1753. **URL:** *http://doi.wiley.com/10.1002/aic.11942*

Laínez, J. M. and Puigjaner, L. (2012), 'Prospective and perspective review in integrated supply chain modelling for the chemical process industry', *Current Opinion in Chemical Engineering* **1**(4), 430–445. **URL:** *http://www.sciencedirect.com/science/article/pii/S2211339812000615 http://linkinghub.elsevier.com/retrieve/pii/S2211339812000615*

Lee, M. H., Han, C. and Chang, K. S. (1999), 'Dynamic Optimization of a Continuous Polymer Reactor Using a Modified Differential Evolution Algorithm', *Industrial & Engineering Chemistry Research* **3**8(12), 4825–4831. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie980373x*

Linninger, A. (1995), 'Generation and assessment of batch processes with ecological considerations', *Computers & Chemical Engineering* **1**9(1), S7–S13. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/009813549500144Q*

Linninger, A. (2017), 'Mathematical Models - Conversations between Minds and Machines, 2040 Visions of Process Systems Engineering'. **URL:** *https://www.youtube.com/watch?v=pqjWATDqqP8&list=PL1Z8XsQQiN4MUNGsCZSeoT_kr9J-loncC&index=6 (last access May 2018)*

Linninger, A., Chowdhry, S. , Bahl, V. , Krendl, H. and Pinger, H. (2000), 'A systems approach to mathematical modeling of industrial processes', *Computers & Chemical Engineering* **2**4(2-7), 591–598. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135400004257*

Lopez Cruz, I., Van Willigenburg, L. and Van Straten, G. (2003), 'Efficient Differential Evolution algorithms for multimodal optimal control problems', *Applied Soft Computing* **3**(2), 97–122. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S1568494603000073*

Luo, N., Zhong, W. , Wan, F. , Ye, Z. and Qian, F. (2015), 'An agent-based service-oriented integration architecture for chemical process automation', *Chinese Journal of Chemical Engineering* **2**3(1), 173–180. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S1004954114001578*

Lutz, C. (2002), The Complexity of Description Logics with Concrete Domains, PhD thesis, Rheinisch-Westfalischen Technischen Hochschule Aachen.

Luyben, M. L., Tyreus, B. D. and Luyben, W. L. (1997), 'Plantwide control design procedure', *AIChE Journal* **4**3(12), 3161–3174. **URL:** *http://doi.wiley.com/10.1002/aic.690431205*

Makowski, M. (2005), ‘A structured modeling technology’, *European Journal of Operational Research* **1**66(3), 615–648. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0377221704004060*

Malmberg, F., Lindblad, J. , Sladoje, N. and Nyström, I. (2011), ‘A graph-based framework for sub-pixel image segmentation’, *Theoretical Computer Science* **4**12(15), 1338–1349. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0304397510006687*

Maravelias, C. T. and Grossmann, I. E. (2003*a*), ‘Minimization of the Makespan with a Discrete-Time State-Task Network Formulation’, *Industrial & Engineering Chemistry Research* **4**2(24), 6252–6257. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie034053b*

Maravelias, C. T. and Grossmann, I. E. (2003*b*), ‘New General Continuous-Time State-Task Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants’, *Industrial & Engineering Chemistry Research* **4**2(13), 3056–3074. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie020923y*

Maravelias, C. T. and Sung, C. (2009), ‘Integration of production planning and scheduling: Overview, challenges and opportunities’, *Computers & Chemical Engineering* **3**3(12), 1919–1930. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135409001501*

Marquardt, W., Morbach, J. , Wiesner, A. and Yang, A. (2010), *OntoCAPE*, RWTHedition, Springer Berlin Heidelberg, Berlin, Heidelberg. **URL:** *http://link.springer.com/10.1007/978-3-642-04655-1*

MathWorks (2015), ‘Matlab (R2015a): Machine Learning Toolbox’. **URL:** *https://mathworks.com/products/statistics.html (last access May 2018)*

Matlab (2018), ‘MathWorks’. **URL:** *https://www.mathworks.com/ (last access May 2018)*

Mayne, D., Rawlings, J. , Rao, C. and Scokaert, P. (2000), ‘Constrained model predictive control: Stability and optimality’, *Automatica* **3**6(6), 789–814. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0005109899002149*

McFarlane, D., Carr, J. , Harrison, M. and McDonald, A. (2003), Auto-ID’s Three R’s: Rules and Recipes for Product Requirements, White paper, Auto-IC Centre, Institute for Manufacturing, University of Cambridge. **URL:** *https://www.ifm.eng.cam.ac.uk/uploads/Research/DIAL/Resources/White_papers/cam-autoid-wh-008.pdf (last access May 2018)*

