2023

Model Based Systems Engineering Approaches to Chemicals and Materials Manufacturing

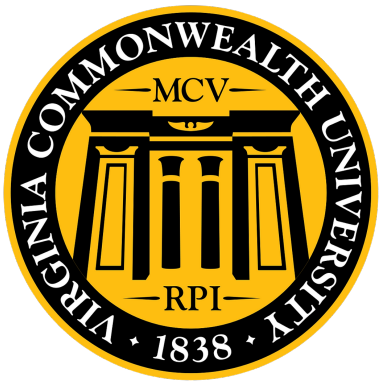
Quang Le

Model Based Systems Engineering Approaches to Chemicals and Materials Manufacturing

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science

by

Quang Le



Advisor: James K. Ferri, PhD

Chemical and Life Science Engineering

Virginia Commonwealth University

Richmond VA, USA

May 2023

Acknowledgement

This dissertation serves as a testament to the idea that the word "child" in the African proverb *It takes a village to raise a child* can be aptly replaced with the word "person".

First and foremost, I would like to express my deepest gratitude to my advisor and mentor, Dr. James Ferri. Your care, mentorship, and guidance have been pivotal to my success. Your honest feedback over the past three years has allowed me to better understand myself, my strengths and weaknesses, and has given me a glimpse of the person I aspire to become in the future. I will forever be grateful for your mentorship.

I would also like to extend my thanks to Dr. Tyler McQuade and Adam Luxon for welcoming me into the McQuade Lab during my freshman year and providing me with a strong foundation in critical thinking and problem-solving skills. I am especially grateful to Adam for introducing me to the world of computational science, which has been an essential component in all of my research projects since freshman year.

Special thanks to Dr. Frank Gupton and Bill Glandorf for agreeing to be part of my dissertation committee. I am grateful to Dr. Gupton for his teachings in Senior Design, which offered valuable insights into chemical and, particularly, pharmaceutical manufacturing. I would like to thank Bill Glandorf for teaching me everything I know about systems thinking and model-based systems engineering.

To my lab mates from the Ferri Lab, and friends from VCU, you have made my college experience an unforgettable one. I was only able to achieve my Bachelor, and now my Master's degree thanks to your friendship, patience and support.

Lastly, it is with absolute conviction that I express my strongest possible, most sincere gratitude towards my parents, Quyền and Dung. It is all thanks to their unwavering and endless support that I can be who I am today. I love you both.

Table of Contents

List of Tables

List of Figures

Abstract

Model-based systems engineering (MBSE) is part of a long-term trend toward model-centric approaches adopted by many engineering disciplines. This work establishes the need for an MBSE approach by reviewing the importance, complexity, and vulnerability of the U.S. chemical supply chains. The origins, work processes, modeling approaches, and supporting tools of the systems engineering discipline (SE) are discussed, along with the limitations of the current Process Systems Engineering (PSE) framework. The case is made for MBSE as a more generalizable and robust approach. Systems modeling strategies for MBSE are introduced, as well as a novel MBSE method that supports the automation tailored and extended to support the analysis of chemical supply chains. This work demonstrate the potential of MBSE approaches in chemical manufacturing by presenting two cases studies involving two different Active Pharmaceutical Ingredients (API), Atropine and Albuterol. The conclusion offers a prospectus on developmental opportunities for extracting greater benefit from MBSE in the design and management of chemical supply chains.

Keywords: *Model-Based Systems Engineering*, *Chemical Manufacturing*, *Supply Chain Resiliency*, *SysML*, *Metamodel*, *Ontology*

Introduction

Systems engineering (SE) as a discipline revolves around the design, analysis, and management of systems. As an engineering discipline, SE utilizes quantitative models, numerical methods, and statistical analyses. These tools allow systems engineers to optimize system performance and make data-driven decisions to ensure the system meets the needs and requirements of the stakeholders. The dimensions of these needs and requirements are diverse and circumspect, accommodating manufacturing, logistics, regulatory, legal, financial, geographic, and political requirements, and higher order analyses. More often than not, requirements are in conflict with each other. Therefore, engineers must make precise trade-offs to achieve the most favorable outcome. Ultimately, the goal of systems engineering as a discipline is to create a system that is reliable, efficient, and easy to maintain through the use of quantitative analysis and modeling techniques. When discussing SE approaches - without focusing on the usage of "models" (which are further discussed in subsection *Systems Architecture*) - we are typically referring to the document-based systems engineering (DBSE) approach. Traditionally, DBSE has been the default SE method, both in academia[1] and in industry[2]. It is mainly because DBSE and its tools are intuitive, easy to implement[3], and most notably, DBSE is evidently still a powerful method as it is practiced in almost every engineering-related industry, with the chemical industry as one of the few exceptions .

Process Systems Engineering (PSE) is a multidisciplinary field that was built on the SE discipline[4] and deals with the design, optimization, control, and management of chemical systems. The use of a model, and the model-based approach has become the standard practice in PSE[4]. Models in PSE research involves the design, optimization, analysis, control of PSE systems to ensure that PSE process operates safely and efficiently. However, PSE is still lacking in its ability to integrate and manage models and modeling tools that are used throughout the manufacturing lifecycle.

Model-based systems engineering is an SE approach that, similar to PSE, focuses on using

models to represent the systems' components and its behaviors, while facilitating communications and collaborations among stakeholders by using the systems model as the single source of truth. MBSE is expected to supplant the document-centric approach that has been practiced by systems engineers in the past by underpinning SE with a more rigorous mathematical foundation through the use of models, languages, and most importantly, systems architecture to provide a means to ensure consistency in model architecture, integration of simulations into design, maintain requirement traceability, and develop standardized artifacts which are both visually intuitive and machine-readable[5]. By leveraging transdisciplinary models to create rigorous, mathematical descriptions of the system of interest, MBSE provides multi-dimensional predictive capabilities, rigorous change management, and design evolution.

There are several objectives to this review. The first objective is to establish the need for MBSE in the design and implementation of resilient manufacturing supply chains by reviewing the importance, vulnerability, and complexity of U.S. chemical supply chains. The second objective is to substantiate that claim by reviewing the pedigree of contemporary SE work processes, modeling approaches, and supporting software tools and to provide an example of how these capabilities were applied to explicate the manufacturing supply chain of API. The third objective is to identify strategic research opportunities in the application of MBSE to advance the design of more resilient supply chains. Although the example provided in this thesis focuses on the manufacturing of pharmaceutical products, specifically APIs, this approach can be applied to chemical, materials, and other manufactured products more generally.

Chapter 1: U.S. Chemical Industry

The U.S. chemical industry is a world leader, with more than $200 billion in announced investment since 2010, supporting a vast domestic supply chain[6]. After decades of decline in domestic

manufacturing, the U.S. chemical industry is investing in domestic manufacturing due to the increased domestic production of oil and natural gas. This industry converts various raw materials into more than 70,000 diverse products that are essential to modern life. Several hundred thousand U.S. chemical facilities use, manufacture, store, or transport chemicals along a complex, global supply chain[7]. Facilities range from petrochemical manufacturers to chemical distributors. With $553 billion in shipments, the United States is the second-largest chemical producing nation after China[6]. End customers include critical infrastructure facilities in several other sectors, which makes the uninterrupted production and transportation of chemicals essential for national and economic security.

Supply Chain Complexity

The lifecycle of a system, natural or artificial, describes the evolution of the system from conception to retirement[8]. Figure 1A, reprinted from Marquardt et al.[9], describes a typical life cycle of a chemical product in an enterprise. For example, consider the oil and gas industry. In their "Perspective", Tavallali et al. describe the process of planning and development of oil fields[10]: an oil and gas project starts with the exploration and appraisal of hydrocarbon sources. Seismic studies are conducted to identify potential location and confirm the drillability (prospects) of hydrocarbon reservoirs. If oil is discovered at a prospect, "permeability, pressures, temperatures, fluids, and hydrocarbon saturation" data are collected for further testing. In addition, the data analysis focuses on well-understood geological properties specific to hydrocarbon such as fluid contacts and faulting. Construction of infrastructures and facilities on the approved oil fields are based on the analysis found during the appraisal process. Once the construction finishes, oil production starts and continues until the reserves are empty. Facilities are removed and the site is restored to its original conditions. Overall, the entire lifecycle from exploration to retirement can take up

to 52 years. A significant part of this lifecycle is on the exploration and appraisal of potential reservoirs. This is because the geophysical and petrophysical data gathered lay the foundation for the infrastructure, production and even retirement of the oil field [10]. Therefore, computational and simulation toolboxes are instrumental in the development of oil fields. Tavallali et al. explore mathematical algorithms and simulation software than were developed to optimize various aspects of oil field design and operation [10]. Marquardt et al. developed Figure 1A as a framework to organize these tools along the lifecycle of a chemical process.

Tavallali et al. demonstrated the complexity and commitment involved throughout the lifecycle of an oil field. We can also observe similar lifecycles (Figure 1A) throughout the chemical industry [12]. Furthermore, the inputs in the "chemical lifecycle" of one market sector is dependent on the output of another sector's. This interdependencies between industries form the chemical supply chain network presented in the right hand side of Figure 1. Figure 1B, adopted from the American Chemistry Council [6], provides a "bird's-eye view" of the chemical supply chain. Materials extracted from deposits of natural resources are processed to become organic and inorganic raw materials. These raw materials are transformed, in large quantities, into chemical building blocks that are required for more complex chemical transformations. These commodity chemicals are synthesized into more high-value, often proprietary, molecules that are required to make the final chemical product. Examples of chemical intermediates are specialty chemicals that are sold based on their performance [6]. These intermediates can either be further formulated to become finals products or considered in and of themselves the final products. In both cases, the outputs are placed on the market as manufactured goods, consumer products or exports. For example, the chemical lifecycle starts from the extraction of hydrocarbons from natural resources. These large molecules are further broken down into smaller, more useful molecules, such as benzene and toluene. Large amount of these commodities chemicals are converted into important stable intermediates, which are

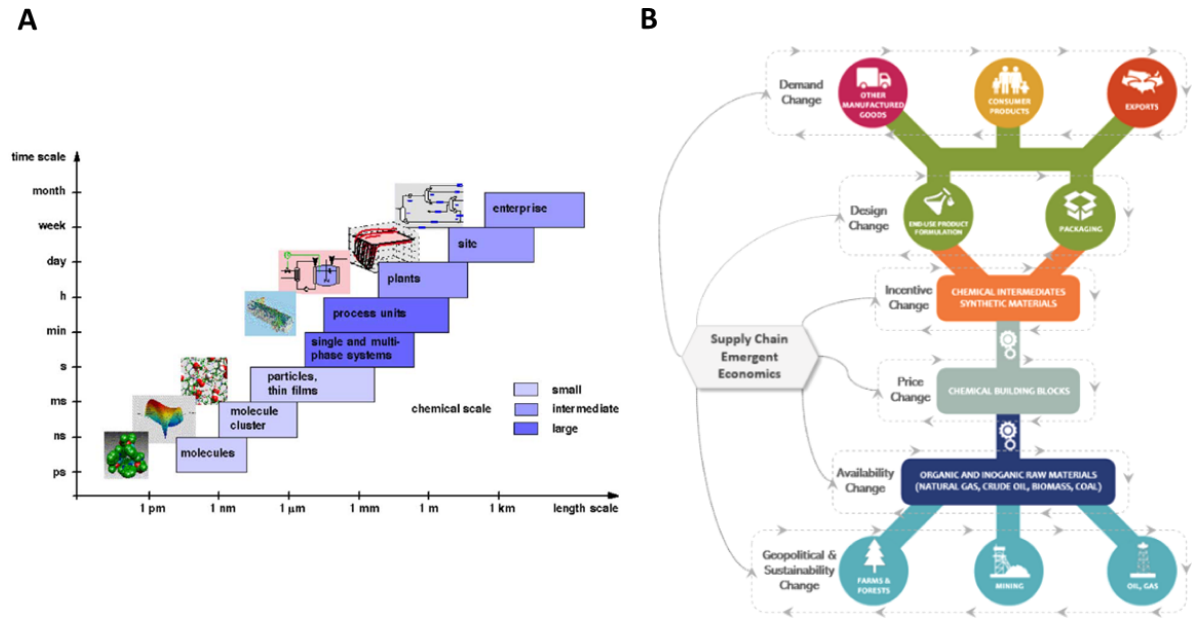


Figure 1: From Chemical Lifecycle to the Chemical Supply Chain. A. The "chemical lifecycle", reprinted from Marquardt et al.[9]. Every chemical transformation in the macro chemical supply chain is enabled by the same "chemical lifecycle". This system is decomposed into sub-systems that exist at different scale time, length and chemical. Each subsystem uses the previous one as input, creating a reverse "waterfall"[11]. The scale of time and length increases proportionally to the complexity of the system while the chemical scale increases from small to large to intermediate. This complex system is responsible for every transformations in the chemical supply chain. B. The "bird's-eye view" of the chemical supply chain, adopted from the American Chemistry Council[6]. Raw inputs "move up" the chemical supply chain through a series of transformations with the goal of becoming a product that can participate in the economy. Each layer is dependent on dynamic market factors (e.g. demand, design, incentive, price, availability, and geopolitical & sustainability). Before each transition is a chemical transformation which requires the previous transformations as input. A supply chain economic emerges to keep up with the complexity happening throughout the chemical supply chain.

key starting materials in the production of specialty chemicals or active pharmaceutical ingredients (APIs). APIs are incorporated into of various dose forms and packaged into products that enable delivery of the desired therapeutic effect. Figure 1B contrasts the manufacturing lifecyle of the

petrochemical industry with the transformation value chain which stretches from natural resources to consumer products. Although more than 95% of consumer products depend on the business of chemistry, the chemical supply chain is a complex and entangled network spanning disciplines, sectors, nations and other supply chains. Disruption to the chemical supply chain can produce domino effects that impacts more than 25% of the U.S. gross domestic product (GDP). This create opportunities for model-based systems engineering approaches to emerge and create solutions to manage this complexity.

Vulnerability of the Chemical Supply Chain Is Increasing

Despite its strategic importance to the United States economy, the business of chemistry and chemicals and materials manufacturing is an unsecured and vulnerable national asset. The U.S. chemical industry relies on offshore manufacturing due to many factors, including additional domestic regulations and their associated costs, increased labor and raw materials costs, and the lack of synergy among multiple regulations that affect the ability to maintain profitability within the United States. During the last thirty years of fine chemicals and pharmaceutical manufacturing in the United States, many key raw materials and/or intermediates have been outsourced to contract manufacturers in Asia and the South Asian peninsula. Over 80% of the medicines taken in the US today are sourced overseas, predominantly in China and India. More than 70% of the U.S. Food and Drug Administration (FDA)-registered API manufacturing facilities are located outside of the U.S[13]. Government priorities focus on identifying the countries with which the United States is developing dependencies regarding chemical supply chain issues and understanding the impacts and repercussions these trends might have on national security. Further emphasis has been placed on research activities to develop analysis and decision support systems that help model the cascading effects of attacks on chemical infrastructure, more accurate assessment of potential consequences,

and an informed allocation of resources[7]. In light of the stresses imposed by the COVID-19 pandemic, critical vulnerabilities in the chemical manufacturing sector have prompted a reconsideration of supply chain strategies for enhanced resilience to disruptions and stressors[14]. In 2019, the U.S. Food and Drug Administration, in partnership with the Department of Health and Human Services (Centers for Medicare and Medicaid Services and the Office of the Assistant Secretary for Preparedness and Response), the Department of Defense, and the Federal Trade Commission reported on the root causes of drug shortages[15]. A key finding of this commission was that drug shortages persist because they do not resolve according to the generally anticipated market responses associated with supply and demand due to the following dynamics. Cost savings measures and "lean" operations have been pursued at the expense of manufacturing and supply chain resilience. Typical markets would respond to a shortage by increasing production, but logistical challenges, especially the complexity of the supply chain, can limit the ability of drug manufacturers to increase production. In the 21st century, the drug supply chain has become longer, more complex, and fragmented. Many chemical products have essentially become federated supply chain from a collection of geographically disparate suppliers. The COVID-19 pandemic has underscored the need for and increased attention on advanced manufacturing to become more responsive. Unfortunately, much of the United States chemical manufacturing capacity is unable to be readily leveraged to address anticipated shortfalls, because key policy and manufacturing decision makers rely on intelligence from local subject matter experts and knowledge embedded in digital (and paper!) artifacts to track/document domestic production capability[15]. Therefore, manufacturing requirements and their accompanying solutions are evaluated in a highly non-systematic way through ad hoc exercises that are repeated for each change in raw materials availability or manufacturer capacity. Manufacturing and supply chain resilience cannot be achieved without first understanding and mapping the key processes, systems, and resources involved in the production of necessary chemicals.

The call to reshore manufacturing is heard everywhere for multiple reasons; creation of good jobs, continued economic growth and well-being are among the most prominent. Despite the wish to bring manufacturing back to the U.S., it is unlikely to occur except in industries where advanced manufacturing techniques can be developed and deployed. Here, we suggest that through a combination of new process technologies accompanied by MBSE and advanced digital tooling, reshoring pharmaceutical production is feasible - realizing a strategic goal that has existed for years[16].

Chapter 2: Systems Engineering As A Discipline

To understand how MBSE can help address supply chain vulnerability, we must understand the underlying discipline of SE. In this section, we provide the history of the SE practice beginning with a definition, the associated organizations and best practices, its state of adoption, and the various tools that allow for the transition from document-based to model-based practices.

What is Systems Engineering?

Systems engineering as a discipline draws from a constellation of other disciplines loosely described as "system science", including: general systems theory[17;18;19], cybernetics[20;21;22], systems dynamics[23;24], and systems thinking[25;26]. Systems engineering is a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods[27]. In the contemporary framework, it is noted that the terms 'engineering' and 'engineered' in their widest sense: "the action of working artfully to bring something about"[27]. Gibson et al. describe the precursors of SE stemming from the challenges of World War II, where mathematicians and physicists were called upon to contribute to the war effort[28]. In particular, they describe SE as having

parent disciplines of industrial management, control theory, operations research, and econometrics. While Buede (2000) identifies the first use of the term "systems engineering" dating to the 1940's by Bell Telephone Laboratories, it was RAND Corporation who played a key role in developing SE as a unique discipline[29]. Since the inception of SE as a discipline, many consortiums have formed to create and accumulate knowledge in the areas of systems analysis, best practices, and specific SE domains (e.g., Cybersecurity, Aerospace, Automotive, etc.). Today, the Department of Defense (DoD) defines SE as "a methodical and disciplined approach for the specification, design, development, realization, technical management, operations and retirement of a system" [30]. It further specializes this notion for mission engineering, "the deliberate planning, analyzing, organizing, and integrating of current and emerging operational and system capabilities to achieve desired warfighter mission effects" [31]. Today, SE stands as a distinct, although interdisciplinary, field with a number of key organizations and professional societies which refined its ongoing practice.

Adoption

Systems engineering practice, either document- or model-based, is used across a number of industries and involves the process of integrating a system specifications throughout its life cycle. This supports and formalizes activities such as requirements specification, trade space analysis, design, system analysis, verification and validation[32]. SE applications span across diverse industry sectors, including: aerospace and avionics[33;34;35;36;37], automotive engineering[38;39;40], biochemical engineering[41;42;43], civil engineering and transportation[44;45], cybersecurity[46;47], defense[48;49;50], disaster management[51;52;53], energy grids[54;55;56], product manufacturing[57;58;59], chemical engineering[60;61;62;63;64], and nuclear engineering[65;66]. However, industries like aerospace, energy, and automotive integrate SE practices comparatively more than industries like consumer electronics, healthcare, and construction[67].

Tools

Many of the core conventions of SE practices, such as "block diagrams," are adopted from other disciplines and substantially predate the addition of "Model-based" to the SE state of the art. The transition from creating diagrams in basic drafting tools (e.g., Visio) to tools that have at least some understanding of what types and relationships work together is, essentially, the transition to MBSE. To facilitate this, INCOSE maintains a database of Systems Engineering Tools[68].

Systems Modeling Language (SysML) (Figure 3) is the language of much, although by no means all, of MBSE. SysML is derived from UML, and so too are many of the most prominent MBSE tools extensions of UML tools[69;70;71]. Large governmental and industrial buyers drove the adoption of SysML[34;30]. Beyond SysML, several competing languages exist in the space, the most prominent of which is the Arcadia method using Capella[72;73]. French engineering goliath Thales created Capella[72] and required its use by vendors in its orbit. Regardless of the relative technical virtues of SysML and Arcadia, the forcing functions for SysML adoption have broader reach and deeper pockets.

Systems Engineering Work Processes

Systems engineers are responsible for the realization, use, and retirement of engineered systems[27]. NASA defined a system as the organization of elements that "function together to produce the capability required to meet a need" [74]. These elements are all things related to the development of the system such as infrastructures, facilities, software, hardware, data, etc. Systems engineers utilize established guidelines to address the needs to organize this body of knowledge. This section discusses the standards that are widely used in industry: The ISO/IEC/IEEE 24748-1:2018(E) which explains the life cycle stages in system development[8], and the V-model as the de facto framework to manage the life cycle stages in systems engineering[75].

Life Cycle Stages

All systems, regardless of properties, evolve from conceptualization to retirement. Therefore, it is easier to manage the evolution of a system-of-interest (SOI) if its progression is standardized. Such standards are called life cycle frameworks. The discipline of SE follows guidelines created by the International Standard Organization, a nongovernmental and independent organization that develops and publishes international standards for various technical and non-technical fields [76]. In 2018, ISO released the first edition of ISO/IEC/IEEE 24748-1:2018(E) (hereafter referred to as ISO 24748), which establishes the concept of a life cycle, define the evolutionary stages involved and most importantly, provide guidelines for life cycle management [8]. In this section, we provide a summary of the life cycle stages defined by ISO 24748.

According to ISO 24748, a typically life cycle consists of six main stages: concept, development, production, utilization, support and retirement [8]. Throughout this life cycle, the systems engineer must assure that "domain experts are involved, that all advantageous opportunities are pursued, and that all significant risks are identified and mitigated" at every stage [77].

The life cycle of a project starts with the realization that there is a need for a new technology or modification to an existing one. This recognition starts the preliminary phase where high-level studies are conducted to identify alternate concepts that meet the identified need. In the selection stage, all potential candidates are further investigated and compared via simulations and mockups. The emerging system-of-interest (SOI) should be the most feasible in terms of technicality, economic and time. The project then moves to development where the requirements, architecture and design of the SOI is defined via building prototypes. This allows accurate documentations of all resources and constraints needed for the SOI to be produced, utilized, supported and retired [8]. Production of the SOI starts once all identified resources are approved and acquired. As production continues, modifications of the product or infrastructure are expected to resolve conflicts, reduce cost or

increase cycle time[8]. All changes to the configurations and requirements of the SOI should be audited and approved before resuming production. The finished SOI usually first operates in a simulated environment where tests and validations are conducted to analyze performance or to look for anomalies. The product that meets the standard are delivered to its intended operating environment. However, it is expected to provide continuous support to the SOI while in service. This support usually comes in the form of upgrades to improve performance and modifications to address anomalies while maintaining the its initial purpose. Once the SOI is no longer needed, it is retired and removed from the operating environment[8].

While we can intuitively understand the life cycle stages, it is not so obvious on how the life cycle should be structured and what role does systems engineers play throughout the life cycle. In 1970, Winston Royce developed the Waterfall framework, the first ever method that defined the structure of a life cycle, including the input, output and the activities happening in every stage[11]. The Waterfall method inspired the creation of various other frameworks that are utilized in different fields that deal with complex system developments. While there is no "one size fits all", the technical nature of the SOI may make one framework more appropriate than others. These properties fall into three major categories: pre-specified and sequential, evolutionary and concurrent, and interpersonal and emergent[78]. In pharmaceutical manufacturing, chemical transformations are pre-specified by the chosen multi-step synthetic route. Therefore, the V-model , which is often used for pre-specified and sequential processes, is our recommended framework of choice to manage chemical manufacturing project cycle.

V-Model

Before 1991, various methods, such as the waterfall and the spiral, were created to manage the development cycles of large and complex software projects. Kevin Forsberg and Harold Mooz

noticed that these methods at the time did not specify the role of the systems engineer and the technical team at any stage of the cycle. This problem made these frameworks impractical for implementation in real project development[79]. To tackle this issue, Forsberg and Mooz released their peer-reviewed paper, "The Relationship of System Engineering to the Project Cycle", which introduced the "V-model"[79]. Over the years, the V-model has become the standard way to explain the role of systems engineers in project development[75;74]. Figure 2, is an adaptation of the original model where the role of systems engineers and methods are incorporated into the life cycle stages.

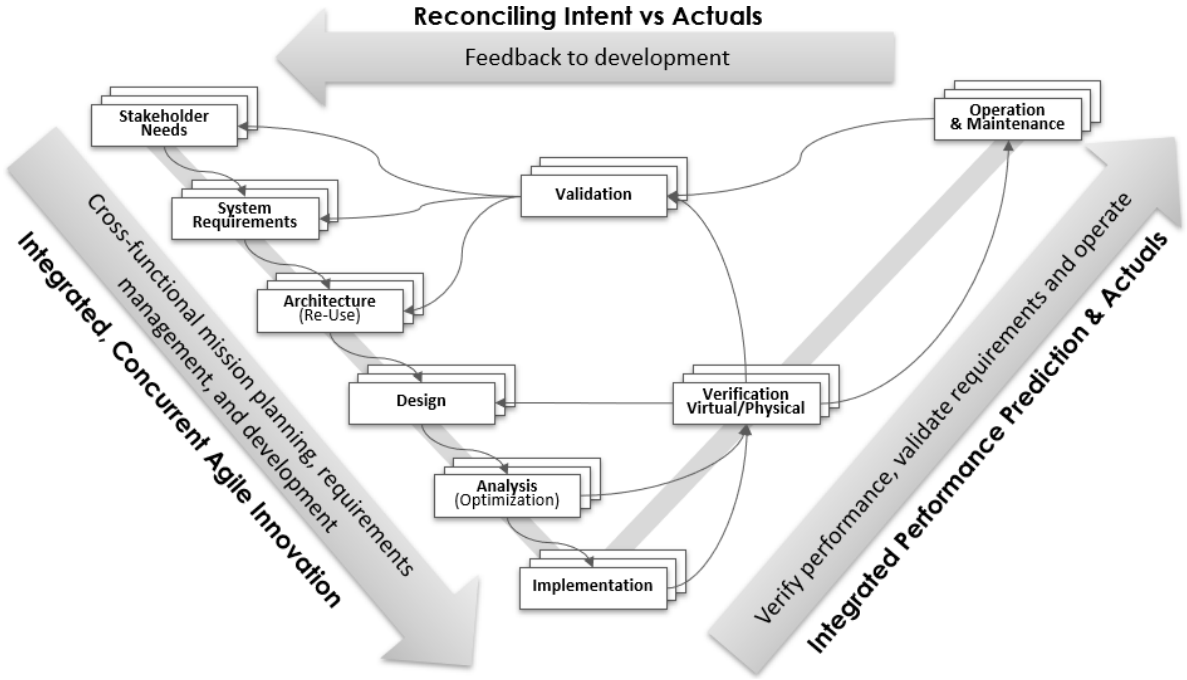


Figure 2: The V-model. Adapted from Forsberg and Mooz 1991[79], this V-model suggests a feedback mechanism from system verification to system design through validation methods. During system design, development frameworks such as Agile[80] is incorporated to promote rapid and iterative design while ensuring proper requirements management. After system implementation, analysis and optimization tools are integrated to verify and improve systems performance. Once the system is in operation, performance review data is fed back to the design process for future improvements and design.

This V-model profile can be divided into three major parts: the left wing of the "V" covers

the concept stage and the development stage, the right wing of the "V" holds the utilization and support stage, and the bottom of the V lies the development stage (left-to-right) as well as the connection between the other two axes (right-to-left). The method starts at the top of the left wing where stakeholder needs are identified. The definition of the system progresses from just an idea to detailed optimization of system design. Cross-disciplinary experience is required to integrate siloed work process and data throughout project development. Here, systems engineers are responsible for creating a cross-functional plan while managing the evolving requirements as the system progresses. Due to the iterative nature of this process, Agile[80] is the recommended method as it allows teams to quickly respond to changes by continuously evaluating requirements, plans and results[80]. While systems engineers do not usually contribute directly to implementation/production of the SOI, they are still responsible for assessing and documenting any changes to system requirements. After implementation, the SOI goes through verification and validation processes that are planned during system design. On this axis, systems engineers are responsible for verifying system performance, validating requirements from the development phase, and ensuring that the SOI delivers the intended services. This role can be supported by integrating simulation and data analytics to provide decision support capabilities. Data gathered from the validation process, which continues even during operation and maintenance, are used as feedback to future projects or to reconfigure the current one. Systems engineers often use the V-model to organize standardized bodies of knowledge, such as ISO/IEC/IEEE 15288:2015 to supplement their ability to manage life cycles within their organization[81].

Process Systems Engineering

In the chemical industry, process systems engineering (PSE) is the default and all-encompassing discipline when working with chemical systems. This calls for a high-level review of PSE, its current

state of the art and opportunities for SE to provide values as a discipline.

What is Process Systems Engineering?

The definition of process systems engineering (PSE), often attributed to Takeichiro Takamatsu[4;82;83], is simply a discipline that contains a set of methodologies required to make chemical engineering decisions[84]. According to Takamatsu, there must be methodologies to address the need for planning[85;86;87], design[88;89;90], operate[91;92], and control[93;94;95] of "any kind of unit operation, chemical and other production processes and chemical industries themselves"[84]. Since 1945, the advancement of computing power has allowed for the development of numerical methods and statistical analysis[83], which extends PSE practices to include methodologies for optimization[96;97;98] and simulation[99;100;101] of chemical systems. Furthermore, as suggested by Klatt and Marquardt, PSE has truly evolved into a field that include models as part of the core practice, also known as the model-based methods[4]. Finally, Takamatsu's definition allows the scope of PSE to encompass other sectors such as energy[101;102;103], petrochemical[10;104], biomedical[105], environmental[106;104] and even steel[107].

Model-based Process Systems Engineering

In this section, we will use the "chemical lifecycle" (Figure 1B) as the framework to explore the model-based approach in PSE or model-based PSE (MBPSE). Hereinafter, the "molecules", "molecule cluster" and "particles, thin films" level will be called the "molecular level", the "single and multi-phase systems" and "process units" level will be called the "process level", and finally the "plants", "site" and "enterprise" level will be called the "enterprise level". We will provide a high-level overview of the mathematical models, conceptual models, and tools that are applicable at different levels of the "chemical lifecycle" starting from the "molecular level".

Mathematical models can be categorized into two groups: data-driven models (DDMs) and non-data driven models (NDDMs). DDMs are statistical methods in which the parameters are optimized based on given data whereas NDDMs are mechanistic methods based on math, physics, and chemistry. NDDMs are often used for optimization and control (deductive) while DDMs are more suitable for prediction and improvement (inductive) . In PSE, non-linear NDDMs, such as mixed-integer non-linear programming (MINLP), are utilized at every level due to their ability to be customized with domain-specific knowledge. For an in-depth and comprehensive overview of all commonly used NDDMs in PSE, readers are referred to the work of Biegler and Grossmann[98]. PSE research at molecular level fall under the guise of "computer-aided molecular design" (CAMD), which optimizes chemical synthesis using molecular properties. Application of NDDMs in CAMD includes: solvent selection[108;109;110], design of molecules[88;111;112], property predictions[113;114;115], and fluid dynamics[116;117]. At the process level, process intensification (PI) is the all-encompassing topic that covers every aspect of chemical operations. PI seeks to reduce the cost of energy consumption, operation expenditure and cycle time by improving industrial processes. PI research utilizes simulation optimization models to improve existing processes, develop novel processes, design more specialized and cost-effective equipment, and advanced control systems throughout the plant. Tian et al. provide a comprehensive overview on the state of the art in PI models (NDDMs and DDMs) and tools that are applicable across multiple industry[118]. For process level reviews that focuses on pharmaceutical manufacturing, readers are encouraged to read the work by Buchholz[119] and Troup and Georgakis[93]. According to Grossmann[120], mathematical models at the enterprise level are developed to optimize: scheduling[86;121], infrastructure design (physical[122;123;124] and digital[125;126;127]), and supply chain management (SCM)[128;129;130;87;131]. As the industry towards PSE 4.0[132], DDMs, especially the use machine learning (ML) models, has become more and more important in PSE research. As machine learning (ML) promises a general, inductive model that requires

no knowledge in physics/chemistry, it can be deployed at any stage of the "chemical lifecycle". Specific development and application of ML and other DDMs have been reviewed extensively in the PSE community[133;132;134;135]. Finally, hybrid modeling, which is the implementation of NDDMs and DDMs in tandem, is encouraged as both DDMs and NDDMs play two different, but equally important roles in MBPSE[136;132].

Conceptual models provide guidance on how to use models more effectively. Conveniently, the "Life Cycle Assessment" (LCA) is a commonly used conceptual model in PSE[137]. Since LCA provides a holistic view of the system's development, its impact in PSE research, especially at the enterprise level, is well-documented[129;128;138]. The 2020 Annual Review of Chemical and Biomolecular Engineering provided an extensive review on how LCA can be applied on "Chemical Processes, Products, and Supply Chains"[12]. Furthermore, LCA serves as the basis of other conceptual models that industry uses to make business decisions that have lasting impact on profitability as well as long-term shareholder values. "Sustainability" is a prime example of such concept[139]. "Sustainability" describes the consideration that profit-seeking businesses should have towards external environments. Specifically, these considerations must promote business operations that bring positive, or minimal to no negative social, economic, and environmental impact. Starting from raw materials, to transformation operations, to energy usage or waste disposal, LCA allows companies to quickly and accurately identify specific parts in their production cycle that may bring adverse impact to the external environment. Furthermore, LCA provides a foundation to apply "sustainability" to the design and implementation of other complex systems[140], such as the supply chain of a company[12]. Finally, LCA is also a tool to effectively represent, organize, reuse, and update knowledge within an organization[141]. An ontology is a conceptual model that formally organizes a body of knowledge. Ontologies are often used to map the flow of information in complex systems that are composed of multiple complex subsystems that are dependent on another. Examples of

systems that have benefited from ontologies are infrastructure of information technology[125;126;128], process control[142] and supply chain[128]. OntoCAPE, which has been in development since the early 2000s, is the most widely used ontology in PSE[141] and has been applied in several modeling and design applications[143;144].

Modeling tools in PSE research come in many forms. Conceptual models, such as LCA and ontologies, can be easily created using any design application, such as Microsoft Powerpoint, or simply drawn on a sheet of paper. DDMs usually exist in free, open-source, and reliable software such as Python and Pytorch for machine learning development[145]. However, NDDMs are often proprietary software packages or applications that vary greatly in scale, format, price, accessibility (easy to use) and robustness. Tian et al. provided a comprehensive summary of the common software and applications that span the entirety of the "chemical supply system" (Figure 1B).

Challenges to Model-Based Process Systems Engineering

In the previous section, we explored the importance of models in PSE by showing the proliferation of PSE models across the "chemical lifecycle". Although there are frameworks, such as LCA and SCM, that allow industry to organize models and processes to work towards the same set of values, such as "sustainability", at every product developmental stage, there is insufficient PSE research in the integration of information and operations between different models across the lifecycle. While there exists a plethora of models at every spectrum in terms of pricing, accessibility (easy to use), and fidelity, it is unrealistic for the majority of the chemical industry to even move toward, let alone adopt, the end-to-end, data-driven methodologies of PSE 4.0[132]. We can explore this proposition via the review of PI by Tian et al.[118]. Tian et al. provided an extensive summary of the different studies in PI equipment (microreactors, rotating packed bed, structured reactors, et al.) and PI methods (membrane absorption, dividing wall column, process

control, et al.) using various models and software packages in multiple tables. Although this is an instructive introduction to process intensification, it is unclear on how to design a process that uses multiple different process-intensified equipment that support one or more process-intensified methods. Every PI method and technique use different optimization algorithms and were modeled using different software suites. In addition, it is difficult to constantly adapt existing control systems to new equipment and methodologies. Therefore, industry is often reluctant to adopt new technologies from academic research as there is often less risk and investment in using the current technology. In their article "Planning, scheduling, and control systems: why can they not work together", Shobrys and White[146] provided two key takeaways about the importance of integration. Firstly, effective integration of systems is difficult to achieve. Enterprise level systems such as project planning or supply chain are composed of multiple dynamic subsystems that are connected by different sets of tools and requirements and require update at different time scales. Marquardt has published multiple works showing how ontologies can be used to design complex physical and digital infrastructures[144;125]. However, Marquardt and Schneider also acknowledged that even with the ontology, it is unrealistic for tools that have different interfaces, have inputs and outputs of different syntax, and require different area of expertise to interoperate at large scale[125]. Furthermore, OntoCAPE, while being the most widely used ontology in PSE[141], has not been extensively tested in PSE research, although researchers use OntoCAPE as the basis to develop ontologies of their own[147;148]. The second key takeaway from Shobrys and White[146] is that systems integration is not a technical issue, but an organizational one. They revealed the communication gaps, or "seams", often form between departments within a company[146]. At every gap, the business loses momentum and fidelity as there is a need to repeat and re-translate information. This problem incentives departments to have more local objectives, to "dump" down the information which produces misunderstanding and inconsistency and ultimately slows down the decision-making process. To solve

this problem, we are reminded of the concept of "Enterprise Modeling" proposed by Lindheim et al. in 1996[149]. The Enterprise Modeling methodology places utmost important in constructing a common reality inside the enterprise, between developers, managers and stakeholders. Construction of this shared understanding is the Level 1 of Enterprise Modeling. While Level 1 only results in a simple design, this output acts as a foundation for future work as it maps every interaction, requirements and dependencies required in the system. Only when Level 1 is finished does the project move to developing (Level 2) and deploying (Level 3) the system. Lindheim et al. suggests that for Level 1 "Enterprise Modeling" to be truly complete, it must involve every discipline in the manufacturing process. The completion of Level 1 makes communication and decision-making much more effective as everyone is working towards a shared vision of the final product. This core understanding will result in the integration of analytical and optimization tools in Level 2 as everyone is aware of the inputs and outputs throughout the system life cycle[149]. At the time of this paper, Level 3 "Enterprise Modeling", deployment and activation of models are still a relatively new are of research. However, Level 1 Modeling requires the cooperation of every discipline in the process plant, including process control, scheduling and supply chain operations. Therefore, by following this method, we are one step closer to Grossmann's goal of "enterprise-wide optimization" (EWO), which "involves the optimization of the operations of supply, manufacturing (batch or continuous) and distribution in a company"[120].

Opportunities for MBSE

Model-based Systems Engineering is the SE methodology that addresses both of these issues in PSE. The goal of MBSE is to provide the enterprise - developers, managers and stakeholders - a single source of authoritative knowledge that spans the system lifecycle. This single source of truth can be designed using the same model elements throughout. Automation and optimization tools

can then be built underneath these model elements, allowing the models to be integrated with each other while working towards a common goal. MBSE provides the necessary components to create digital twins that can keep up with the increasing growth in complexity of the "chemical supply chain" (Figure 1A).

Model-Based Systems Engineering

In earlier discussions, we demonstrated that MBSE represents a significant departure from both DBSE and the current model-based approach in PSE. In this section, we will delve deeper into the value of MBSE by exploring its roots, the role of "models" and modeling, the modeling infrastructure in the form of languages, the crucial aspect of systems architecture that underpins MBSE, and the current reductionist perspective that is central to MBSE methodologies.

What is Model-Based Systems Engineering?