McFate, C., Forbus, K. and Hinrichs, T. (2014), ‘Using Narrative Function to Extract Qualitative Information from Natural Language Texts’, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* pp. 373–379. **URL:** *http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/download/8466/8444*

McGuire, P. (2007), ‘Getting started with pyparsing’, *O’Reilly, California, US* . **URL:** *http://index-of.co.uk/Tutorials/Getting Started with Pyparsing.pdf*

Medina-González, S., Pozo, C. , Corsano, G. , Guillén-Gosálbez, G. and Espuña, A. (2017), ‘Using Pareto filters to support risk management in optimization under uncertainty: Application to the strategic planning of chemical supply chains’, *Computers & Chemical Engineering* **9**8, 236–255. **URL:** *http://dx.doi.org/10.1016/j.compchemeng.2016.10.008 http://linkinghub.elsevier.com/retrieve/pii/S0098135416303246*

Meijer, K., Frasincar, F. and Hogenboom, F. (2014), ‘A semantic approach for extracting domain taxonomies from text’, *Decision Support Systems* **6**2, 78–93. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0167923614001031*

Meissner, A. (2011), The ALCN Description Logic Concept Satisfiability as a SAT Problem, *in* 'Studies in Computational Intelligence', Vol. 381, Springer, pp. 253–263. **URL:** *http://link.springer.com/10.1007/978-3-642-23418-7_22*

Mele, F. D., Kostin, A. M. , Guillen-Gosálbez, G. and Jimenez, L. (2011), 'Multiobjective Model for More Sustainable Fuel Supply Chains. A Case Study of the Sugar Cane Industry in Argentina', *Industrial & Engineering Chemistry Research* **5**0(9), 4939–4958. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie101400g*

Méndez, C. A. and Cerdá, J. (2002), 'An efficient MILP continuous-time formulation for short-term scheduling of multiproduct continuous facilities', *Computers and Chemical Engineering* **2**6(4-5), 687–695.

Méndez, C. A., Cerdá, J. , Grossmann, I. E. , Harjunkoski, I. and Fahl, M. (2006), 'State-of-the-art review of optimization methods for short-term scheduling of batch processes', *Computers & Chemical Engineering* **3**0(6-7), 913–946. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135406000287*

MESA (2018), 'Business To Manufacturing Markup Language (B2MML)'. **URL:** *http://www.mesa.org/en/b2mml.asp (last access May 2018)*

Mesarovic, M. D., Macko, D. and Takahara, Y. (1970), *Theory of Hierarchical Multilevel Systems*, Academic Prress, System Research Center, Case Western Reserve University, Cleveland, OHIO.

Michalewicz, Z., Janikow, C. Z. and Krawczyk, J. B. (1992), 'A modified genetic algorithm for optimal control problems', *Computers & Mathematics with Applications* **2**3(12), 83–94. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/089812219290094X*

Mitchell, M. (1999), *An Introduction to Genetic Algorithms*, The MIT Press.

Mitsos, A. (2012), Applied Numerical Optimization Lecture Notes, Technical report, RWTH Aachen.

Moniz, S., Barbosa-Póvoa, A. P. , de Sousa, J. P. and Duarte, P. (2014), 'Solution Methodology for Scheduling Problems in Batch Plants', *Industrial & Engineering Chemistry Research* **5**3(49), 19265–19281. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie403129y*

Morbach, J. (2009), A Reusable Ontology for Computer-Aided Process Engineering, PhD thesis, RWTH Aachen University.

Morbach, J., Wiesner, A. and Marquardt, W. (2009), 'OntoCAPE - A (re)usable ontology for computer-aided process engineering', *Computers & Chemical Engineering* **3**3(10), 1546–1556. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135409000362*

Morbach, J., Yang, A. and Marquardt, W. (2007), 'OntoCAPE - A large-scale ontology for chemical process engineering', *Engineering Applications of Artificial Intelligence* **2**0(2), 147–161. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0952197606001199*

Mota, B., Gomes, M. I. , Carvalho, A. and Barbosa-Povoa, A. P. (2015), 'Towards supply chain sustainability: economic, environmental and social design and planning', *Journal of Cleaner Production* **1**05, 14–27. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0959652614007719*

Motard, R. (1983), 'Computer technology in process systems engineering', *Computers & Chemical Engineering* **7**(4), 483–491. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/0098135483800246*

Muñoz, E. (2011), Knowledge management technology for integrated decision support systems in process industries, PhD thesis, UPC.