According to INCOSE, "Model-based systems engineering is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases." [150]. At the heart of MBSE are models which form the basis of the design process and artifact generation [151]. As systems (and systems of systems) become more complex and modular [38;152], the need to manage large amounts of information, stakeholders, and associated documentation becomes increasingly important [5;153;38;153;152;5]. Dickerson and Marvis provided a brief history of MBSE, with contributions from fields like mathematics and computer science [154]. Wymore, one of the early pioneers of MBSE, defined six categories of requirements: input/output, technology, performance, cost, trade-off, and system test, and provided the early mathematical foundation for MBSE [155]. Today, MBSE covers a wide array of modeling languages with associated guidance

documents and standards, described in more detail in section 5.2[156]. A benefit of MBSE is that complementary modeling techniques can be integrated, such as simulation[157;158], agent based modeling[159], decision analysis[160;161;162], failure modes and effects analysis[33], virtual reality[163], and digital twins[164;165].

Models and Modeling Languages in Systems Engineering

Models are ubiquitous in engineering. Models are simplified representations of real-world systems, which either exist or have yet to be realized, and therefore can be used to describe existing systems and to aid the design of how a system could or should be[152]. While Gass described modeling as the process of translating a problem into the language of mathematics to examine a problem[166], not all models are mathematical in nature - simple block diagrams are models as well as mathematical representations and simulations[166;167]. In SE approaches, and especially MBSE, models are used to support the system development life cycle, commonly represented by the waterfall model, spiral model, or V-model[156]. MBSE modeling approaches provide graphical representations and semantic representations. Graphical representations are used to communicate shared understandings among stakeholders, while semantic representations are used to support computations based on the rules governing the interaction between model components[32;32;156]. Models are necessarily incomplete, intentionally omitting elements, relationships, and distinctions that are not relevant to the particular purpose of the model and the modeler[168].

Models are also used to describe a subject of interest in a way that is useful for a specific purpose. As this task is extremely open-ended, most models are built as extensions of a set of baseline concepts that offer some guidance on how to approach the problem. Most commonly, these conceptual baselines take the form of metamodels or ontologies[32] (expanded in the next section). Building on top of the conceptual foundation is the medium of expressing a model.

This medium is called a language, and it may be textual (e.g. Clash)[169], graphical (e.g. Systems Modeling Language or SysML)[170], or a combination of these and other forms. Regardless, modeling languages let practitioners to build models using the same set of model elements throughout the design process. Most modeling languages support the creation of digital twins, which incentive MBSE modeling software, such as IBM Rhapsody[70] or Dassault Systèmes No Magic, to integrate or allow custom automation and optimization capabilities in their toolkit.

One of the earliest modeling languages is the Integration Definition (IDEF) family of languages, originating in the 1970[171]. Within the IDEF family, some commonly used languages include IDEF0 and IDEF3 (business function modeling), and IDEF1X (data modeling)[172]. The Unified Modeling Language (UML) was developed to describe aspects of software systems, and is the *de facto* standard for software development, but has been extended to include and describe business processes[154;173]. Today one of the most widely used languages in SE modeling is SysML[174], which was created by SysML Partners, a consortium of software companies and developers who organized to create SysML in 2003[175]. SysML extends UML and supports SE activities including specification, analysis, design, verification, and validation of systems through a number of diagrams describing requirements, behaviors, structures, and parametrics[154;176]. These four components are known as the four pillars of SysML[177] (Figure 3). However, SysML has inherited many of the weakness and esoteric complexities of UML. An RFP has been issued for a successor language, and a proposal is expected in the fourth quarter of 2022[178]. The proposed SysML v2 simplifies and unifies many problematic aspects of SysML, along with providing textual syntax and applications programming interfaces (API) specifications[179]. Other modeling languages include the Life-cycle Modeling Language (LML)[180;176], Service oriented architecture Modeling Language (SoaML)[181], and the Energy Systems Language[182]. These are among innumerable domain-specific languages that can be found in computer science, software engineering, and adjacent disciplines.

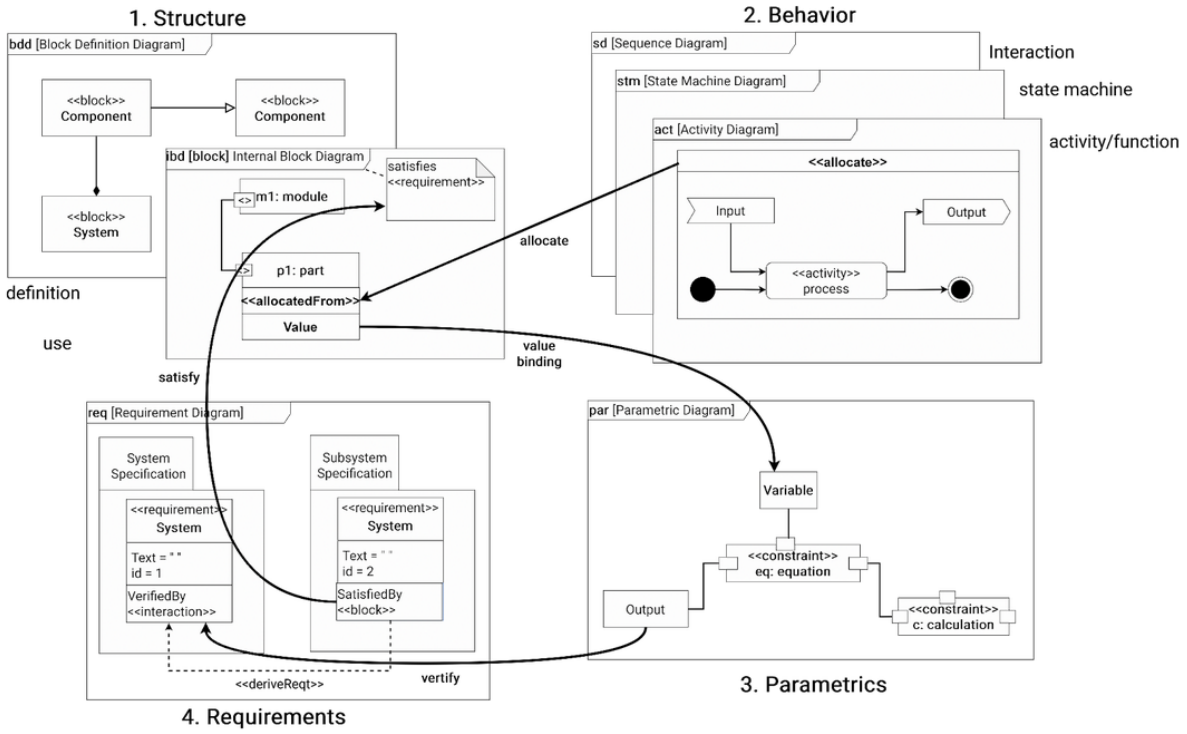


Figure 3: SysML. The abstract representation of the four pillars of SysML[177] (structure, behavior, parametrics and requirements) with their corresponding diagram types, and the relationships between the components in each diagram.

Systems Architecture

In subsection *Challenges to Model-Based Process Systems Engineering*, we explain the lack of research in the integration between models and methodologies in MBPSE. The same problem can also be observed in MBSE. Without a clear direction on how models should interact with other, we are left with an SE or PSE approach that have a "collection of model-based activities rather than model-based engineering"[3]. Therefore, success in MBSE is defined by the commit to systems architecture. Systems architecture is the deliberate design of systems taking into consideration its various components and their relationships, in order to facilitate integration and meet specific goals and requirements. Similar to the concept of "Enterprise Modeling", the core value of systems architecture is to create a shared vision of systems' design between developers, managers, and

stakeholders. This common reality facilitates effective communication, information management and most importantly, it enforces new systems to be designed with built-in integration. Practitioners of MBSE use metamodels and ontologies as the frameworks for systems architecting.

Kübler et al. suggest that reality (i.e., the actual item of interest) is represented by a model, which is defined by a metamodel[57]. Therefore, a metamodel is simply a model that describes another model. Metamodels allow the modeling language to enforce syntax, constraints, and patterns on the models built with it[32]. Metamodels create structure and consistency within the model and across its associated documentation and products[183]. Since a metamodel is simply a model that describes the base constructs used by another model, it is possible to recursively define metamodels of metamodels, until a fully self-descriptive, non-decomposable model is reached. For example, the metamodel for SysML, the four pillars of SysML[177], includes specifications of concepts like "block" and "activity" that are foundational to all models built with the language. However, underneath the SysML metamodel lies the (UML) metamodel that defines the elements and relationships used to create UML models, such as class diagrams, sequence diagrams, and state machine diagrams.

Similar to the concept of a metamodel, ontology is the science of classification and description of things and their relationships. When used with a definite article, "the ontology" of a particular domain provides a dictionary of terms that humans and machines can use to describe a particular set of topics and the relationships among them[32]. Ontologies and metamodels serve very similar roles within SE, both providing descriptive baselines for an area of interest. However, ontologies are typically formulated with open-world assumptions, while metamodels are closed-world implementations. i.e., an ontology asserts that when certain terms are encountered, we can all agree on what they mean; a metamodel requires that all data is traceable to terms it has established. Although this may seem like a dramatic distinction, it turns out to be far more philosophical than practical in most applications[32]. Beyond the scope of this review, the Semantic Web[184] is a good

representation of the open-world disposition of ontology.

Reductionist Metamodels

Minimizing system representations is of both fundamental and applied interest. The mathematical and scientific interest arises from the linkage of system complexity and size: minimal systems models are less complex. The practical engineering interest is that the size and redundancy of a model challenge the effectiveness of the SE processes. Model "reduction" is therefore attractive in increasing the adoption of MBSE approaches. With the number of modeling languages which exist, and their varying levels of complexity, Schindel posed the question, "What is the smallest model of a system?" [185]. To address this question, Schindel developed the "Systematica" (S*) metamodel [185], shown in Figure 4. S* is an innovative, reductionist metamodel which describes system behavior with respect to subjective stakeholder views and technical behaviors. S* describes requirements, designs, and emergent properties of a system in a relational or object information model. At the time of its release, the metamodel was unique in its level of succinctness in explicating the system element interactions required to achieve the purpose of the system.

S* contemplates two critical elements: stakeholder features and functional interactions among the design components that compose the system. Stakeholder features traverse a wide range of expectations including cost, performance, and compatibility, and can be expressed in both technical language and common vernacular. Functional interactions refer to the manner in which one design component impact the current or future behavior of another component. The behaviors of individual components (or agents) remove as they interact can often lead to complex synergy, also known as emergent behavior. Stakeholder features are modeled as objects which define, in stakeholder language, the high level requirements, which flow down to the functional, logical, and physical levels. The technical behaviors are described through interaction models - a high level

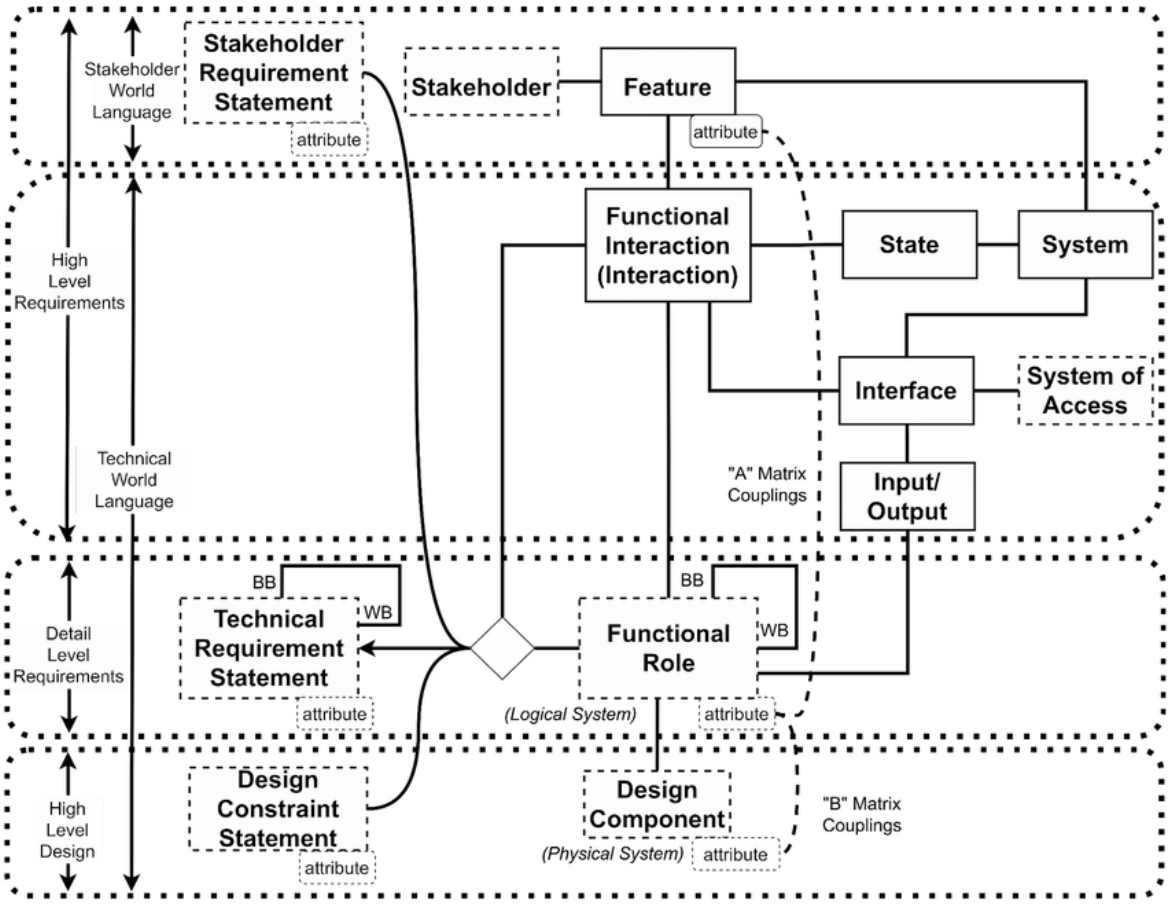


Figure 4: The Systematica (S*) Metamodel. A compact version of the S* metamodel developed by William Shindel (2011)[185]. Such reductionist representations provide a vastly more accessible entry point to systems engineering.

interaction model, state model, and a detail level interaction model. The high level interaction model consists of the name, definition of the interaction, and major attributes of the interaction. In contrast the detailed interaction model also includes an interaction diagram for each interaction showing the input-output relationships between agents and the requirements statements that the describe agent roles. The high level interaction model and state model express the overall system behavior, while the detail interaction model includes interaction diagrams which include input-output relationships and requirements statements[185]. The S* model has been used for various applications, including developing and diagramming patterns[186], supporting decision making[160],

product life-cycle management[59], and facilitating design thinking[187].

Chapter 3: MBSE Strategy for Pharmaceutical Manufacturing - Atropine Sulfate Use Case

Model-based sourcing (MBS) is a technology that supports automated identification of manufacturing solutions to expedite the procurement process for buyers and optimize capacity utilization for manufacturing suppliers. Manufacturing objectives, or more generally stakeholder (or system) features specify the production goal. These features include technical elements, economic factors, regulatory restrictions, raw materials provenance, and other constraints. Features frame the system requirements, including raw material volumes and manufacturing transformations. These requirements are decomposed into sub-systems of process cells, unit operations, and equipment archetypes. The network of features, requirements, and archetypes forms the system-of-systems that establishes the overall requirements governing achievement of the manufacturing objective. To find manufacturing solutions, i.e. configurations of manufacturing capabilities including equipment, personnel and management, these requirements are allocated to manufacturing suppliers. Manufacturing solutions can be further prioritized by higher level stakeholder constraints, such as simultaneous satisfaction of multiple production targets. We have developed a process and initial software capable of automatic allocation of manufacturing assets for using digital assisted management of model parameters, prototype modeling and simulation systems integration, and automated allocation and optimization of manufacturing supply solutions.

The approach illustrated in Figure 2 can be applied using the ASK metamodel, which is on the left hand side of Figure 5, to develop general patterns of couplings between stakeholder features, transformation requirements, and equipment archetypes for a collection of unit operations common to specialty and fine chemical manufacturing processes. Further, using an inventory of domestic manufacturing assets and corresponding semantic ontology for equipment attributes, equipment

instances can be automatically allocated to fulfill manufacturing requirements. The automatic allocation process utilizes transformation requirements, e.g. compatibility, temperature and pressure ranges, and volume, as well as stakeholder features, e.g. regulatory standards and geographic distribution of manufacturing supply sites, and considers the federation and/or distribution of manufacturing objectives across single or multiple sites.

Authoritative Source of Systems Knowledge

In 2015, Procter & Gamble (P&G) created a SysML profile of the S* metamodel in IBM's Rhapsody SysML tool[70] for the purpose of modeling multi-domain phenomena. Subsequent use of the profile exposed improvement opportunities and the model has evolved over the past seven years of usage. We collaborated with P&G to produce the current improved profile known as the Authoritative source of Systems Knowledge (ASK). Because of the fractal nature of the multi-domain systems modeling approach, models built with ASK are also referred to as System Fractals, shown in Figure 5.

ASK differs from S* in the elimination of one metamodel class to enable better support for decomposition of systems into sub-systems and in terms of how it links "Requirements" to other elements of the model. ASK asserts that all features of any system are realized via either causing or preventing the system roles (AKA sub-systems) in response to interactions between the roles of the system. All interactions are assumed to occur via one of four categories of "flow" reflecting conservation laws or inequalities: mass (balance of mass), energy (first law of thermodynamics), force (balance of linear momentum), and information (Clausius-Duhem inequality and proposed conservation law in information theory). Interactions can be intentional (desired) or coincidental (tolerated). Modeling constructs called attributes are used to describe, or quantify where possible, the state of a system behavior and to characterize the behavioral ability of a system to cause a

specific interaction, i.e. to produce a flow that affects another system role. Mathematical coupling objects called constraints are used to account for the nature-imposed cause and effect phenomena emerging from interactions; i.e. how one system role attribute affects another. Requirement statements assign targets and limits on attributes. The system architectural pattern is complete when all system roles and interactions have been captured. System design is complete when all interactions are sufficiently constrained by requirements statements. System allocation is complete when all system roles have been assigned to physical things with behaviors and attributes that are aligned to system role attribute values and requirement statements.

Application of ASK in Pharmaceutical Manufacturing

Here, we show the first application of ASK by creating an architecture of a typical active pharmaceutical ingredient (API) manufacturing system. The left hand side of Figure 5 shows the mapping of each model element in ASK to a critical component in a manufacturing process.