Muñoz, E., Capón-García, E. , Espuña, A. and Puigjaner, L. (2012), 'Ontological framework for enterprise-wide integrated decision-making at operational level', *Computers & Chemical Engineering* **4**2, 217–234. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135412000440*

Muñoz, E., Capón-García, E. , Laínez-Aguirre, J. M. , Espuña, A. and Puigjaner, L. (2014), 'Using mathematical knowledge management to support integrated decision-making in the enterprise', *Computers & Chemical Engineering* **6**6, 139–150. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135414000738*

Muñoz, E., Capón-García, E. , Laínez-Aguirre, J. M. , Espuña, A. and Puigjaner, L. (2015), 'Supply chain planning and scheduling integration using Lagrangian decomposition in a knowledge management environment', *Computers & Chemical Engineering* **7**2, 52–67. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135414001811*

Muñoz, E., Capón-García, E. , Laínez-Aguirre, J. M. , Espuña, A. and Puigjaner, L. (2017), 'Decision-Support System for Integration of Transactional Systems and Analytical Models in the Pharmaceutical Industry', *Trends and Applications in Software Engineering, Advances in Intelligent Systems and Computing* pp. 177–186. **URL:** *http://link.springer.com/10.1007/978-3-319-48523-2_17*

Muñoz, E., Capón-García, E. , Moreno-Benito, M. , Espuña, A. and Puigjaner, L. (2011), 'Scheduling and control decision-making under an integrated information environment', *Computers & Chemical Engineering* **3**5(5), 774–786. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135411000354*

Muñoz, E., Capón-García, E. and Puigjaner, L. (2017), 'ANSI/ISA 88-95 Standards Based-Approach for Improved Integration of Recipes and Operational Tasks Supported by Knowledge Management', *Computer Aided Chemical Engineering* **4**0, 2335–2340. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444639653503913*

Muñoz, E., Espuña, A. and Puigjaner, L. (2010), 'Towards an ontological infrastructure for chemical batch process management', *Computers & Chemical Engineering* **3**4(5), 668–682. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135409003160*

Musulin, E., Roda, F. and Basualdo, M. (2013), 'A knowledge-driven approach for process supervision in chemical plants', *Computers & Chemical Engineering* **5**9, 164–177. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135413002020*

Namur (2003), Requirements to be met by Systems for Recipe-Based Operations - NE 33, Technical report, Interessengemeinschaft Automatisierungstechnik der Prozessindustrie - User Association of Automation Technology in Process Industries.

National Research Council (1999), *Impact of Advances in Computing and Communications Technologies on Chemical Science and Technology*, National Academies Press, Washington, D.C. **URL:** *http://www.nap.edu/catalog/9591*

Navigli, R. and Velardi, P. (2004), 'Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites', *Computational Linguistics* **3**0(2), 151–179. **URL:** *http://www.mitpressjournals.org/doi/10.1162/089120104323093276*

Nelson, P. R. and Shull, R. S. (1997), 'Organizing for an initial implementation of S88', *ISA Transactions* **3**6(3), 189–195. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S001905789700013X*

Nie, Y., Biegler, L. T. and Wassick, J. M. (2012), 'Integrated scheduling and dynamic optimization of batch processes using state equipment networks', *AIChE Journal* **5**8(11), 3416–3432. **URL:** *http://doi.wiley.com/10.1002/aic.13738*

Nocedal, J. and Wright, S. J. (1999), *Numerical Optimization*, Springer.

Nof, S. Y., ed. (2009), *Springer Handbook of Automation*, Springer Berlin Heidelberg, Berlin, Heidelberg. **URL:** *http://ebooks.cambridge.org/ref/id/CBO9781107415324A009 http://link.springer.com/10.1007/978-3-540-78831-7*

Nortcliffe, A., Thompson, M. , Shaw, K. , Love, J. and Fleming, P. (2001), 'A framework for modelling in S88 constructs for scheduling purposes', *ISA Transactions* **4**0(3), 295–305. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0019057800000525*

Novas, J. M. and Henning, G. (2010), 'Reactive scheduling framework based on domain knowledge and constraint programming', *Computers & Chemical Engineering* **3**4(12), 2129–2148. **URL:** *http://dx.doi.org/10.1016/j.compchemeng.2010.07.011 http://linkinghub.elsevier.com/retrieve/pii/S0098135410002590*

Noy, N. F. and McGuinness, D. (2001), 'Ontology development 101: a guide to creating your first ontology'. **URL:** *http://protege.stanford.edu/publications/ontology_development/ontology101.pdf (last access May 2018)*

Object Management Group (2017), 'OMG Systems Modeling Language (OMG SysML)'. **URL:** *http://www.omg.org/spec/SysML/ (last access May 2018)*

OMG (2008), 'Software & Systems Process Engineering Meta-Model Specification', *Object Management Group Inc.* . **URL:** *http://www.omg.org/spec/SPEM/2.0/PDF (last access May 2018)*

OWL API (2018), 'OWL API'. **URL:** *http://owlapi.sourceforge.net/ (last access May 2018)*

Palmisano, I. and Olwapi, T. (2011), The Rough Guide to the OWL API : a tutorial, Technical report, Department of Computer Science University of Manchester. **URL:** *http://owlapi.sourceforge.net/owled2011_tutorial.pdf (last access May 2018)*

Pardo, C., Pino, F. J. , García, F. , Piattini, M. and Baldassarre, M. T. (2012), 'An ontology for the harmonization of multiple standards and models', *Computer Standards & Interfaces* **3**4(1), 48–59. **URL:** *http://dx.doi.org/10.1016/j.csi.2011.05.005 http://linkinghub.elsevier.com/retrieve/pii/S0920548911000675*

Park, J., Du, J. , Harjunkoski, I. and Baldea, M. (2014), 'Integration of Scheduling and Control Using Internal Coupling Models', *Computer Aided Chemical Engineering* **3**3, 529–534. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444634566500892*

Pauwels, P. and Terkaj, W. (2016), 'EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology', *Automation in Construction* **6**3, 100–133. **URL:** *http://dx.doi.org/10.1016/j.autcon.2015.12.003 http://linkinghub.elsevier.com/retrieve/pii/S0926580515002435*

Pistikopoulos, E. N. and Diangelakis, N. A. (2016), Towards the integration of process design, control and scheduling: Are we getting closer?, *in* 'Computers & Chemical Engineering', Vol. 91, pp. 85–92. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135415003609*

Pochet, Y. and Wolsey, L. a. (2006), *Production Planning by Mixed Integer Programming*, Springer Series in Operations Research and Financial Engineering, Springer New York. **URL:** *http://link.springer.com/10.1007/0-387-33477-7*

Protege (2015), 'Stanford Center for Biomedical Informatics Research (BMIR)', *Stanford University School of Medicine* .

PSE (2018), gProms ProcessBuilder, Technical report, PSE Enterprise. **URL:** *https://www.psenterprise.com/concepts/equation-oriented (last accessed May 2018)*

Pyomo (2018), ‘Pyomo’. **URL:** *http://www.pyomo.org/ (last access May 2018)*

Python (2018), ‘Python’. **URL:** *https://www.python.org/ (last access May 2018)*

Ra, M., Yoo, D. , No, S. , Shin, J. and Han, C. (2012), ‘The Mixed Ontology Building Methodology Using Database Information’, *Proceedings of the International MultiConference of Engineers and Computer Scientists* **I**. **URL:** *http://www.iaeng.org/publication/IMECS2012/IMECS2012_pp68-73.pdf*

Rawlings, B. J. B. (2000), ‘Industrial Process Control’, *IEEE Control Systems Magazine* **2**0(3), 38–52. **URL:** *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=845037*

Remolona, M. F. M., Conway, M. F. , Balasubramanian, S. , Fan, L. , Feng, Z. , Gu, T. , Kim, H. , Nirantar, P. M. , Panda, S. , Ranabothu, N. R. , Rastogi, N. and Venkatasubramanian, V. (2017), ‘Hybrid ontology-learning materials engineering system for pharmaceutical products: Multi-label entity recognition and concept detection’, *Computers & Chemical Engineering* . **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135417301382*

Roda, F. and Musulin, E. (2014), ‘An ontology-based framework to support intelligent data analysis of sensor measurements’, *Expert Systems with Applications* **4**1(17), 7914–7926. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0957417414003741*

Roth, A., Gerbaud, V. , Boix, M. and Montastruc, L. (2017), A Meta-ontology to Design Sustainable Project in a Competitive Stakeholder’s Context, *in* ‘Computer Aided Chemical Engineering’, Elsevier, pp. 1903–1908. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444639653503196*

Rubik, F. and Scholl, G. (2002), ‘Integrated Product Policy (IPP) in Europe - a development model and some impressions’, *Journal of Cleaner Production* **1**0(5), 507–515. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0959652602000161*

Sanchez, A., Rotstein, G. , Alsop, N. , Bromberg, J. , Gollain, C. , Sorensen, S. , Macchietto, S. and Jakeman, C. (2002), ‘Improving the development of event-driven control systems in the batch processing industry. A case study’, *ISA Transactions* **4**1(3), 343–363. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0019057807600937*

Sargent, R. (2005), ‘Process systems engineering: A retrospective view with questions for the future’, *Computers & Chemical Engineering* **2**9(6), 1237–1241. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135405000141*

Schaefer, J. J. (1996), ‘ISA-S88.01-1995 impact on design and implementation of batch control systems’, *ISA Transactions* **3**5(4), 351–356. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0019057896000407*

Scholten, B. (2007*a*), *The Road to Integration, A guide to Applying the ISA-95 Standard in Manufacturing*, ISA - Instrumentation, System, and Automation Society.