The ASK profile starts with a set of objects representing all stakeholders. Stakeholder needs or requirements are stored in targets and limits properties, or attributes, of "Feature" objects. Feature objects are connected to stakeholders requesting the feature. The feature attributes are connected via equations (constraints) to the system properties they govern. In cases where feature properties specify the state of a system, there are independent variables in constraints where system properties are dependent variables. In cases where feature properties are used assess the performance of a system, feature properties are dependent variables in constraints where system properties are independent variables. Systems have properties to define their state. System state changes are changes to properties values over time. Therefore, we can think of stakeholder requirements as requirements for system state change. Subsystem properties change in response to the passage of flow of mass, energy, force (momentum), or information between subsystems. The passage of time

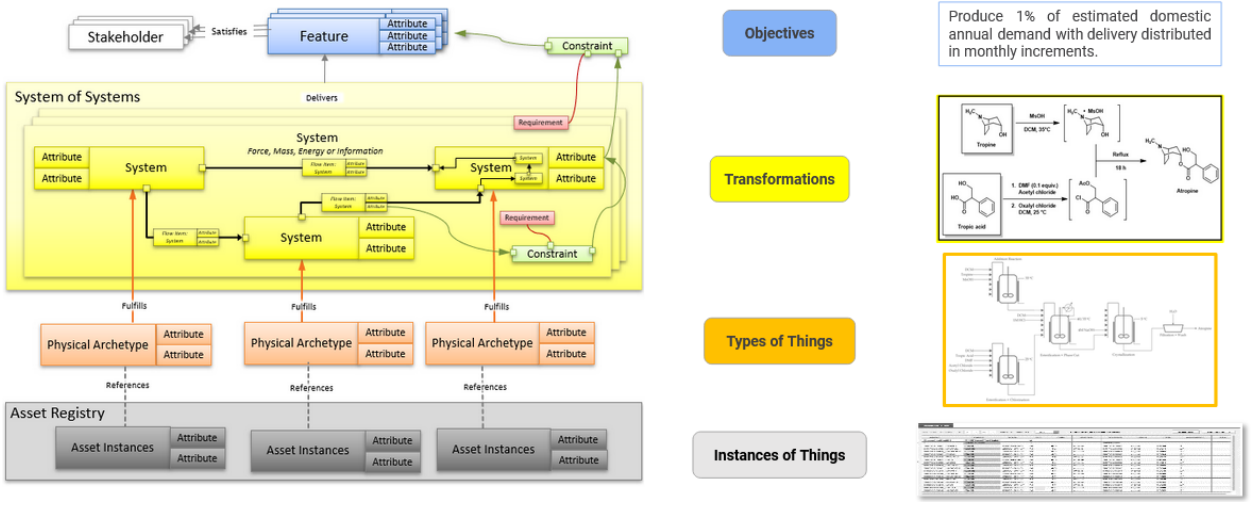


Figure 5: A Model-Based Approach to Find Manufacturing Solutions. Left. Authoritative Systems Knowledge (ASK) metamodel. An end-to-end systems engineering metamodel, providing traceability from Stakeholders and their desired Features, through a logical System of Systems fractal decomposition, to a concrete Physical realization. This incorporates both the semantics and the physics of the system to create a single, holistic representation. Right. Manufacturing objectives, transformation requirements, equipment categories, and asset instances are modeled using a systematic, hierarchical semantic ontology that relates stakeholder features, systems requirements, physical archetypes, and an asset registry to find manufacturing solutions.

and/or flows into a subsystem can trigger flows out of a subsystem. These flows are sub-systems in and of themselves and therefore have properties and states. In addition to modeling dependencies between feature properties and system properties, constraints are also used within the modeling of systems to specify the effect of one system property on other within or across systems. "Physical Archetype" objects are created to represent the types of real-world things that could be acquired to play the roles of various systems in the systems model. This activity is called "allocation" and it includes the mapping of system properties to physical archetype properties. Information about available instances (hereinafter "Assets") of the physical archetypes is gathered to enable implementation of systems. Finally, "Requirement" objects have properties that place limits on the values of system properties and therefore limits on asset properties (via system to physical

archetype to asset property mapping). Requirement properties are often dependent variables of constraints whose independent variables include feature and/or system properties.

Figure 5 also illustrates how ASK can be applied to manufacturing infrastructure. Pharmaceutical manufacturing starts with a set of objectives from stakeholders. A typical objective is to satisfy domestic annual demand of a target molecule, an API for example, by producing at set amount with delivery distributed in monthly increments. This "manufacturing objective" from a particular stakeholder is the request, or the "Feature" of that stakeholder. The target API, the required set amount of API (kg) and the delivery time for every set amount, are the *attributes* of the "manufacturing objective" Feature. Based on the Feature attributes, manufacturers seek out patented synthetic routes that would realistically allow them to satisfy the monthly delivery constraints. The synthetic route is defined as a collection of transformations associated with converting at least one stable chemical species into another until the starting materials become the final API. The raw materials and transformation requirements for the manufacturing process can be converted to a systems representation. Materials requirements specify the amount of each material needed to meet the manufacturing objective – and thereby, constrain the volume/mass and chemical compatibility of transformations in the manufacturing process. Transformation requirements are the operating conditions required for the chemical conversion to occur. Each transformation associated with a manufacturing process can be categorized by an archetype. Transformation archetypes (e.g. heating, dissolution, reaction) form transformation systems, or unit operations. Each unit operation has one or more physical archetype (e.g. heating jacket, reaction vessel, filter) that is capable of performing a collection of transformation archetypes. Physical archetypes are mapped to real instances of equipment (assets) that are capable of performing the role(s) of the physical archetypes. Each asset is selected based on the its ability to satisfy the raw material requirements (e.g. volume and material compatibility) and the transformation requirements (e.g. maximum and

minimum temperature, pressure) of the transformation system.

Overview of Atropine Sulfate

Atropine is an anticholinergic medication that can be administered before anesthesia as it has the ability to reduce salivation as well as maintaining a normal heart rate. Atropine Sulfate, a common salt form of Atropine, is used for intravenous (IV) administration due to its solubility in water. In this work, we chose the synthetic route described in patent WO/2016/016692 due to its simplicity while covering many basic unit operations. The invention provides a high yield, one pot process for the synthesis of Atropine using Tropine and Tropic acid as starting materials.[188].

The synthetic route of Atropine Sulfate, following patent WO/2016/016692, contains five chemical transformations. Figure 6 shows that each transformation is organized into separate, colored "cells". This concept mimics the recently developed manufacturing approach called cellular manufacturing. Cellular manufacturing is a type of manufacturing process that organizes production into small, self-contained cells, in which each cell is designed to perform specific functions in the production line[189]. In the current context, each "process cell" (PC) is responsible for a specific chemical transformation required to make Atropine Sulfate. Each process cell has one or more stable inputs, one or more stable outputs, a series of unit operations and requirements that turn inputs into outputs. Process cells are numbered backwards starting from the final API to the first transformation(s). This setup allows for back integration from API to registered starting material and integration of current Good Manufacturing Processes (cGMP) regulations for API. The synthesis starts with process cell 3 (PC3), converting Tropic Acid to Acetyltropoyl Chloride (red). In PC2, Tropine is converted to Tropine Methanesulfonate in a separate vessel and in parallel with PC3. Once finished, we combine PC2 and PC3 to make PC1c (green), which is the synthesis of Crude Atropine. In PC1b (blue), Crude Atropine is purified via recrystallization. Finally, the

recrystallized Atropine is converted into Atropine Sulfate in PC1a (violet), ending the synthesis.

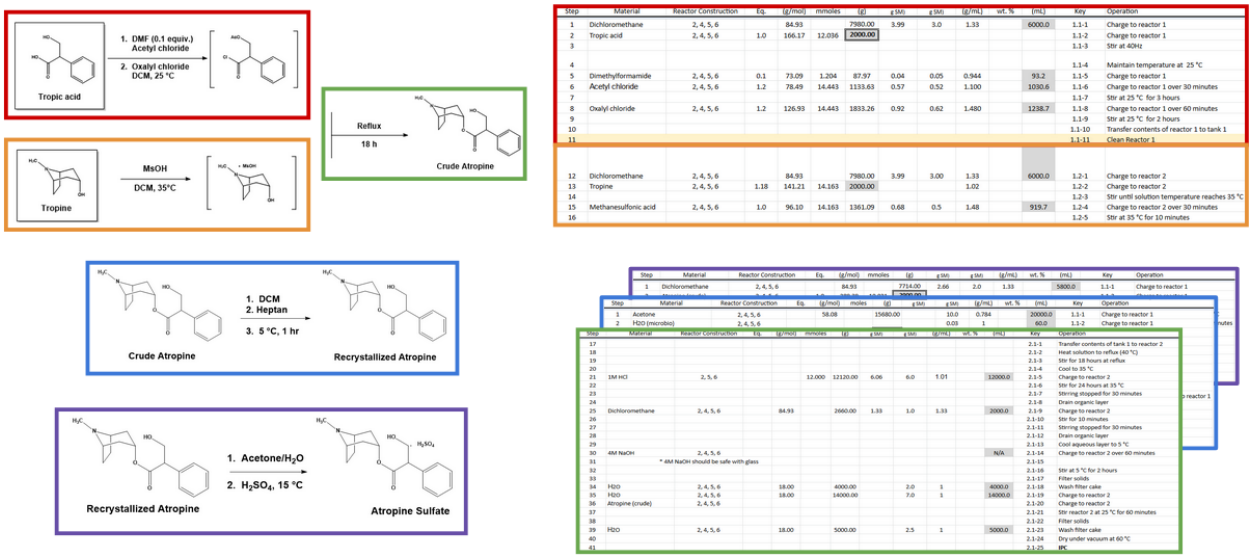


Figure 6: Process chemistry for the active pharmaceutical ingredient (API) Atropine. **A**. Atropine synthetic sequence with five process cells (PCs): PC3 Acetyltropoyl Chloride Formation System (red); PC2 Tropine Methanesulfonate Formation System (orange); PC1c Crude Atropine Formation System (green); PC1b Recrystallized Atropine Formation System (blue); PC1a Atropine Sulfate Hemihydrate Formation System (violet); **B**. Recipe spreadsheet detailing information related to raw materials and chemical transformations as well as their requirements at every step for every PC.

Figure 6 displays the step-by-step recipe of each PC in the form of a batch record. Information such as: required raw materials, material compatibility, mole, mole equivalence to starting materials, mass, volume, density and operation conditions are specified at every step. The organization of batch records into PCs makes the information easier to digest while helping readers locate their location in the synthetic route at every step. Chemical engineers can easily extend allocate the recipe to equipment archetypes by constructing Process Flow Diagrams (PFDs). However, this approach is unable to adjust to sudden changes that are associated with dynamic stakeholder requirements. Consider a situation where the stakeholder request increases from 1% of domestic annual demand to 3%. As a process engineer of a manufacturing plant, we must assess the risk related to this sudden

change by answering the following series of questions. After the production objective increases from 1% to 3%, what is the amount of raw material requirements for every chemical required? What is the current volume requirements of every equipment to keep up with the three-fold increase in production every month? Do we have the assets fit to keep up with this increase in demand? Is the current manufacturing/synthetic route suitable for this increase in demand? It is not trivial to answer these questions by following the DBSE or MBPSE approach where information related to raw material requirements, transformation requirements and equipment specifications are stored and optimized in separate documents/models which all have different syntax, interfaces and output formats.

Atropine Recipe Management System

"Enterprise-wide optimization" [120] requires integration of all system components so that we can optimize "enterprise level" operations such as supply chain and manufacturing systems. In the previous section, the importance for systems to be dependent on all stakeholder requirements emerged. Here, we show the steps required to make the Recipe Management System (RMS) (Figure 7), a system of systems integration of "enterprise level" components whose values are dependent on stakeholder requirements.

The development of an RMS starts by creating the stakeholder Feature object, "Atropine Doses Requested". In this case, our objective is to create "X" amount of Atropine doses, where each dose is "Y" amount (kg) of Atropine Sulfate. Combining with the expected yield of Atropine Sulfate, we can calculate the total amount (kg) of Atropine required by: taking the product of the "Doses Requested" attribute and "mg Atropine per Dose" attribute, divide that product by 1,000,000 to convert from mg to kg, and divide that again by the "Expected Yield" attribute to get the production target of Atropine Sulfate in kg. All of this calculation is completed in the "Starting Target

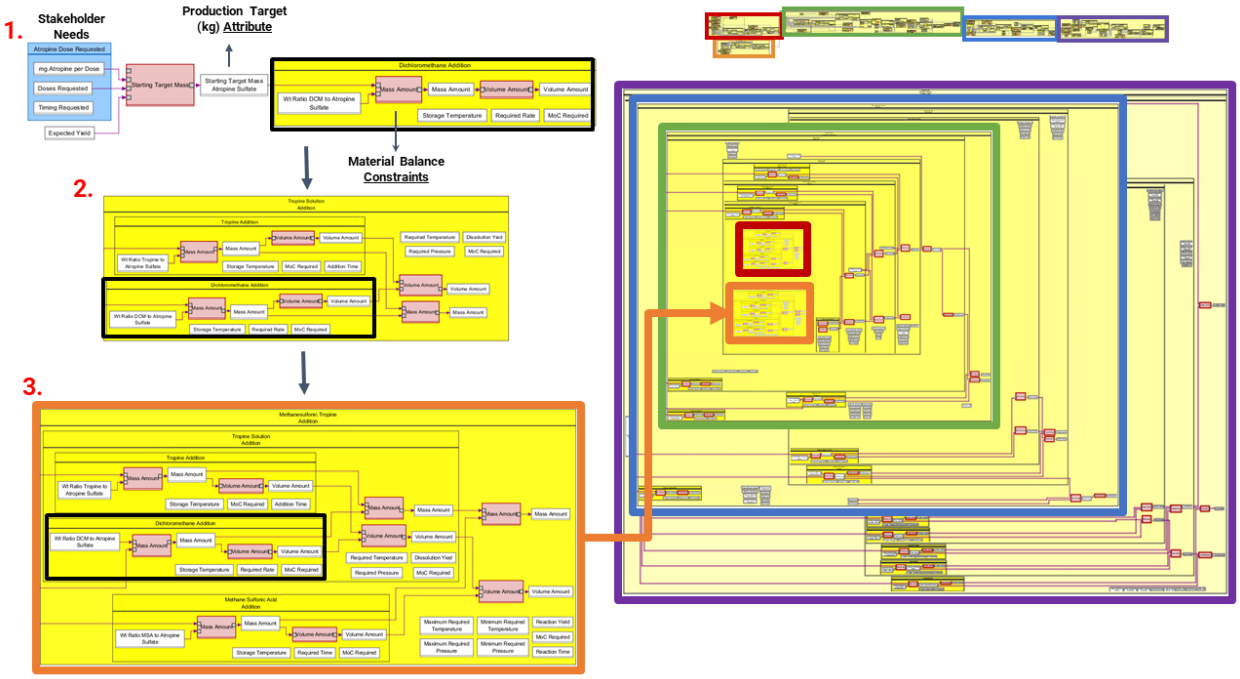


Figure 7: The Recipe Management System. Material requirement system as a series of nested raw material subsystems constrained by stoichiometry, manufacturing objectives, chemical transformations and process conditions.

Mass" constraint, which outputs the final value to the "Starting Target Mass Atropine Sulfate" attribute. According to the recipe spreadsheet (Figure 6), the first unit operation in the manufacture of Tropine Methanesulfonate (PC2) is a liquid transfer of Dichloromethane (DCM). To calculate the mass amount required for this liquid transfer, we multiply the production target with the mass equivalent attribute, "Mass Ratio DCM w Atopine Sulfate", between this particular DCM addition and the production target itself. The resulting mass can then be combined with density of DCM to calculate the equivalent volume required for this particular addition. We can also specify other material requirements such as "Storage Temperature" (^oC), "Required Rate" (m^3/s) and "MoC (materials of construction) Required". These constraints and attributes particular to this specific DCM addition is stored in the "Dichloromethane Addition" system. In Step 1 of Figure 7, using the ASK profile, we demonstrate: (1) how information can be represented, calculated, and stored,

(2) how manufacturing objectives ("Atropine Dose Requested") can be integrated to a simple supply chain management system ("Dichloromethane Addition") in manner that allows the system's material requirements ("Mass Amount" and "Volume Amount") to be dependent on stakeholder requirements ("mg Atropine per Dose"), (3) how to specify other requirements ("Storage Temperature", "Required Rate" and "MoC Required") that are specific to the functionality of the process. Using the same technique, we can create a "Tropine Addition" system for Tropine and combine it with the previous DCM addition system to make a "Tropine mixture" system. The total mass of this mixture is the sum between the previous two mass. We also assume that the volume of this mixture is approximately the summation of the volume of DCM and Tropine. For any given "mixture" system, we can provide operation conditions such as "Required Temperature" and "Mix Time" which can be used for simulations describing the phenomena in transformation archetypes (e.g. distillation, mixing). In Step 2 of Figure 7, we just covered how the RMS keep track of the material requirements (mass and volume) as we move forward as well as the transformation requirements at every step. In Step 3, we show that an RMS for a particular PC, PC2 in this case, can be quickly created by repeating Step 1 and 2. We carry out Step 1 whenever there is a new material addition and Step 2 whenever an operation involving the entire mixture is encountered. Once RMS for other PCs are created using Step 3, we can integrate and organize these RMSs to match the synthesis of Atropine depicted on the LHS of Figure 6. The RMS for PC2 (orange) and PC3 (red) are placed parallel to each other as they are made in parallel and in separate containers. PC2 and PC3 are combined to mimic the actual "Crude Atropine" reaction that occur when mixing Tropine Methanesulfonate and Acetyltropoyl Chloride. Furthermore, PC2 and PC3 are nested inside PC1c (green) as there are work-up steps after the reaction. "Crude Atropine" (green) is then recrystallized (blue) and sulfonated to make Atropine Sulfate (violet). To summarize, this demonstrates a management system that is designed based on the chemical transformations re-

quired to make an API and are capable of adapting to stakeholder objectives throughout the entire management system. All attributes that specify operating conditions (e.g. temperature, pressure, flow rate, time) are used for the calculation of phenomena described by transformation archetypes (e.g. heating, mixing) in the Transformation Requirements System (TRS).

Atropine Transformation Requirements System

In the previous section,we developed an "enterprise level" system of systems that managed the material and transformation requirements of Atropine manufacturing. Here, we will create a system model that capture the actual "process level" operations required to manufacture Atropine. Figure 8 presents the birds-eye view of the system of systems called the Transformation Requirements System (TRS). The TRS is a collection of chemical transformation systems (Process Cells). Each PC consists of a series of unit operation systems that follow the exact progression laid out in the RMS. Each unit operation system uses the operating conditions specified in the RMS (e.g. temperature, pressure, flow rate) as input to calculate the time (seconds) required to perform the transformation archetypes (e.g. heating, dissolution, reaction) on a particular material amount (e.g. mass, volume). The passage of flow carrying mass, energy, (momentum) or information from one unit operation affect the time it takes to reach the desired state of the unit operation receiving that flow.

The unit operations in the TRS are constructed using the practice of Pattern-based Systems Engineering (PBSE). PBSE is a methodology that leverages the concept of patterns to improve the efficiency in engineering and design. In science, patterns are recurring phenomena that are observable in nature. In engineering, a pattern is a recurring solution to a common problem that has proven to be effective. These patterns are used to guide the design and development of systems, ensuring that tried and tested approaches are incorporated into the process. The utilization of

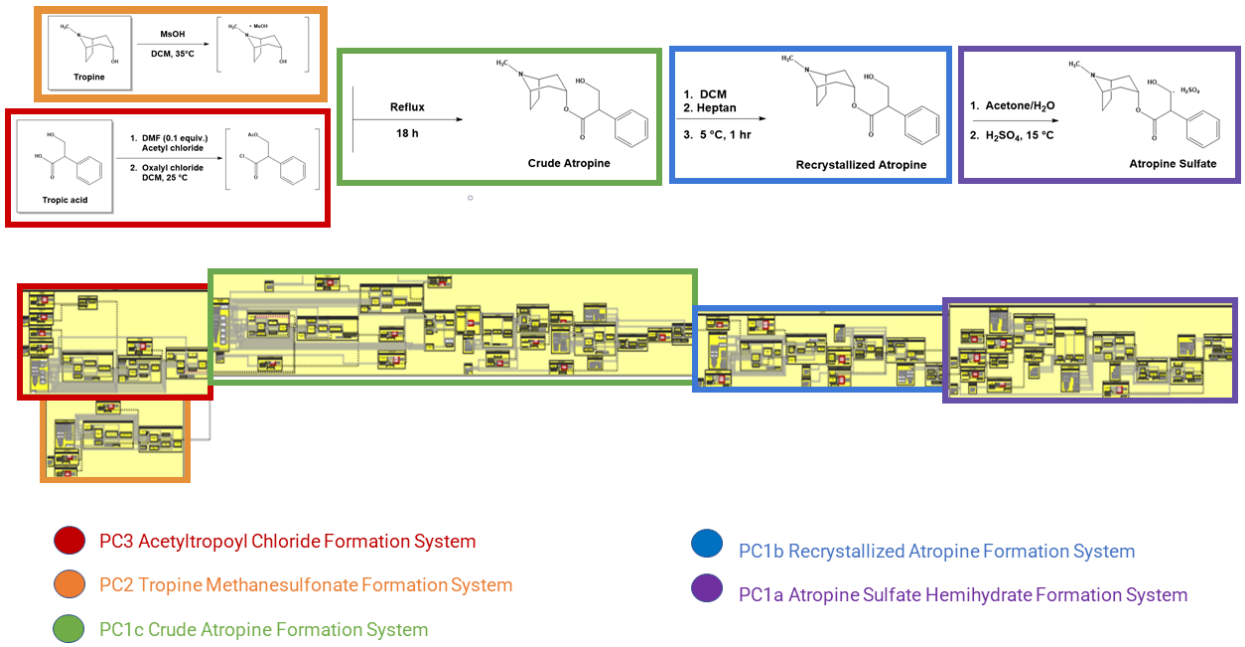


Figure 8: Transformation Requirements System. A high level, structural model of Atropine manufacturing system's transformation requirements highlighted by process cell. The architectural model follows the Atropine synthetic route, starting from Tropic acid and Tropine as key starting materials. In parallel, Tropic acid and Tropine are synthesized to create PC3 Acetyltropoyl Chloride Formation System (red) and PC2 Tropine Methanesulfonate Formation System (orange) respectively. PC3 and PC2 are inputs to produce Crude Atropine in PC1c Crude Atropine Formation System (green). After undergoing recrystallization in PC1b Recrystallized Atropine Formation System (blue), the recrystallized Atropine is sulfonated to create Atropine Sulfate in PC1a Atropine Sulfate Hemihydrate Formation System (violet). All "formation systems" are connected by passages of flows of mass, energy and information.

patterns leads to a more efficient and effective engineering process, and the consistent application of patterns across projects results in the emergence of PBSE as a discipline[190]. PBSE promotes reusability and standardization in the engineering process, resulting in a more streamlined and effective approach to system development. In the case of chemical manufacturing, we develop patterns for transformation archetypes using simple transport phenomena models that are commonly used in unit operation design. The specifications of these patterns are discussed in subsection *Sim-*

ulation Authoring. By creating a repository of generalizable patterns of unit operations, we can streamline the modeling process by simply instantiating new instances of required patterns inside a PC. On top of that, we were able to add distributed version control capabilities into our pattern repository. This automation allows us to seamlessly update the current pattern or replace it with a different one while maintaining the integrity of the entire system. Once a PC has received the required unit operation patterns, they can be connected via passage of flows from one unit operation to another by following the direction of the RMS. Flows can carry information between unit operations inside or across PCs. The passage of time and/or flows into a unit operation can trigger flows out of another unit operation. This trigger of input and output continues until no out-flow can be detected in a system that receives an in-flow. The architecture of the TRS is complete once all system roles, interactions and transformations have been captured via flows and constraints.