Scholten, B. (2007*b*), ‘Where batch ends and integration begins’. **URL:** *https://www.isa.org/ (last access December, 2016) https://goo.gl/Ux21Wb*

Seay, J. R. (2015), ‘Education for sustainability: Developing a taxonomy of the key principles for sustainable process and product design’, *Computers & Chemical Engineering* **8**1, 147–152. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135415000812*

Seborg, D. E., Edgar, T. F. and Mellichamp, D. A. (2004), *Process Dynamics and Control*, Wiley.

Shah, N. (2005), ‘Process industry supply chains: Advances and challenges’, *Computers & Chemical Engineering* **2**9(6), 1225–1235. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S009813540500013X*

Shah, N., Pantelides, C. and Sargent, R. (1993), 'A general algorithm for short-term scheduling of batch operations - II. Computational issues', *Computers & Chemical Engineering* **1**7(2), 229–244. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/009813549380016G*

Shobrys, D. E. and White, D. C. (2000), 'Planning, scheduling and control systems: why can they not work together', *Computers & Chemical Engineering* **2**4(2-7), 163–173. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135400005081*

Shokrian, M. and High, K. A. (2014), 'Application of a multi objective multi-leader particle swarm optimization algorithm on NLP and MINLP problems', *Computers & Chemical Engineering* **6**0, 57–75. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S009813541300241X*

Silvente, J., Kopanos, G. M. , Pistikopoulos, E. N. and Espuña, A. (2015), 'A rolling horizon optimization framework for the simultaneous energy supply and demand planning in microgrids', *Applied Energy* **1**55, 485–501. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0306261915007230*

Simon, H. A. (1960), *The new science of management decision*, Harper & Brothers Publishers, New York.

Simon, L. L., Osterwalder, N. , Fischer, U. and Hungerbühler, K. (2008), 'Systematic Retrofit Method for Chemical Batch Processes Using Indicators, Heuristics, and Process Models', *Industrial & Engineering Chemistry Research* **4**7(1), 66–80. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie070044h*

Stephanopoulos, G., Henning, G. and Leone, H. (1990*a*), 'MODEL.LA. A modeling language for process engineering - I. The formal framework', *Computers & Chemical Engineering* **1**4(8), 813–846. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/009813549087040V*

Stephanopoulos, G., Henning, G. and Leone, H. (1990*b*), 'MODEL.LA. A modeling language for process engineering - II. Multifaceted modeling of processing systems', *Computers & Chemical Engineering* **1**4(8), 847–869. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/009813549087041M*

Sürie, C. and Reuter, B. (2015), Supply Chain Analysis, *in* H.Stadtler and C.Kilger, eds, 'Supply chain management and advanced planning Concepts, Models, Software, and Case Studies', Springer Heidelberg, Heidelberg, chapter Supply Cha, pp. 29–54. **URL:** *http://link.springer.com/10.1007/978-3-540-74512-9*

Takamatsu, T. (1983), 'The nature and role of process systems engineering', *Computers & Chemical Engineering* **7**(4), 203–218. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/009813548380012X*

The International symposium on process systems engineering (1982), *in* 'PSE '82 : Kyoto', Kyoto.

Tolksdorf, G., Esche, E. , Wozny, G. and Repke, J.-U. (2017), Automatic Generation of Simulation Code for Embedding Custom Unit Operations in CAPE Software, *in* 'Computer Aided Chemical Engineering', Vol. 40, Elsevier, pp. 463–468. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444639653500799*

Trokanas, N., Bussemaker, M. , Velliou, E. , Tokos, H. and Cecelja, F. (2015), BiOnto: An Ontology for Biomass and Biorefining Technologies, *in* 'Computer Aided Chemical Engineering', Vol. 37, pp. 959–964. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B978044463577850005X*

Üney, F. and Türkay, M. (2006), 'A mixed-integer programming approach to multi-class data classification problem', *European Journal of Operational Research* **1**73(3), 910–920. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0377221705006909*

United Nations (2015), Transforming our world: the 2030 Agenda for Sustainable Development, Technical report, United Nations. **URL:** *https://sustainabledevelopment.un.org/post2015/transformingourworld (last access May 2018)*

UnitedNations (2007), Indicators of Sustainable Development: Guidelines and Methodologies, Technical report, United Nations, New York.

Upreti, S. R. (2004), ‘A new robust technique for optimal control of chemical engineering processes’, *Computers & Chemical Engineering* **2**8(8), 1325–1336. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135403002175*

Van Rossum, G. (1993), ‘An introduction to Python for UNIX/C programmers’, *Proceedings of the NLUUG najaarsconferentie* pp. 1–8. **URL:** *http://liuj.fcu.edu.tw/net_pg/python/Intro-Python.pdf (last access May 2018)*

Vegetti, M. and Henning, G. (2015), An Ontological Approach to Integration of Planning and Scheduling Activities in Batch Process Industries, *in* ‘Computer Aided Chemical Engineering’, Vol. 37, pp. 995–1000. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/B9780444635778500115*

Vegetti, M., Roldán, L. , Gonnet, S. , Leone, H. and Henning, G. (2016), ‘A framework to represent, capture, and trace ontology development processes’, *Engineering Applications of Artificial Intelligence* **5**6(February), 230–249. **URL:** *http://www.scopus.com/inward/record.url?eid=2-s2.0-84881407554&partnerID=tZOtx3y1 http://linkinghub.elsevier.com/retrieve/pii/S0952197616301762*

Venkatasubramanian, V. (2009*a*), ‘Cyberinfrastructure, informatics, and intelligent systems: Commentary on the new thematic area for Computers and Chemical engineering’, *Computers & Chemical Engineering* **3**3(1), 2–3. **URL:** *http://www.sciencedirect.com/science/article/pii/S0098135408001762*

Venkatasubramanian, V. (2009*b*), ‘DROWNING IN DATA: Informatics and modeling challenges in a data-rich networked world’, *AIChE Journal* **5**5(1), 2–8. **URL:** *http://doi.wiley.com/10.1002/aic.11756*

Venkatasubramanian, V., Zhao, C. , Joglekar, G. , Jain, A. , Hailemariam, L. , Suresh, P. , Akkisetty, P. , Morris, K. and Reklaitis, G. (2006), ‘Ontological informatics infrastructure for pharmaceutical product development and manufacturing’, *Computers & Chemical Engineering* **3**0(10-12), 1482–1496. **URL:** *http://linkinghub.elsevier.com/retrieve/pii/S0098135406001578*

Vivaldi, J. and Rodríguez, H. (2010*a*), ‘Finding Domain Terms using Wikipedia’, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC’10)* pp. 386–393. **URL:** *http://www.lrec-conf.org/proceedings/lrec2010/pdf/748_Paper.pdf*

Vivaldi, J. and Rodríguez, H. (2010*b*), ‘Using Wikipedia for term extraction in the biomedical domain: first experiences’, *Procesamiento del Lenguaje Natural* **4**5, 251–254. **URL:** *http://journal.sepln.org/index.php/pln/article/viewFile/809/663*

Vogel-Heuser, B., Fay, A. , Schaefer, I. and Tichy, M. (2015), ‘Evolution of software in automated production systems: Challenges and research directions’, *Journal of Systems and Software* **1**10, 54–84. **URL:** *http://dx.doi.org/10.1016/j.jss.2015.08.026 http://linkinghub.elsevier.com/retrieve/pii/S0164121215001818*

W3C (2018), ‘OWL 2 Web Ontology Language’. **URL:** *https://www.w3.org/TR/2012/REC-owl2-overview-20121211/ (last access May 2018)*

Whitley, D. and Rowe, J. (2008), Focused no free lunch theorems, *in* ‘Proceedings of the 10th annual conference on Genetic and evolutionary computation - GECCO ’08’, ACM Press, New York, New York, USA, p. 811. **URL:** *http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.7505&rep=rep1&type=pdf http://portal.acm.org/citation.cfm?doid=1389095.1389254*

Williams, H. P. (1978), *Model Building in Mathematical Programming*, 5th editio edn, Wiley.

Williams, T. J. (1994), *The Purdue enterprise reference architecture*, Vol. 24, ISA - Instrument Society of America. **URL:** *http://www.tandfonline.com/doi/abs/10.1080/09511928908944389 http://linkinghub.elsevier.com/retrieve/pii/0166361594900175*

Williams, T. J., ed. (1989), *A Reference Model for Computer Integrated Manufacturing (CIM): A Description from the Viewpoint of Industrial Automation*, Instrument Society of America. **URL:** *http://www.pera.net/Pera/who_cim_ref.html*

Wolsey, L. (1998), *Integer Programming*, Wiley, New York.

Xu, G. and Papageorgiou, L. (2009), 'A mixed integer optimisation model for data classification', *Computers & Industrial Engineering* **5**6(4), 1205–1215. **URL:** *http://discovery.ucl.ac.uk/59339/*

Ydstie, B. E. (2017), 'Distributed decision making in complex organizations: the adaptive enterprise, 2040 Visions of Process Systems Engineering'. **URL:** *http://stephanopoulos-symposium.mit.edu/ https://www.youtube.com/watch?v=cIeIkB9-p_8&list=PL1Z8XsQQiN4MUNGsCZSeoT_kr9J-loncC&index=4*

You, F. and Grossmann, I. E. (2009), 'Integrated multi-echelon supply chain design with inventories under uncertainty: MINLP models, computational strategies', *AIChE Journal* **7**(PART 1), 419–440. **URL:** *http://doi.wiley.com/10.1002/aic.12010*