Simulation Authoring

With both the supply management and the manufacturing operations required to product Atropine, we develop a digital twin by integrating simulations into the system architecture. Here, we integrate simulation authoring capabilities to simulate the collection of unit operations patterns instantiated inside every PC.

Figure 9 provides a closer look of the TRS for PC2 Tropine Methanesulfonate (top) and what the "molecular level" simulation of PC2 looks like (bottom). According to patent WO/2016/016692, there are five steps required to produce Tropine Methanesulfonate. The architectural system (middle) starts with the parallel material additions of liquid DCM (blue) and solid Tropine (red). The end of the transfer initiates the dissolution of Tropine in DCM at 35^oC (brown). Once Tropine fully dissolved, Methanesulfonic acid is added drop-wise into the mixture (green). The reaction mixture is continuously stirred at 35^oC until the reaction completes (violet) [188]. Each of these steps

is described by a pattern and they are connected together via flows. As mentioned before, every pattern is created based on one or more transport phenomena models that were developed to model unit operations. For example, we assume that Tropine Methanesulfonate is produced in a baffled, jacket agitated vessel. With that assumption, we now have a model for the heat transfer from the jacket of a baffled tank described by McCabe et al.[191]. Using this heat transfer model, we can build a heat transfer simulation that can calculate the time required to heat the Tropine-DCM mixture to 35^oC. Similar models can be found in McCabe et al. to describe the phenomena of liquid transfer and dissolution required to create the patterns required to model PC2. In the simulation, we can direct the flows of mass, energy and information from one pattern to another to simulate the process of forming Tropine Methanesulfonate from Tropine (key starting material). Executing the completed systems model of PC2 results in a fully simulated manufacturing of Tropine Methanesulfonate (Figure 9 bottom), which contains a diverse set of unit operations such as liquid transfer, solid transfer, heating, dissolution, and reaction. The output of this simulation is the operating time of each unit operation and the cycle time of Tropine Methanesulfonate, all of which are dependent on the production objective calculated in the RMS (Figure 7).

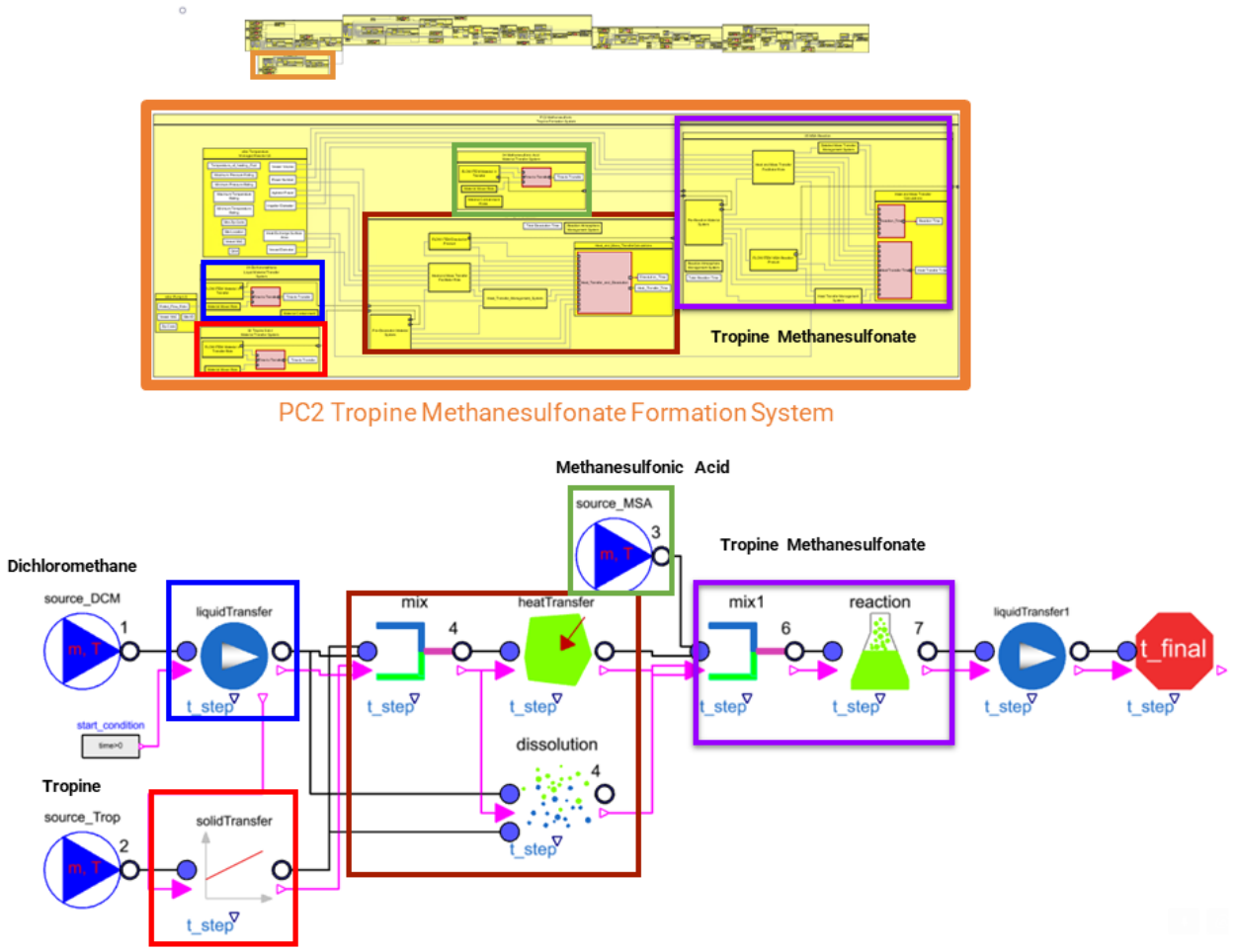


Figure 9: Digital Twin in MBSE. A closer look of a Process Cell using PC2 Tropine Methanesulfonate Formation System as an example (top). The process starts with the liquid addition of DCM (blue) and the solid addition of Tropine (red) in parallel. Tropine is then dissolved in DCM at 35^oC (brown). After the drop-wise addition of Methanesulfonic acid (green), the reaction mixture is stirred at 35^oC until a solution forms (violet). Each unit operation is connected via passage of flow of mass, energy and information. Executing the model results in the complete simulation of PC2, which is the formation of Tropine Methanesulfonate from the starting material Tropine (bottom). Each simulation elements is built using transport phenomena concepts that are commonly used in PSE. The output of this simulation is the cycle time required to manufacture Tropine Methanesulfonate.

The outcome of this use case is a digital twin—an architectural model that enables the integration of simulations whose outputs depend on stakeholder objectives. With our automation and

version control capabilities, any modification to the model architecture or underlying simulation can be swiftly shared across the enterprise. This approach allows for scalability and improved fidelity of the systems model without compromising the shared reality and model integrity. Ultimately, integrating advanced simulation suites from the PSE community will enhance the MBSE digital twin's resolution, further streamlining the modeling process. However, calculating cycle times necessitates not only transformation requirements but also equipment that (a) possess the necessary properties for transformations to occur (e.g., impeller diameter, filter pressure, heat exchange surface area), and (b) meet the specifications required for carrying out the transformations (e.g., maximum and minimum temperature requirements, volume, MoC). In Chapter 4, we present automation capabilities that can automatically allocate suitable assets based on production objectives and transformation requirements embedded in architectural models.

Chapter 4: Automatic Allocation of Assets - Albuterol Case Study

Overview of Albuterol

Albuterol, a short-acting β2-adrenergic receptor agonist (SABA) primarily used for treating bronchospasm associated with asthma and chronic obstructive pulmonary disease (COPD), works by relaxing smooth muscle in the airways, resulting in bronchodilation and improved airflow[192]. Although available in various forms, the most common administration method is through aerosol metered-dose inhalers (ProAir HFA, Proventil HFA, others) due to their rapid therapeutic effect and ease of use[193]. Since October 2022, Albuterol inhalation solution, specifically the Albuterol Sulfate Inhalation Solution 0.5% used in hospitals and healthcare systems, has been on the FDA drug shortage list[194]. The American Lung Association (ALA) attributes this shortage to the shutdown of Akron Operating Company, LLC[195], a major U.S. manufacturer of this specific Albuterol form. This closure increased demand on Nephron Pharmaceuticals, the only remaining U.S. manufacturer capable of producing the same product. While the FDA and ALA continue to monitor the situation, the only current solution to this shortage is waiting for supply to meet demand, emphasizing the importance of finding a new U.S. supplier for this crucial API.

Albuterol Synthetic Routes

In this case study, we compare two Albuterol synthetic routes: one proposed by Babad et al. in 1988[196] (hereinafter Albuterol 1), and the other proposed in the international patent WO 92/04314[197] (hereinafter Albuterol 2).

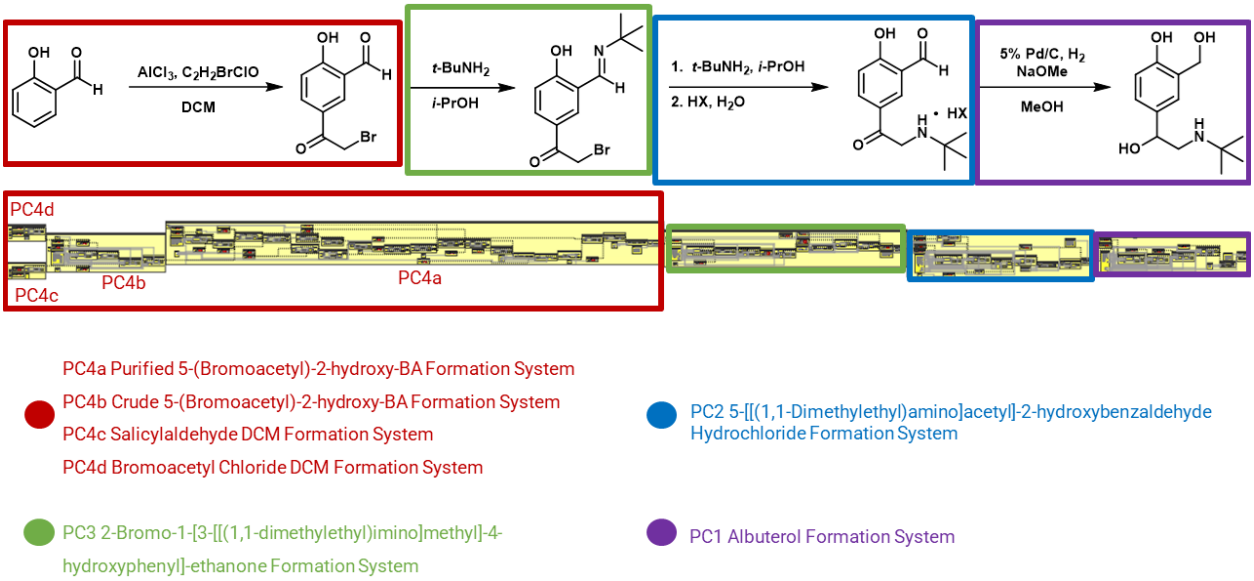


Figure 10: Albuterol 1 Architectural Systems Model. The architectural model follows the synthesis route suggested by Babad et al.[196], using Salicylaldehyde as the key starting material. In parallel, Salicylaldehyde (PC4c) and Bromoacetyl Chloride (PC4d) were prepared in separate tanks containing Dichloromethane. These two solutions are combined in PC4b containing Aluminum Chloride to produce crude 5-(Bromoacetyl)-2-hydroxy-BA (benzaldehyde). Crude 5-(Bromoacetyl)-2-hydroxy-BA (benzaldehyde) are recrystallized in PC4a and reacted with Tert-butylamine (t-$BuNH_2$) in 2-propanol to produce 2-Bromo-1-[3-[[(1,1-dimethylethyl)imino]methyl]-4-hydroxyphenyl]-ethanone (stage II product) in PC3. The resulting stage II product reacted with t-$BuNH_2$ and Hydrocloric acid in 2-propanol solution to produce 5-[[(1,1-Dimethylethyl)amino]acetyl]-2-hydroxybenzaldehyde Hydrochloride (Stage III product). Finally, in PC1, Stage III product are hydrogenized in Palladium catalyst to produce Abuterol.

The top half of Figure 10 shows the synthesis route proposed in Albuterol 1, using salicylaldehyde as the key starting material. This synthesis route has 4 transformations, which are described in the TRS on the bottom half of Figure 10. The production starts at PC4d and PC4c, where Bromoacrtyle Chloride and Salicylaldehyde are mixed, in parallel in two separate tanks containing Dichloromethane. These two solutions come together in PC4c to create crude 5-(Bromoacetyl)-2-hydroxy-BA (benzaldehyde). Crude 5-(Bromoacetyl)-2-hydroxy-BA in

PC4c are transferred to another reactor in PC4a for recrystallization. In PC3, the purified 5-(Bromoacetyl)-2-hydroxy-BA reacted with tert-butylamine in 2-propanol to create 2-Bromo-1-[3-[[(1,1-dimethylethyl)imino]methyl]-4-hydroxyphenyl]-ethanone or stage II product (green). Stage II product is then reacted to tert-butylamine and 12 M HCl to create 5-[[(1,1-Dimethylethyl)amino]acetyl]-2-hydroxybenzaldehyde Hydrochloride or stage III product (blue) in PC2. Finally, stage III product is hydrogenated in Palladium catalyst in PC1 to produce Albuterol (purple).

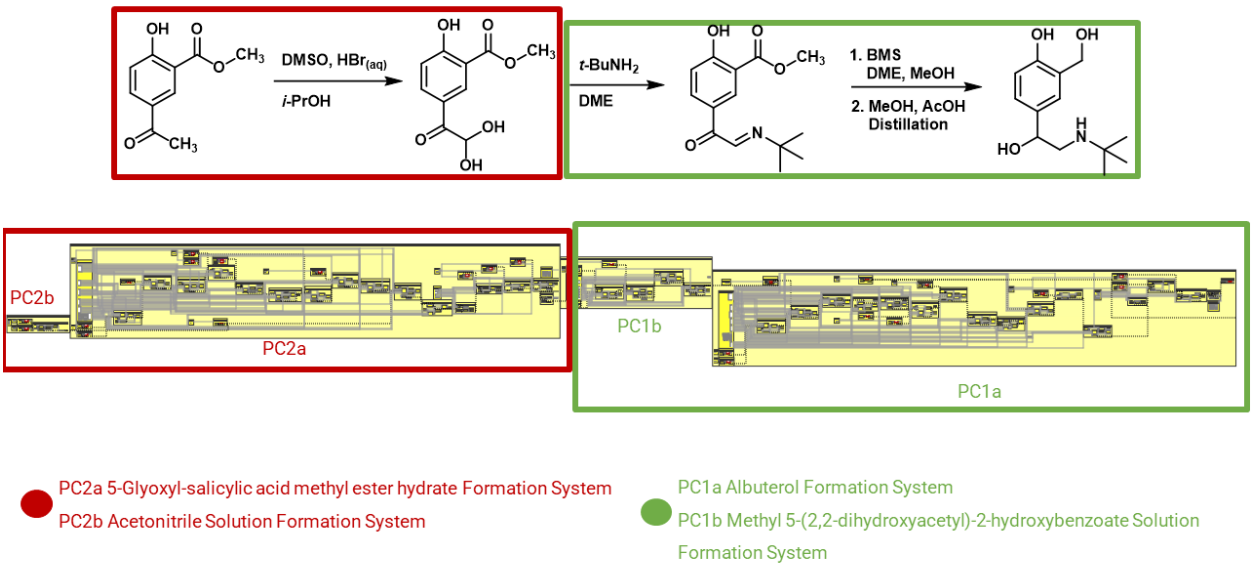


Figure 11: Albuterol 2 Architectural Systems Model. The architectural model follows the synthesis route proposed in patent WO 92/04314[197], which uses Methyl 5-acetylsalicylate as the key starting material. In PC2a, methyl 5-acetylsalicylate is reacted with Dimethylsulfoxide, Hydrobromic acid and 2-propanol to create 5-glyoxyloyl-salicyclic acid methyl ester hydrate (5-glyoxyloyl-salicyclic acid ME), which is precipitated out using diluted Acetonitrile prepared in PC2b. 5-glyoxyloyl-salicyclic acid ME is combined with ethylene glycol diethyl ether (DME) and t-$BuNH_2$ in PC2b to produce a dihydroxyacetyl intermediate, which is distilled in a solution of Borane dimethyl sulfide complex and DME to create aryethanolamine intermediates in PC2a. Finally, aryethanolamine intermediates are concentrated in Methanol and Acetic acid to create Albuterol.

Figure 11 describes the synthesis and TRS of Albuterol 2. The production starts in PC2a (red) where the key starting material, methyl 5-acetylsalicylate, reacted with dimethylsulfoxide and hydrobromic acid (48%) to make hydroxybenzoate intermediates[197]. These intermediates are mixed with 2.4 N sulfuric acid and Acetonitrile solution from PC2b (red) to make 5-glyoxyloyl-salicyclic acid methyl ester hydrate (methyl ester hydrate). In PC1b (green), the methyl ester hydrate reacts with ethylene glycol diethyl ether (DME) and tert-butylamine to make Methyl 5-(2,2-dihydroxyacetyl)-2-hydroxybenzoate (dihydroxyacetyl). The dihydroxyacetyl interediate are concentrated in 10 M Borane dimethyl sulfide complex, DME and methanol in PC1a to make aryethanolamine intermediates. Finally, the aryethanolamine intermediates are distilled in Methanol and Acetic acid to create Albuterol. To make this synthesis route comparable to Albuterol 1, the following sulfonation step to create Albuterol Sulfate[197] has been removed from the TRS, leaving us with only Albuterol as the final product in PC1a.

In a comparison of the two synthetic routes to Albuterol, several key differences were observed (Table 1). Albuterol 1 is characterized by a more complex, 4-stage synthesis, consisting of 99 steps, whereas Albuterol 2 involves a more streamlined, 2-stage synthesis with 50 steps. The workup for Albuterol 1 is considerably more labor-intensive, necessitating heavy workup[196], while Albuterol 2 requires no such workup. Additionally, Albuterol 1 synthesis involves hydrogenation using palladium catalysts[196], whereas Albuterol 2 requires multiple distillation steps involving bromine compounds[197]. Notably, the MoC requirements for Albuterol 1 are more flexible than Albuterol 2 due to the use of hydrobromic acid[197] which requires strictly glass-lined reactors. Lastly, the reaction conditions for Albuterol 1 involve low temperatures (-3^{o}C) and high pressure (60 psi)[196], whereas Albuterol 2 requires high temperatures (100^{o}C)[197]. It is also worth mentioning that Albuterol 1 yields a liquid product, while Albuterol 2 results in a solid product.