Yu, L., Schüller, A. and Epple, U. (2014), 'On the engineering design for systematic integration of agent-orientation in industrial automation', *ISA Transactions* **5**3(5), 1404–1409. **URL:** *http://dx.doi.org/10.1016/j.isatra.2013.12.029 http://linkinghub.elsevier.com/retrieve/pii/S0019057813002395*

Yunes, T., Aron, I. D. and Hooker, J. N. (2010), 'An Integrated Solver for Optimization Problems', *Operations Research* **5**8(2), 342–356. **URL:** *http://pubsonline.informs.org/doi/abs/10.1287/opre.1090.0733*

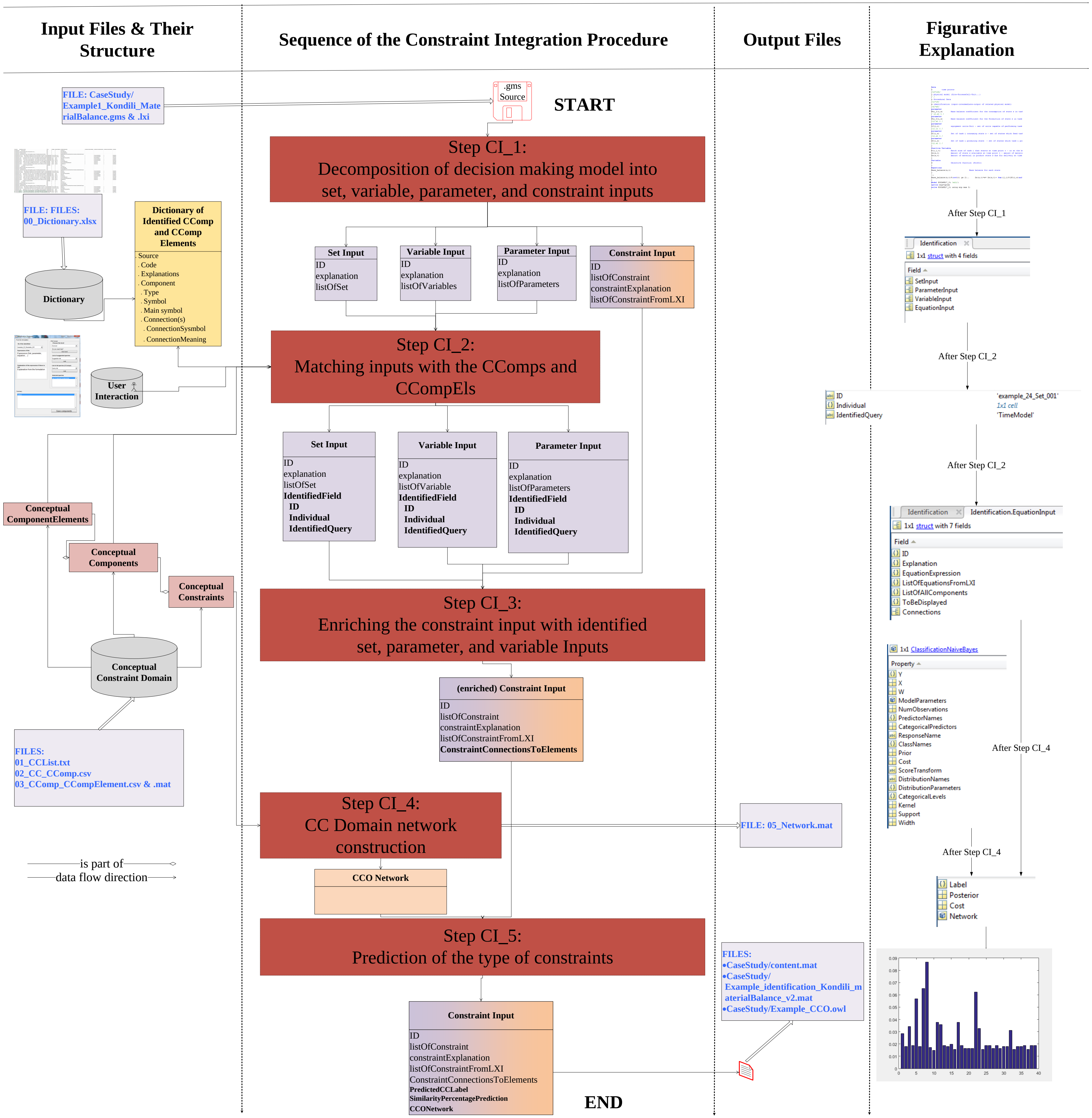
YWorks (2018), 'yEd Graph Editor'. **URL:** *https://www.yworks.com/ (last access May 2018)*

Zhang, G., Lu, J. and Gao, Y. (2015), *Multi-Level Decision Making*, Vol. 82 of *Intelligent Systems Reference Library*, Springer Berlin Heidelberg, Berlin, Heidelberg. **URL:** *http://link.springer.com/10.1007/978-3-662-46059-7*

Zhang, Q. and Grossmann, I. E. (2016), 'Enterprise-wide optimization for industrial demand side management: Fundamentals, advances, and perspectives', *Chemical Engineering Research and Design* **1**16, 114–131. **URL:** *http://dx.doi.org/10.1016/j.cherd.2016.10.006 http://linkinghub.elsevier.com/retrieve/pii/S0263876216303367*

Zhou, L., Pan, M. , Sikorski, J. J. , Garud, S. , Aditya, L. K. , Kleinelanghorst, M. J. , Karimi, I. A. and Kraft, M. (2017), 'Towards an ontological infrastructure for chemical process simulation and optimization in the context of eco-industrial parks', *Applied Energy* **2**04, 1284–1298. **URL:** *http://dx.doi.org/10.1016/j.apenergy.2017.05.002 http://linkinghub.elsevier.com/retrieve/pii/S0306261917304944*

Zhuge, J. and Ierapetritou, M. (2012), 'Integration of Scheduling and Control with Closed Loop Implementation', *Industrial & Engineering Chemistry Research* **5**1(25), 8550–8565. **URL:** *http://pubs.acs.org/doi/abs/10.1021/ie3002364 http://pubs.acs.org/doi/10.1021/ie3002364*


Input Files & Their Structure
Sequence of the Constraint Integration Procedure
Output Files
Figurative Explanation
FILE: CaseStudy/Example1_Kondili_MaterialBalance.gms & .lxi
.gms Source
START
Step CI_1: Decomposition of decision making model into set, variable, parameter, and constraint inputs
FILE: FILES: 00_Dictionary.xlsx
Dictionary of Identified CComp and CComp Elements
. Source
. Code
. Explanations
. Component
. Type
. Symbol
. Main symbol
. Connection(s)
. ConnectionSysmbol
. ConnectionMeaning
Dictionary
Set Input
ID
explanation
listOfSet
Variable Input
ID
explanation
listOfVariables
Parameter Input
ID
explanation
listOfParameters
Constraint Input
ID
listOfConstraint
constraintExplanation
listOfConstraintFromLXI
User Interaction
Step CI_2: Matching inputs with the CComps and CCompEls
Set Input
ID
explanation
listOfSet
IdentifiedField
ID
Individual
IdentifiedQuery
Variable Input
ID
explanation
listOfVariable
IdentifiedField
ID
Individual
IdentifiedQuery
Parameter Input
ID
explanation
listOfParameters
IdentifiedField
ID
Individual
IdentifiedQuery
Conceptual ComponentElements
Conceptual Components
Conceptual Constraints
Step CI_3: Enriching the constraint input with identified set, parameter, and variable Inputs
Conceptual Constraint Domain
(enriched) Constraint Input
ID
listOfConstraint
constraintExplanation
listOfConstraintFromLXI
ConstraintConnectionsToElements
FILES:
01_CCList.txt
02_CC_CComp.csv
03_CComp_CCompElement.csv & .mat
Step CI_4: CC Domain network construction
FILE: 05_Network.mat
CCO Network
is part of
data flow direction
Step CI_5: Prediction of the type of constraints
FILES:
•CaseStudy/content.mat
•CaseStudy/Example_identification_Kondili_materialBalance_v2.mat
•CaseStudy/Example_CCO.owl
Constraint Input
ID
listOfConstraint
constraintExplanation
listOfConstraintFromLXI
ConstraintConnectionsToElements
PredictedCCLabel
SimilarityPercentagePrediction
CCONetwork
END
After Step CI_1
Identification
1x1 struct with 4 fields
Field
SetInput
ParameterInput
VariableInput
EquationInput
After Step CI_2
ID 'example_24_Set_001'
Individual 1x1 cell
IdentifiedQuery 'TimeModel'
After Step CI_2
Identification
Identification.EquationInput
1x1 struct with 7 fields
Field
ID
Explanation
EquationExpression
ListOfEquationsFromLXI
ListOfAllComponents
ToBeDisplayed
Connections
1x1 ClassificationNaiveBayes
Property
Y
X
W
ModelParameters
NumObservations
PredictorNames
CategoricalPredictors
ResponseName
ClassNames
Prior
Cost
ScoreTransform
DistributionNames
DistributionParameters
CategoricalLevels
Kernel
Support
Width
After Step CI_4
After Step CI_4
Label
Posterior
Cost
Network
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
5
10
15
20
25
30
35
40