Table 1: High-level comparison of transformation requirements between Albuterol 1 and Albuterol 2

Albuterol 1	**Albuterol 2**
99 Steps, 4-stage Synthesis	50 Steps, 2-stage Synthesis
Heavy Workup Required	No Workup Required
Hydrogenation with Palladium Catalysts Required	Multiple Distillation Involving Bromine Compounds Required
Flexible MoC* Requirements	Strict MoC* Requirements
Low Temperaturea (-3^{o}C) Requirements and **High Pressure** (60 psi) Requirements	**High Temperature** Requirements (100^{o}C)
Liquid Product	Solid Product

* Material of Construction

Motivation

The difference in transformation requirements for various syntheses necessitates assets with specific specifications tailored to each synthesis. In the chemical industry, the process of allocating assets compatible with both unit operations and transformation requirements is typically manual, relying on local domain experts and documentation in the form of digital artifacts or papers, as discussed in Chapter 1. The objective of this case study is to demonstrate the automated and rapid allocation of suitable assets to products with differing requirements by incorporating automation capabilities into architectural models. This case study aims to illustrate the conceptual framework of equipment pattern architecture, demonstrate how this architecture enables automatic asset allocation through examples and visualizations, and compare the two Albuterol syntheses and

their respective allocated assets.

Equipment Pattern Architecture

Before incorporating automation into the TRS, which includes the transformation requirements in table 1, we must first develop an architecture for equipment patterns that accurately represents both (1) the requirements and (2) the roles in which the assets can participate. It is evident that equipment of different archetypes, such as containers and filters, have distinct roles and requirements. However, distinguishing between equipment within an archetype is more challenging. For instance, what are the differences between a vessel and a mixer in the context of modeling? Are they the same thing? How about a mixer and a reactor? Without clear differentiation between these types of equipment, automation may assign unit operations with assets that lack the necessary structure or requirements to perform those operations. Consequently, it is crucial to establish a modeling framework capable of adapting to the diversity of chemical manufacturing assets in terms of their roles and requirements.

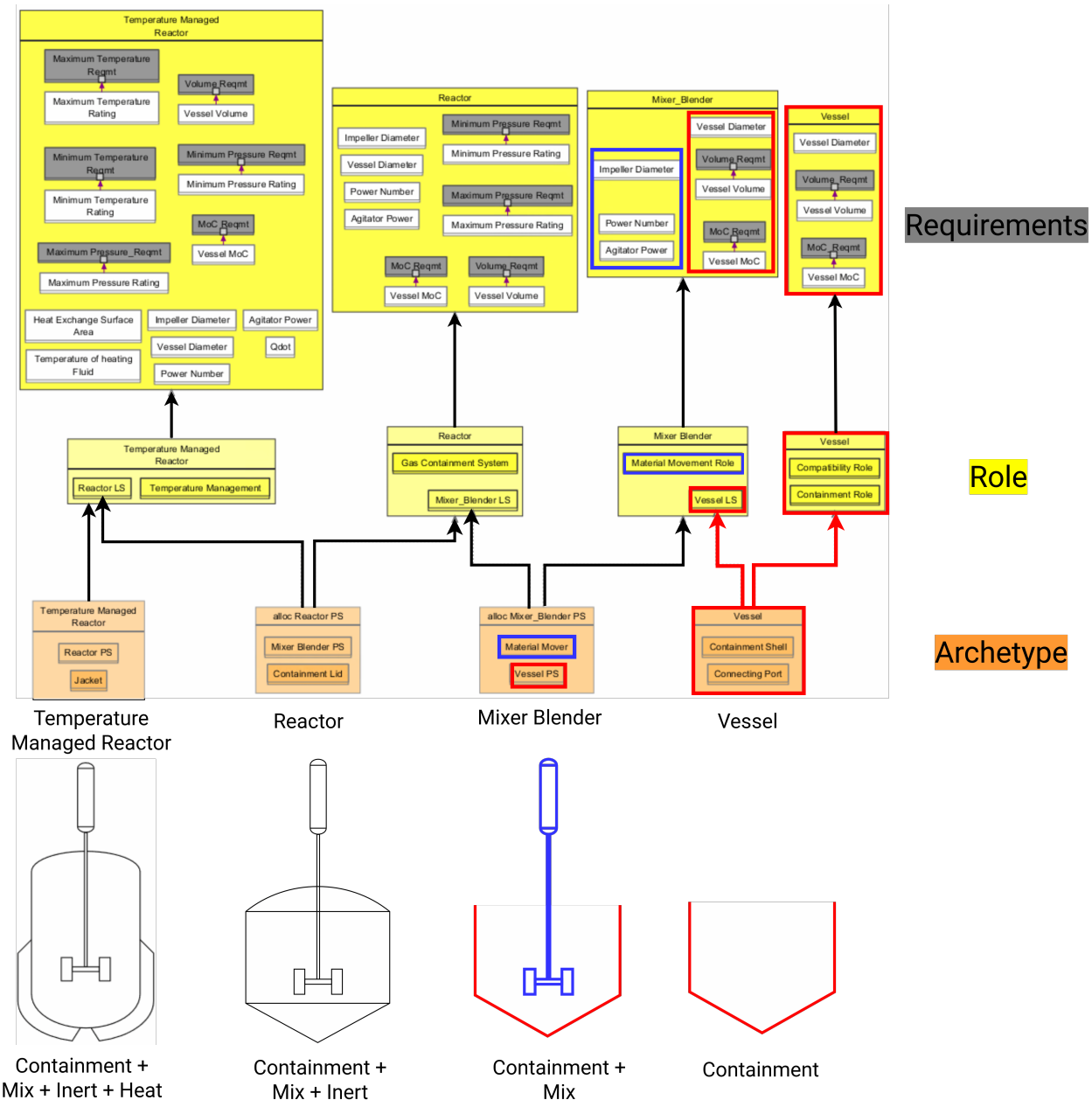


Figure 12: Equipment Pattern-on-Pattern Architecture. This figure showcases the pattern-on-pattern architecture of equipment archetypes within the "container" archetype (orange), illustrating the evolution of roles (yellow) and requirements (gray) as the archetype structure becomes more complex. The vessel (red) represents the simplest archetype, followed by the mixer/blender with added agitation attributes (blue), the reactor with inert environment capabilities, and finally, the temperature-managed reactor (TMR) with temperature control requirements. This scalable framework streamlines the construction of our pattern portfolio.

Figure 12 illustrates the specific pattern-on-pattern architecture of equipment archetypes belonging to the broader "container" archetype (orange). As the archetype "evolves" in terms of its structure, its roles (yellow) and requirements (gray) also evolve. The figure begins on the outer right, where the simplest "container" archetype, the vessel (red), is displayed. The vessel has only the role of a container, and thus, its requirements are limited to volume and MoC. The next evolution of the "container" archetype is the mixer/blender, which is inherently a vessel. Consequently, the mixer/blender possesses the role and requirements of the vessel (red), along with necessary attributes for agitation (blue), such as impeller diameter and agitator power. This pattern-on-pattern framework makes the process of building our pattern portfolio scalable and manageable. Following the same logic, the next generation is the reactor, which has the role of creating an inert environment, in addition to the agitation and containment roles from the mixer/blender. To create an inert environment, the reactor must withstand non-atmospheric pressure ranges, requiring minimum and maximum pressure specifications. Finally, the temperature-managed reactor (TMR) is the last generation shown on the outer left of Figure 12. The TMR can perform the roles of the previous "container" archetypes and temperature management, necessitating two additional requirements—minimum and maximum temperature—compared to the reactor.

It is crucial to note that the equipment pattern architecture is built only to accommodate the automatic allocation (auto-allocation) feature and exists outside and in parallel to the TRS. When an equipment pattern is called within any TRS, only the patterns containing the requirements are invoked. These patterns include (1) the requirements that take attributes from the RMS as inputs (e.g., reaction temperature and pressure) and (2) the attributes enabling a unit operation to occur (e.g., impeller diameter, agitator power, and power number for dissolution). When the auto-allocation feature is called, information flows from the requirements to the roles requiring those specifications, then to the equipment archetypes capable of performing those roles, and finally to the

actual assets that (1) match the required archetypes and (2) have specifications deemed "suitable" for the transformation requirements. The process of selecting "suitable" requirements is explained in Figure 13.

Automatic Allocation of Assets

Automatic allocation or auto-allocation is the process of automatically and rapidly allocating suitable assets required in a complete TRS via embedded automation. In this section, we illustrate the allocation process and the auto-allocation results, which is the manufacturing solution for each Albuterol synthesis.

Illustration of the Allocation Process

Figure 13 visually illustrates the allocation algorithm and its relationship with the equipment pattern architecture show in Figure 12.

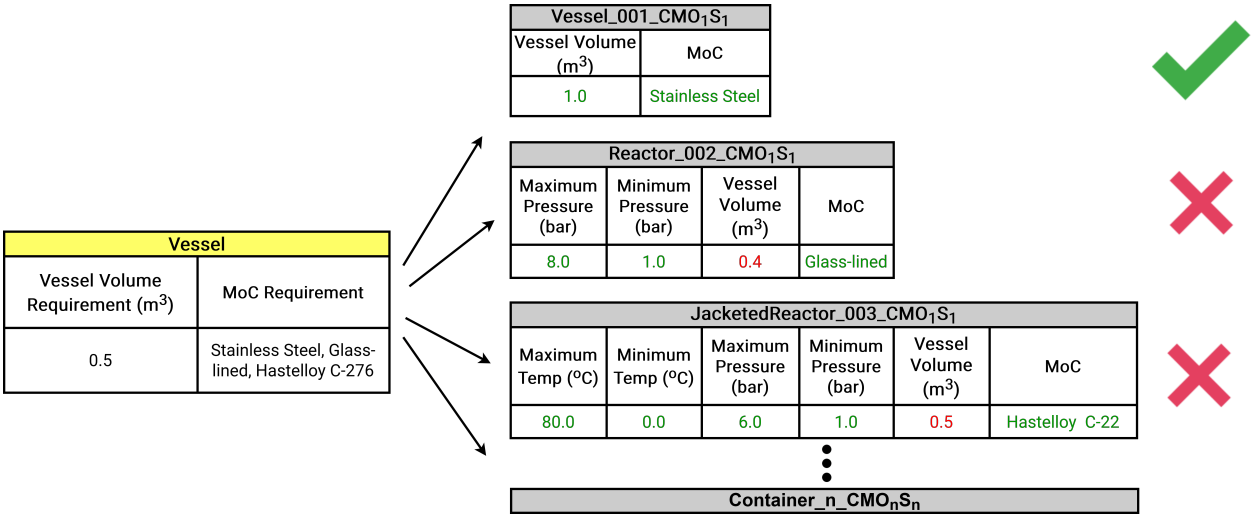


Figure 13: The Process of Equipment Allocation. This figure demonstrates the auto-allocation process for an Equipment of Interest (EOI) requiring a 0.5 m^3 vessel made of specific materials. The EOI is compared to assets belonging to the "container" archetype across various toll manufacturers' sites, considering volume, material compatibility, and archetype evolution. This example highlights the complexity of allocation, emphasizing the importance of allocation and an established equipment architecture.

In Figure 13, the equipment of interest (EOI) is a vessel required to have a volume of 0.5 m^3 and be made of stainless steel, glass-lined, or Hastelloy C-276. The vessel with these requirements is compared to equipment that (1) belongs to the "container" archetype, (2) contains the "vessel" archetype in their evolution, and (3) has suitable specifications. The first comparison is made between the EOI and Vessel-001-CMO_1S_1 from site 1 of toll manufacturer (toller) 1. Since this vessel has a volume of 1.0 m^3 and is made of stainless steel, it is considered suitable. The second comparison is between the EOI and Reactor-001-CMO_1S_1, as Figure 12 shows that the reactor archetype can perform the role of the vessel archetype. However, Reactor-001-CMO_1S_1 is not chosen, as its volume (0.4 m^3) is less than the required volume of 0.5 m^3. The third comparison is made between the EOI and JacketedReactor-001-CMO_1S_1, as jacketed reactor assets are considered to be of the TMR archetype. JacketedReactor-001-CMO_1S_1 does not meet the requirements of the

EOI, as it has the same insufficient vessel volume. This example demonstrates that allocation is not as linear a process as one might imagine. While this asset is technically suitable as it has the same volume as the EOI, in reality, an asset with a slightly larger volume is preferable to prevent spillage. The required volume "difference" is a variable that can be adjusted in the allocation algorithm before initiating the allocation process. The same concept applies to temperature and pressure as well. Moreover, while the MoC between the EOI and the jacketed reactor is not identical, Hastelloy C-22 is considered superior to Hastelloy C-276[198], so this asset meets the MoC requirements for the EOI. Logic related to MoC is also a crucial component of the allocation algorithm, as it can lead to rejecting many assets with suitable MoC or accepting ones that lack suitable MoCs. The allocation process for this EOI continues until all assets belonging to the "container" archetype in all sites of all tollers are compared. While Figure 13 illustrates auto-allocation of one equipment, auto-allocation can also be carried out on the scale of one PC, multiple PCs, or the entire project.

It is essential to remember that the requirement values of all EOIs come from the RMS, which contains values dependent on stakeholder requirements (features). A change in stakeholder requirements, such as a modification in synthesis route or production objective, will result in a different auto-allocation output. Therefore, similar to simulation (Figure 9), auto-allocation depends on stakeholder requirements. This concept will be relevant when comparing allocation results between the two albuterol syntheses in later sections.

Finally, assets with parameters compatible with the transformation requirements are called "fit" assets. While "fit" assets represent only a small portion of available assets, the design space remains significant for the entire manufacturing system. This is because not all "fit" equipment is efficient to deploy. Assessment of "goodness" can be used to limit the selection of overqualified equipment. However, the topic of "goodness" is outside the scope of the current chapter, as it is related to quantitative measurements and optimization, which are independent of the process of

automatic asset allocation.

Manufacturing Solution

The auto-allocation results of a full architectural model consist of a collection of asset groups, with each group representing all qualified assets for a specific EOI. Each asset is associated with its current site location and the respective toller who owns the site. An equipment configuration is a collection that contains one asset from each EOI's asset group. Thus, a manufacturing solution is defined as a distinct equipment configuration situated at a specific site, owned by a particular toller. In this section, we provide visual illustrations of manufacturing solutions and compare manufacturing solutions between the two Albuterol synthesis routes. Specifically, these are the "best" manufacturing solutions for a **production objective of 29 kg of Albuterol**, which we estimated as 1% of the annual demand for this compound[199].

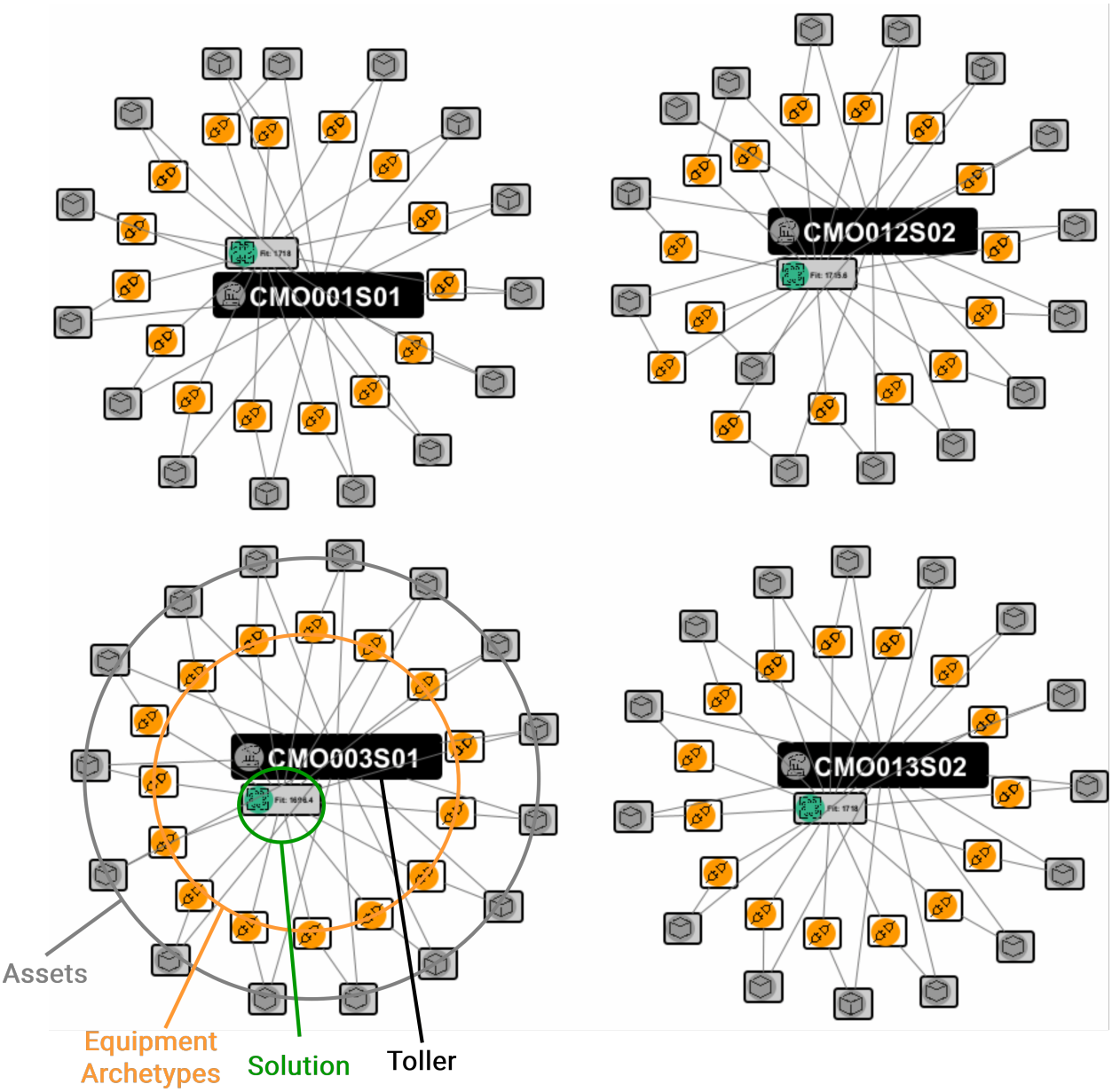


Figure 14: Visual Illustration of Manufacturing Solutions For Albuterol 1. Figure 14 displays four groups, each representing a manufacturing solution from a specific toller and their site(s) capable of meeting the production objective following Albuterol 1. The central green object contains the solution ID and its "goodness" score. It connects to a dark block indicating the toller and their qualified site, e.g., *CMO003S01* represents site 01 of toller 003. The toller is linked to the required equipment archetypes (orange) at that site, which in turn connect to the qualified, auto-allocated assets (gray).

Figure 14 shows 4 groups, where each group is a manufacturing solution of a particular toller and their specific site(s) that are capable of satisfying the production objective by following Albuterol 1. At the center of the group lies a green object containing solution ID and its "goodness" score. The solution is connected to a dark block that contains a toller and their qualified site in a form of toller-site. For example, *CMO003S01* means site 01 of toller 003. The toller is connected to all the equipment archetypes (orange) that (1) has the roles and requirements needed in the architectural model, and (2) belong in that particular site. Finally, each equipment archetype is connected to a qualified, auto-allocated asset (gray).

Equipment Archetype	Process Cell	Qualified Asset Instances	CMO_1S_1	CMO_3S_1	$CMO_{12}S_2$	$CMO_{13}S_2$
Filter	PC1	73	23	6	35	9
Temperature Managed Reactor Condenser	PC1	36	12	12	9	3
Filter	PC2	73	23	6	35	9
Temperature Managed Reactor	PC2	75	27	13	24	11
Dryer	PC2	31	14	7	4	6
Dryer	PC3	26	11	6	3	6
Filter	PC3	12	3	3	3	3
Temperature Managed Reactor	PC3	65	27	13	24	1
Filter	PC4a	41	3	3	32	3
Temperature Managed Reactor Condenser	PC4a	53	18	10	22	3
Dryer	PC4a	32	14	7	4	7
Filter	PC4a	54	10	5	33	6
Temperature Managed Reactor Condenser	PC4b	68	27	14	24	3
Vessel	PC4c	343	258	24	26	35
Vessel	PC4d	343	258	24	26	35
	Total	1325	728	153	304	140

Figure 15: Auto-Allocation Results of Albuterol 1. Figure 15 provide a detailed summary of all qualified sites and assets to manufacture Albuterol by following Albuterol 1. The data table illustrates equipment archetypes, PCs, total qualified assets, and the number of qualified assets at each site. The figure reveals that only four sites have one or more assets for all required equipment archetypes, reflecting unique requirements and constraints for Albuterol 1. The allocation process considers various factors, including equipment roles and available options, to identify the most suitable manufacturing solutions.

The assets shown in Figure 14 is part of the total assets qualified, which are laid out in Figure 15. Each row is an equipment archetype, the PC (colored by TRS) in which contains said equipment archetype, the total amount of qualified assets at all sites, and the number of qualified assets at each qualified sites. While there can be more than 4 sites that contain the qualified assets for any particular equipment archetype, only 4 sites have one or more assets for all equipment archetype required. Observation of the data table in Figure 15 give us some insights onto why only 4 sites are qualified when using Albuterol 1. Starting at PC1, only a total of 36 Temperature Managed Reactor Condenser (TMRC) across 4 sites are qualified compared to 53 and 68 from PC4a and PC4b respectively. The high operating pressure (60 PSI or 4.1 bar) and low operating temperature (0^oC) combined with hydrogenation[196] set up a unique set of requirements that not many assets can satisfy (Table 1. Furthermore, Albuterol 1 requires a large amount of solvents in PC4 and PC3[196], which is reflected not only in TMRC but also the amount of qualified filters compared to PC2 and PC1. Finally, as illustrated in Figure 12 and 13, all "container" archetype can, and are allocated to fulfill the role of a vessel. Thus, there is always a large amount of options when a vessel is required.

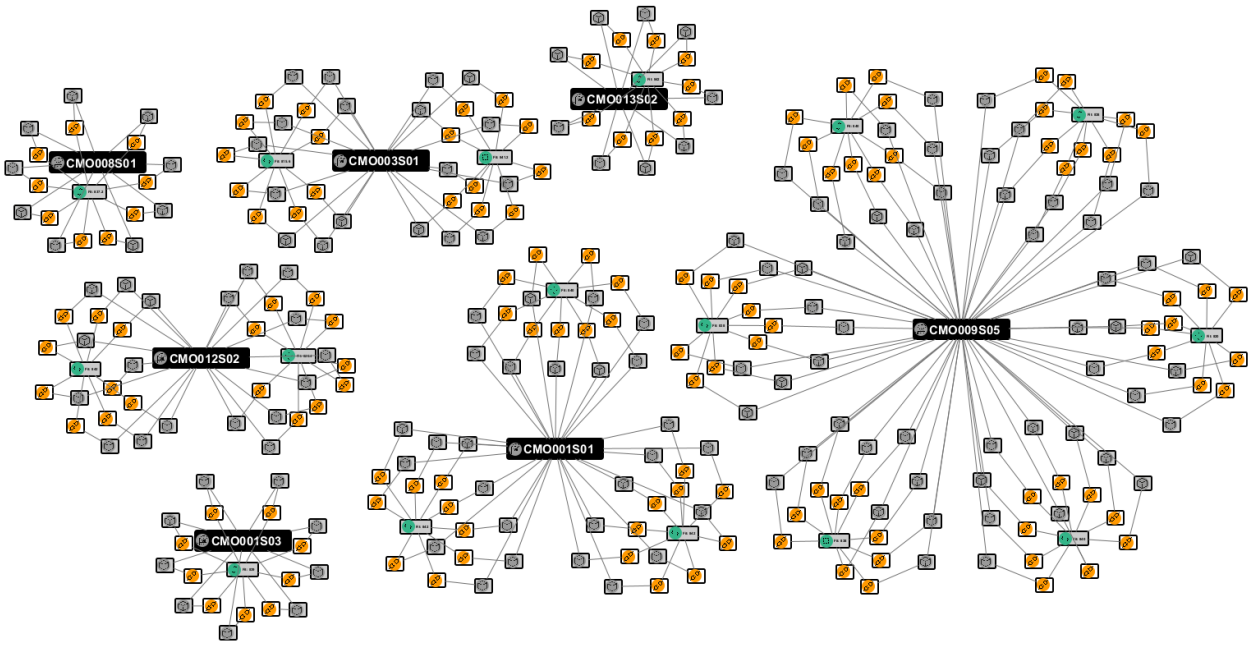
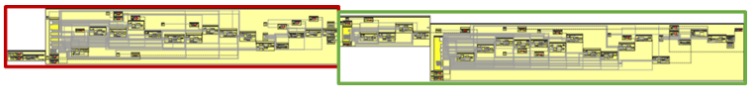


Figure 16: Visual Illustration of Manufacturing Solutions For Albuterol 2.

In contrast, with the same manufacturing objective, Figure 16 shows that Albuterol 2 has 7 qualified sites. Not only are there more qualified assets per qualified sites, each site also has multiple solutions, allowing for more flexible asset selection. This is not possible in Albuterol 1 (Figure 15) because there is a limited number of qualified asset for one or more equipment at every site. For example, across all 4 sites in Albuterol 1, there are only 3 assets that can satisfy the requirements of the required filter in PC3. As these 3 qualified filter assets are of the same specifications, there is actually only 1 "best" manufacturing solution per site. In Albuterol 2, however, there are sites sites with more than 1 solutions due to having qualified assets of different specifications. The prime example would be CMO009S05 or CMO_9S_5, which has 6 manufacturing solutions of the same "goodness" score. While there are only 9 filters in CMO_9S_5 that can satisfy the filter in PC1a (Figure 17), there are 6 different types of filter assets with comparable "goodness" score. This results in 6 different manufacturing solutions with similar "goodness" score in one site. In contrast, there is only 1 solution in CMO_8S_1 as there is only 1 filter asset that can satisfy the filtration

requirements in PC1a (Figure 17). Overall, there is not a big difference between the amount of qualified assets between the same equipment archetype in Albuterol 2. The biggest difference lies in the amount of qualified TMRC between PC1 (151 assets) and PC2 (120 assets) due to the strict MoC requirements caused by using hydrobromic acid in PC2[197], albeit the numbers themselves are still higher than the ones in 15). Finally, the difference in the number qualified sites between the two routes can also be attributed to the difference in volume requirements, as Albuterol 2 does not require as much containment volume because of the absence of workups and the presence of multiple distillation in between steps[197].



Equipment Archetype	Process Cell	Qualified Asset Instances	CMO_1S_1	CMO_1S_3	CMO_3S_1	CMO_8S_1	CMO_9S_5	$CMO_{12}S_2$	$CMO_{13}S_2$
Dryer	PC1a	77	19	10	7	6	25	4	6
Filter	PC1a	57	3	6	3	1	9	32	3
Temperature Managed Reactor Condenser	PC1a	151	27	5	14	31	47	24	3
Temperature Managed Reactor	PC1b	173	32	5	17	33	57	26	3
Dryer	PC2a	71	15	9	6	6	25	4	6
Filter	PC2a	76	8	9	4	4	12	33	6
Temperature Managed Reactor Condenser	PC2a	120	28	2	14	19	44	10	3
Vessel	PC2b	551	261	6	24	133	65	26	36
	Total	1276	393	52	89	233	284	159	66

Figure 17: Auto-Allocation Results of Albuterol 2.

Summary

Table 2 provides a high-level summary of the two Albuterol routes in terms of requirements and auto-allocation results. Overall, manufacturing Albuterol by following Albuterol 1 requires two times the amount of steps compared to Albuterol 2 (99 vs. 50 steps). As a result, Albuterol 1 requires two times the amount of equipment that Albuterol 2 needs. However, this 2:1 ratio flips as almost twice the amount of sites can accommodate the manufacturing requirements of Albuterol 2 than that of Albuterol 1. Consequentially, there are more tollers that can manufacture Albuterol using Albuterol 2 (6 vs. 4 tollers). Finally, this ratio converges towards parity at the end, where

the number of qualified assets for each route is approximately equal (about 4% difference). Thus, we can conclude that Albuterol 2 is a more accessible synthesis route than Albuterol 1 for the manufacturing of Albuterol

Table 2: Summary Table Comparing Heuristics with Auto-allocation Results Between Albuterol 1 and Albuterol 2.

Synthesis Route	**Steps**	**Equipment Archetypes**	**Sites**	**Tollers**	**Assets**
Albuterol 1	99	15	4	4	1325
Albuterol 2	50	8	7	6	1276

This work is the first to demonstrate a rapid, holistic, and systematic approach to estimating the manufacturability of APIs, with auto-allocation results that depend on editable stakeholder objectives, as shown in the MBSE approach in Figure 5. Combined with simulation authoring, this foundation has significant potential for future expansion and enhancement. Subsequent work can incorporate more complex stakeholder objectives related to environmental, social, governance, and regulatory aspects, offering richer and more realistic solutions applicable to the pharmaceutical manufacturing sector.

Conclusion

We establish the discipline of systems engineering (SE), specifically model-based systems engineering (MBSE), as a source of untapped potential in chemicals and materials manufacturing and supply chain management. We start with an overview of the U.S. chemical industry, identify the vulnerability in the current chemical supply chain and introduce the emergence of supply chain economics in chemical manufacturing. We then provide a detailed review on the discipline of SE from which MBSE emerges. Furthermore, we explore the practice of Process Systems Engineering (PSE) by reviewing the model-based approach in PSE, identify the current challenges in PSE, and how MBSE can be used to overcome these challenges. This transition into a review of MBSE where we explain the concept of "Systems Architecture" with its various modeling languages and frameworks. By leveraging our understanding of MBSE methodologies, we've developed the Authoritative source of Systems Knowledge (ASK) as the specific MBSE method suited for chemical manufacturing. To demonstrate its capabilities, we've utilized ASK to provide digital mappings of materials, process, equipment and requirements to manufacture API Atropine and API Albuterol. Finally, we've displayed the possible digital tools that can be integrated underneath digital architectural models, such as simulation authoring to calculate cycle time for Atropine and automatic-allocation of qualified assets to manufacture different syntheses of Albuterol.

References

1. Muller G. Systems engineering research methods. Procedia Computer Science. 2013;16:1092–1101.

2. Huldt T, Stenius I. State-of-practice survey of model-based systems engineering. Systems engineering. 2019;22(2):134–145.

3. Lerat JP. 5.5. 2 Three Reasons why Document-based SE (usually) works better than (most of) MBSE. In: INCOSE International Symposium;20:723–738. Wiley Online Library, 2010.

4. Klatt KU, Marquardt W. Perspectives for process systems engineering—Personal views from academia and industry. Computers & Chemical Engineering. 2009;33(3):536–550.

5. Herzig SJ, Qamar A, Reichwein A, Paredis CJ. A conceptual framework for consistency management in model-based systems engineering. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference;54792:1329–1339, 2011.

6. American Chemistry Council . 2019 Guide to the Business of Chemistry. `https://www.americanchemistry.com/chemistry-in-america/data-industry-statistics/resources/2019-guide-to-the-business-of-chemistry`, 2019.

7. U.S. Department of Homeland Security . Chemical Sector-Specific Plan - 2015. `https://www.cisa.gov/sites/default/files/publications/nipp-ssp-chemical-2015-508.pdf`, 2015.

8. International Organization for Standardization . ISO/IEC/IEEE 24748-1:2018(E) International Standard - Systems and software engineering - Life cycle management - Part 1: Guidelines for life cycle management. `https://www.iso.org/standard/72896.html`, 2018.

9. Marquardt W, Wedel L, Bayer B. Perspectives on lifecycle process modeling. In: AIChE Symposium no. 323 in AIChE Symposium Series:192-214. AIChE, 2000.

10. Tavallali MS, Karimi I, Baxendale D. Process systems engineering perspective on the planning and development of oil fields. AIChE Journal. 2016;62(8):2586–2604.

11. Royce WW. Managing the development of large software systems: concepts and techniques. In: Proceedings of the 9th international conference on Software Engineering:328–338, 1987.

12. Kleinekorte J, Fleitmann L, Bachmann M, *et al.* Life cycle assessment for the design of chemical processes, products, and supply chains. Annual review of chemical and biomolecular engineering. 2020;11:203–233.

13. Hahn A. Op-ed: The coronavirus pandemic underscores the need to bring drug manufacturing back to U.S.. `https://www.cnbc.com/2020/09/10/op-ed-the-coronavirus-pandemic-underscores-the-need-to-bring-drug-manufacturing-back-to-u` `html`, 2020.

14. U.S. White House . Building Resilient Supply Chains, Revitalizing American Manufacturing, and Fostering Broad-Based Growth. `https://www.whitehouse.gov/wp-content/uploads/2021/06/100-day-supply-chain-review-report.pdf?utm_source=sfmc%E2%80%8B&utm_medium=email%E2%80%8B&utm_campaign=20210610_Global_Manufacturing_Economic_Update_June_Members`, 2021.

15. U.S. Food and Drug Administration . Drug Shortages: Root Causes and Potential Solutions. `https://www.fda.gov/media/132058/download`, 2019.

16. U.S. National Science & Technology Council . Strategy For American Leadership In Advanced

Manufacturing. https://trumpwhitehouse.archives.gov/wp-content/uploads/2018/10/Advanced-Manufacturing-Strategic-Plan-2018.pdf, 2018.

17. Von Bertalanffy L. An outline of general system theory.. British Journal for the Philosophy of science. 1950.

18. Xu LD. The contribution of systems science to information systems research. Systems Research and Behavioral Science: The Official Journal of the International Federation for Systems Research. 2000;17(2):105–116.

19. Adams KM, Hester PT, Bradley JM, Meyers TJ, Keating CB. Systems theory as the foundation for understanding systems. Systems Engineering. 2014;17(1):112–123.

20. Wiener N. Cybernetics or Control and Communication in the Animal and the Machine. MIT press, 2019.

21. Ashby WR. An Introduction to Cybernetics. Chapman & Hall Ltd., 1957.

22. Beer S. Cybernetics and management. Journal of Symbolic Logic. 1960;25(3).

23. Forrester JW. Industrial dynamics. Journal of the Operational Research Society. 1997;48(10):1037–1041.

24. Casati F, Ilnicki S, Jin L, Krishnamoorthy V, Shan MC. Adaptive and dynamic service composition in eFlow. In: Advanced Information Systems Engineering: 12th International Conference, CAiSE 2000 Stockholm, Sweden, June 5–9, 2000 Proceedings 12:13–31. Springer, 2000.

25. Checkland P. Soft Systems Methodology: A 30 year Retrospective. Systems Thinking. Systems practice. 1999:A1–A66.

26. Meadows DH. Thinking in systems: A primer. Chelsea Green Publishing, 2008.

27. Sillitto H, Martin J, McKinney D, *et al.* Systems engineering and system definitions. In: INCOSE, 2019.

28. Gibson J, Scherer W, Gibson W, Smith M. How To Do Systems Analysis: Casebook and Primer. , 2016.

29. Buede D. The engineering design of systems: Methods and models. Hoboken: Wiley. 2000.

30. U.S. Department of Defense . Defense Acquisition Guidebook. `https://www.dau.edu/tools/dag`.

31. U.S. Department of Defense (DoD) . DOD INSTRUCTION 5000.88 ENGINEERING OF DEFENSE SYSTEMS. `https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500088p.PDF`, 2020.

32. Madni AM, Sievers M. Model-based systems engineering: motivation, current status, and needed advances. In: Disciplinary convergence in systems engineering research:311–325. Springer, 2018.

33. Hecht M. Use of SysML to generate failure modes and effects analyses for microgrid control systems. Insight. 2020;23(2):21–31.

34. Friedland B, Malone R, Herrold J. Systems engineering a model based systems engineering tool suite: The Boeing approach. In: INCOSE International Symposium;26:386–398. Wiley Online Library, 2016.

35. Kaslow D, Soremekun G, Kim H, Spangelo S. Integrated model-based systems engineering (MBSE) applied to the Simulation of a CubeSat mission. In: 2014 IEEE Aerospace Conference:1–14. IEEE, 2014.

36. Karban R, Andolfato L, Bristow P, *et al.* Model based systems engineering for astronomical projects. Modeling, Systems Engineering, and Project Management for Astronomy VI. 2014;9150:208–222.

37. Spangelo SC, Cutler J, Anderson L, *et al.* Model based systems engineering (MBSE) applied to Radio Aurora Explorer (RAX) CubeSat mission operational scenarios. In: 2013 IEEE Aerospace Conference:1–18. IEEE, 2013.

38. Albers A, Scherer H, Bursac N, Rachenkova G. Model based systems engineering in construction kit development–two case studies. Procedia CIRP. 2015;36:129–134.

39. Taub AI, Luo AA. Advanced lightweight materials and manufacturing processes for automotive applications. Mrs Bulletin. 2015;40(12):1045–1054.

40. Xu Z, Dang Y, Munro P. Knowledge-driven intelligent quality problem-solving system in the automotive industry. Advanced Engineering Informatics. 2018;38:441–457.

41. Evin E, Uludağ Y. Bioanalytical device design with model-based systems engineering tools. IEEE Systems Journal. 2020;14(3):3139–3149.

42. Bennion EP, Ginosar DM, Moses J, Agblevor F, Quinn JC. Lifecycle assessment of microalgae to biofuel: comparison of thermochemical processing pathways. Applied Energy. 2015;154:1062–1071.

43. Liu H, Lu T. Autonomous production of 1, 4-butanediol via a de novo biosynthesis pathway in engineered Escherichia coli. Metabolic Engineering. 2015;29:135–141.

44. Matar M, Osman H, Georgy M, Abou-Zeid A, El-Said M. Evaluation of civil infrastructure sustainability: A Model-Based Systems Engineering (MBSE) approach. In: ECPPM:327–334, 2014.

45. Scott W, Fullalove R, Arabian G, Campbell P. Case study: A model based systems engineering (mbse) framework for characterising transportation systems over the full life cycle. In: INCOSE International Symposium;26:916–932. Wiley Online Library, 2016.

46. Mažeika D, Butleris R. MBSEsec: Model-based systems engineering method for creating secure systems. Applied Sciences. 2020;10(7):2574.

47. Bayuk JL, Horowitz BM. An architectural systems engineering methodology for addressing cyber security. Systems Engineering. 2011;14(3):294–304.

48. Topper JS, Horner NC. Model-based systems engineering in support of complex systems development. Johns Hopkins APL technical digest. 2013;32(1).

49. Robinson K, Tramoundanis D, Harvey D, Jones M, Wilson S. Demonstrating model-based systems engineering for specifying complex capability. In: Systems Engineering/Test & Evaluation Conference, 2010.

50. Piaszczyk C. Model based systems engineering with department of defense architectural framework. Systems Engineering. 2011;14(3):305–326.

51. Soyler A, Sala-Diakanda S. A model-based systems engineering approach to capturing disaster management systems. In: 2010 IEEE International Systems Conference:283–287. IEEE, 2010.

52. Lambert JH, Parlak AI, Zhou Q, *et al.* Understanding and managing disaster evacuation on a transportation network. Accident Analysis & Prevention. 2013;50:645–658.

53. Martí JR, Hollman JA, Ventura C, Jatskevich J. Dynamic recovery of critical infrastructures: real-time temporal coordination. International Journal of Critical Infrastructures. 2008;4(1-2):17–31.

54. Lopes AJ, Lezama R, Pineda R. Model based systems engineering for smart grids as systems of systems. Procedia Computer Science. 2011;6:441–450.

55. Rahman MM, Oni AO, Gemechu E, Kumar A. Assessment of energy storage technologies: A review. Energy Conversion and Management. 2020;223:113295.

56. Thorisson H, Lambert JH, Cardenas JJ, Linkov I. Resilience analytics with application to power grid of a developing region. Risk Analysis. 2017;37(7):1268–1286.

57. Kübler K, Scheifele S, Scheifele C, Riedel O. Model-based systems engineering for machine tools and production systems (model-based production engineering. Procedia Manufacturing. 2018;24:216–221.

58. Steimer C, Fischer J, Aurich JC. Model-based design process for the early phases of manufacturing system planning using SysML. Procedia CIRP. 2017;60:163–168.

59. Li S, El-Mounayri H, Zhang W, Schindel B, Sherey J. Implementation of systems engineering model into product lifecycle management platform. In: IFIP International Conference on Product Lifecycle Management:601–608. Springer, 2015.

60. Lee D, Jayaraman A, Kwon JSI. Identification of cell-to-cell heterogeneity through systems engineering approaches. AIChE Journal. 2020;66(5):e16925.

61. Floudas CA. Research challenges, opportunities and synergism in systems engineering and computational biology. AIChE Journal. 2005;51(7):1872–1884.

62. Colombo KWE, Kharton VV. Reliability analysis of a multi-stack solid oxide fuel cell from a systems engineering perspective. Chemical Engineering Science. 2021;238:116571.

63. Agrawal R. Chemical engineering for a solar economy (2017 PV Danckwerts Lecture). Chemical Engineering Science. 2019;210:115215.

64. Burgess AA, Brennan DJ. Application of life cycle assessment to chemical processes. Chemical Engineering Science. 2001;56(8):2589–2604.

65. Navas J, Tannery P, Bonnet S, Voirin JL. Bridging the gap between model-based systems engineering methodologies and their effective practice–a case study on nuclear power plants systems engineering. Insight. 2018;21(1):17–20.

66. Stork D, Agostini P, Boutard JL, *et al.* Developing structural, high-heat flux and plasma facing materials for a near-term DEMO fusion power plant: the EU assessment. Journal of nuclear materials. 2014;455(1-3):277–291.

67. Marwick AD. Knowledge management technology. IBM systems journal. 2001;40(4):814–830.

68. INCOSE . The Systems Engineering Tools Database Working Group. `https://www.incose.org/incose-member-resources/working-groups/transformational/se-tools-database`, 2022.

69. Dassault Systèmes . SysML Plugin. `https://www.3ds.com/products-services/catia/products/no-magic/addons/sysml-plugin/`.

70. IBM . IBM Engineering Systems Design Rhapsody – Architect for Systems Engineers. `https://www.ibm.com/products/architect-for-systems-engineers`.

71. Sparx Systems . . `https://sparxsystems.com/products/ea/index.html`, 2023.

72. Eclipse Foundation . Capella MBSE Tool. `https://www.eclipse.org/capella/`, 2022.

73. Eclipse Foundation . Capella MBSE Tool - Arcadia. `https://www.eclipse.org/capella/arcadia.html`, 2023.

74. Dunbar B, Garrett S. 2.0 Fundamentals of Systems Engineering Ch. 2. National Aeronautics and Space Administration, 2020.

75. U.S. Department of Transportation . Systems Engineering for Intelligent Transportation Systems. https://ops.fhwa.dot.gov/publications/seitsguide/, 2007.

76. International Standard Organization (ISO) . What We Do. https://www.iso.org/what-we-do.html, 2022.

77. Walden DD, Roedler GJ, Forsberg K, Hamelin RD, Shortell TM. Systems engineering handbook: A guide for system life cycle processes and activities. John Wiley & Sons, 2015.

78. SEBoK . System Lifecycle Models. https://www.sebokwiki.org/wiki/System_Lifecycle_Models, 2022.

79. Forsberg K, Mooz H. The relationship of system engineering to the project cycle. In: INCOSE international symposium;1:57–65. Wiley Online Library, 1991.

80. Atlassian . Agile. https://www.atlassian.com/agile.

81. International Organization for Standardization . ISO/IEC/IEEE 15288(E) International Standard. Systems and software engineering - System life cycle processes. First Edition. https://www.iso.org/standard/63711.html, 2015.

82. Perkins J. Education in process systems engineering: past, present and future. Computers & chemical engineering. 2002;26(2):283–293.

83. Ponton J. Process Systems Engineering: Halfway through the first century. Chemical Engineering Science. 1995;50(24):4045–4059.

84. Takamatsu T. The nature and role of process systems engineering. Computers & chemical engineering. 1983;7:203–218.

85. Tapia JFD, Lee JY, Ooi RE, Foo DC, Tan RR. A review of optimization and decision-making models for the planning of CO2 capture, utilization and storage (CCUS) systems. Sustainable Production and Consumption. 2018;13:1–15.

86. Maravelias CT. General framework and modeling approach classification for chemical production scheduling. AIChE Journal. 2012;58(6):1812–1828.

87. Liu S, Shah N, Papageorgiou LG. Multiechelon supply chain planning with sequence-dependent changeovers and price elasticity of demand under uncertainty. AIChE journal. 2012;58(11):3390–3403.

88. Chemmangattuvalappil NG, Ng DK, Ng LY, Ooi J, Chong JW, Eden MR. A review of process systems engineering (PSE) tools for the design of ionic liquids and integrated biorefineries. Processes. 2020;8(12):1678.

89. Agrawal V, Shenoy UV. Unified conceptual approach to targeting and design of water and hydrogen networks. AIChE Journal. 2006;52(3):1071–1082.

90. Cervera-Padrell AE, Skovby T, Kiil S, Gani R, Gernaey KV. Active pharmaceutical ingredient (API) production involving continuous processes–a process system engineering (PSE)-assisted design framework. European journal of pharmaceutics and biopharmaceutics. 2012;82(2):437–456.

91. Yan L, Hu S, Ma D. Visual method for operating optimization of process system. In: Computer Aided Chemical Engineering;15:1394–1398. Elsevier, 2003.

92. Urich MD, Vemula RR, Kothare MV. Multivariable model predictive control of a novel rapid pressure swing adsorption system. AIChE Journal. 2018;64(4):1234–1245.

93. Troup GM, Georgakis C. Process systems engineering tools in the pharmaceutical industry. Computers & Chemical Engineering. 2013;51:157–171.

94. Ma DL, Braatz RD. Robust identification and control of batch processes. Computers & Chemical Engineering. 2003;27(8-9):1175–1184.

95. Iftakher A, Mansouri SS, Nahid A, *et al.* Integrated design and control of reactive distillation processes using the driving force approach. AIChE Journal. 2021;67(6):e17227.

96. Grossmann IE, Trespalacios F. Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming. AIChE Journal. 2013;59(9):3276–3295.

97. Pistikopoulos EN, Dua V, Bozinis NA, Bemporad A, Morari M. On-line optimization via off-line parametric optimization tools. Computers & Chemical Engineering. 2002;26(2):175–185.

98. Biegler LT, Grossmann IE. Retrospective on optimization. Computers & Chemical Engineering. 2004;28(8):1169–1192.

99. Schopfer G, Yang A, Wedel L, Marquardt W. CHEOPS: A tool-integration platform for chemical process modelling and simulation. International Journal on Software Tools for Technology Transfer. 2004;6:186–202.

100. Sarna S, Patel N, Corbett B, McCready C, Mhaskar P. Process-aware data-driven modelling and model predictive control of bioreactor for the production of monoclonal antibodies. The Canadian Journal of Chemical Engineering. 2022.

101. Subramanian ASR, Gundersen T, Adams TA. Modeling and simulation of energy systems: A review. Processes. 2018;6(12):238.

102. Gassner M, Maréchal F. Thermo-economic optimisation of the polygeneration of synthetic natural gas (SNG), power and heat from lignocellulosic biomass by gasification and methanation. Energy & Environmental Science. 2012;5(2):5768–5789.

103. Andika R, Nandiyanto ABD, Putra ZA, *et al.* Co-electrolysis for power-to-methanol applications. Renewable and Sustainable Energy Reviews. 2018;95:227–241.

104. Rafiee A, Khalilpour KR, Milani D, Panahi M. Trends in CO2 conversion and utilization: A review from process systems perspective. Journal of environmental chemical engineering. 2018;6(5):5771–5794.

105. Paim Á, Cardozo NS, Pranke P, Tessaro IC. Process system engineering methodologies applied to tissue development and regenerative medicine. Cutting-Edge Enabling Technologies for Regenerative Medicine. 2018:445–463.

106. Koppol AP, Bagajewicz MJ, Dericks BJ, Savelski MJ. On zero water discharge solutions in the process industry. Advances in Environmental Research. 2004;8(2):151–171.

107. Kano M, Nakagawa Y. Data-based process monitoring, process control, and quality improvement: Recent developments and applications in steel industry. Computers & Chemical Engineering. 2008;32(1-2):12–24.

108. Gebreslassie BH, Diwekar UM. Homogenous multi-agent optimization for process systems engineering problems with a case study of computer aided molecular design. Chemical Engineering Science. 2017;159:194–206.

109. Diab SA, Jolliffe HG, Gerogiorgis DI. Plantwide technoeconomic analysis and separation solvent selection for continuous pharmaceutical manufacturing: Ibuprofen, artemisinin, and diphenhydramine. In: Computer Aided Chemical Engineering;41:85–120. Elsevier, 2018.

110. McBride K, Sundmacher K. Computer-aided design of solvents for the recovery of a homogeneous catalyst used for alkene hydroformylation. In: Computer Aided Chemical Engineering;37:2075–2080. Elsevier, 2015.

111. Pernalete CG, Baten J, Urbina JC, Arévalo JF. A molecular reconstruction feed characterization and CAPE OPEN implementation strategy to develop a tool for modeling HDT reactors for light petroleum cuts. In: Computer Aided Chemical Engineering;37:359–364. Elsevier, 2015.

112. Chen J, Fang Z, Qiu T. Molecular reconstruction model based on structure oriented lumping and group contribution methods. Chinese journal of chemical engineering. 2018;26(8):1677–1683.

113. Villazón-León V, Bonilla-Petriciolet A, Tapia-Picazo J, Segovia-Hernández J, Corazza M. A review of group contribution models to calculate thermodynamic properties of ionic liquids for process systems engineering. Chemical Engineering Research and Design. 2022.

114. Nor MSM, Abd Manan Z, Mustaffa AA, Suan CL. An Evaluation of Thermodynamic Models for the Prediction of Solubility of Phytochemicals from Orthosiphon Staminues in Ethanol. In: Computer aided chemical engineering;37:2087–2092. Elsevier, 2015.

115. Siddhaye S, Camarda KV, Topp E, Southard M. Design of novel pharmaceutical products via combinatorial optimization. Computers & Chemical Engineering. 2000;24(2-7):701–704.

116. Ali SF, Yeung H. Experimental investigation and numerical simulation of two-phase flow in a large-diameter horizontal flow line vertical riser. Petroleum science and technology. 2010;28(11):1079–1095.

117. Ibrahim MT, Zacharias J, Briesen H, Först P. Heat transfer to a stationary cubic particle in

a laminar tube flow: Computational fluid dynamics simulations and experiments. Journal of Food Engineering. 2020;274:109833.

118. Tian Y, Demirel SE, Hasan MF, Pistikopoulos EN. An overview of process systems engineering approaches for process intensification: State of the art. Chemical Engineering and Processing-Process Intensification. 2018;133:160–210.

119. Buchholz S. Future manufacturing approaches in the chemical and pharmaceutical industry. Chemical Engineering and Processing: Process Intensification. 2010;49(10):993–995.

120. Grossmann I. Enterprise-wide optimization: A new frontier in process systems engineering. AIChE Journal. 2005;51(7):1846–1857.

121. Amorim P, Pinto-Varela T, Almada-Lobo B, Barbosa-Póvoa APF. Comparing models for lot-sizing and scheduling of single-stage continuous processes: Operations research and process systems engineering approaches. Computers & chemical engineering. 2013;52:177–192.

122. Lukszo Z, Weijnen MP, Negenborn RR, De Schutter B. Tackling challenges in infrastructure operation and control: cross-sectoral learning for process and infrastructure engineers. International journal of critical infrastructures. 2009;5(4):308–322.

123. Ajah A, Herder P, Grievink J, Weijnen M. Hierarchical markov reliability/availability models for energy & industrial infrastructure systems conceptual design. In: Computer aided chemical engineering;21:1753–1758. Elsevier, 2006.

124. Herder PM, Turk AL, Subrahmanian E, Westerberg AW. Challenges for process systems engineering in infrastructure design. Computers & Chemical Engineering. 2000;24(2-7):1775–1780.

125. Schneider R, Marquardt W. Information technology support in the chemical process design life cycle. Chemical engineering science. 2002;57(10):1763–1792.

126. Naka Y, Hirao M, Shimizu Y, Muraki M, Kondo Y. Technological information infrastructure for product lifecycle engineering. Computers & Chemical Engineering. 2000;24(2-7):665–670.

127. Kikuchi Y, Hirao M. Activity and Information Infrastructure for Risk-Based Process Design. In: Computer Aided Chemical Engineering;27:1023–1028. Elsevier, 2009.

128. Laínez JM, Schaefer E, Reklaitis GV. Challenges and opportunities in enterprise-wide optimization in the pharmaceutical industry. Computers & chemical engineering. 2012;47:19–28.

129. Avraamidou S, Baratsas SG, Tian Y, Pistikopoulos EN. Circular Economy-A challenge and an opportunity for Process Systems Engineering. Computers & Chemical Engineering. 2020;133:106629.

130. Sousa RT, Liu S, Papageorgiou LG, Shah N. Global supply chain planning for pharmaceuticals. Chemical engineering research and design. 2011;89(11):2396–2409.

131. Perea-Lopez E, Ydstie BE, Grossmann IE. A model predictive control strategy for supply chain optimization. Computers & Chemical Engineering. 2003;27(8-9):1201–1218.

132. Reis MS, Saraiva PM. Data-centric process systems engineering: A push towards PSE 4.0. Computers & Chemical Engineering. 2021;155:107529.

133. Pistikopoulos EN, Barbosa-Povoa A, Lee JH, *et al.* Process systems engineering–the generation next?. Computers & Chemical Engineering. 2021;147:107252.

134. Lee JH, Shin J, Realff MJ. Machine learning: Overview of the recent progresses and implications for the process systems engineering field. Computers & Chemical Engineering. 2018;114:111–121.

135. Shang C, You F. Data analytics and machine learning for smart process manufacturing: Recent advances and perspectives in the big data era. Engineering. 2019;5(6):1010–1016.

136. Sansana J, Joswiak MN, Castillo I, *et al.* Recent trends on hybrid modeling for Industry 4.0. Computers & Chemical Engineering. 2021;151:107365.

137. Jacquemin L, Pontalier PY, Sablayrolles C. Life cycle assessment (LCA) applied to the process industry: a review. The International Journal of Life Cycle Assessment. 2012;17:1028–1041.

138. Grossmann IE. Challenges in the new millennium: product discovery and design, enterprise and supply chain optimization, global life cycle assessment. Computers & Chemical Engineering. 2004;29(1):29–39.

139. Bakshi BR, Fiksel J. The quest for sustainability: Challenges for process systems engineering. AIChE journal. 2003;49(6):1350–1358.

140. Guillén-Gosálbez G, You F, Galán-Martín Á, Pozo C, Grossmann IE. Process systems engineering thinking and tools applied to sustainability problems: current landscape and future opportunities. Current Opinion in Chemical Engineering. 2019;26:170–179.

141. Batres R. Ontologies in process systems engineering. Chemie Ingenieur Technik. 2017;89(11):1421–1431.

142. Seidenberg JR, Khan AA, Lapkin AA. Boosting autonomous process design and intensification with formalized domain knowledge. Computers & Chemical Engineering. 2023;169:108097.

143. Morbach J, Yang A, Marquardt W. OntoCAPE—A large-scale ontology for chemical process engineering. Engineering applications of artificial intelligence. 2007;20(2):147–161.

144. Brandt SC, Morbach J, Miatidis M, Theißen M, Jarke M, Marquardt W. An ontology-based

approach to knowledge management in design processes. Computers & Chemical Engineering. 2008;32(1-2):320–342.

145. Paszke A, Gross S, Massa F, *et al.* Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems. 2019;32.

146. Shobrys DE, White DC. Planning, scheduling and control systems: why cannot they work together. Computers & chemical engineering. 2002;26(2):149–160.

147. Cuenca J, Larrinaga F, Curry E. DABGEO: A reusable and usable global energy ontology for the energy domain. Journal of Web Semantics. 2020;61:100550.

148. Cuenca J, Larrinaga F, Curry E. MODDALS methodology for designing layered ontology structures. Applied Ontology. 2020;15(2):185–217.

149. Lindheim C, Totland T, Lien K. Enterprise modeling a new task for process systems engineers?. Computers & chemical engineering. 1996;20:S1527–S1532.

150. INCOSE T. Systems engineering vision 2020. INCOSE, San Diego, CA, accessed Jan. 2007;26(2019):2.

151. Holt J, Perry S, Payne R, Bryans J, Hallerstede S, Hansen FO. A model-based approach for requirements engineering for systems of systems. IEEE Systems Journal. 2104;9(1):252–262.

152. Nielsen CB, Larsen PG, Fitzgerald J, Woodcock J, Peleska J. Systems of systems engineering: basic concepts, model-based techniques, and research directions. ACM Computing Surveys (CSUR). 2015;48(2):1–41.

153. Logan P, Harvey D, Spencer D. Documents are an essential part of model based systems engineering. In: INCOSE International Symposium;22:1899–1913. Wiley Online Library, 2012.

154. Dickerson CE, Mavris D. A brief history of models and model based systems engineering and the case for relational orientation. IEEE Systems Journal. 2013;7(4):581–592.

155. Wymore AW. Model-based systems engineering. CRC press, 1993.

156. Estefan JA. Survey of model-based systems engineering (MBSE) methodologies. Incose MBSE Focus Group. 2007;25(8):1–12.

157. MacCalman A, Lesinski G, Goerger S. Integrating external simulations within the model-based systems engineering approach using statistical metamodels. Procedia Computer Science. 2016;95(436–441).

158. Bjorkman EA, Sarkani S, Mazzuchi TA. Using model-based systems engineering as a framework for improving test and evaluation activities. Systems Engineering. 2013;16(3):346–362.

159. Acheson P, Dagli C, Kilicay-Ergin N. Model based systems engineering for system of systems using agent-based modeling. Procedia Computer Science. 2013;16:11–19.

160. Peterson T, Schindel B. Explicating System Value through First Principles: Re-Uniting Decision Analysis with Systems Engineering. In: INCOSE International Symposium;26:74–89. Wiley Online Library, 2016.

161. Bleakley G, Lapping A, Whitfield A. 6.6. 2 Determining the right solution using sysml and model based systems engineering (MBSE) for trade studies. In: INCOSE International Symposium;21:783–795. Wiley Online Library, 2011.

162. Russell M. Using MBSE to enhance system design decision making. Procedia Computer Science. 2012;8:188-193.

163. Mahboob A, Weber C, Husung S, Liebal A, Krömker H, others . Model based systems engineering (MBSE) approach for configurable product use-case scenarios in virtual environments.

In: DS 87-3 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 3: Product, Services and Systems Design, Vancouver, Canada, 21-25.08. 2017:281–290, 2017.

164. Madni AM, Madni CC, Lucero SD. Leveraging digital twin technology in model-based systems engineering. Systems. 2019;7(1):7.

165. Schluse M, Atorf L, Rossmann J. Experimentable digital twins for model-based systems engineering and simulation-based development. In: 2017 annual ieee international systems conference (syscon):1–8. IEEE, 2017.

166. Gass SI. Decision-aiding models: validation, assessment, and related issues for policy analysis. Operations Research. 1983;31(4):603–631.

167. Cloutier R, Sauser B, Bone M, Taylor A. Transitioning systems thinking to model-based systems engineering: Systemigrams to SysML models. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2014;45(4):662–674.

168. Collier ZA, Lambert JH. Principles and methods of model validation for model risk reduction. Environment systems and decisions. 2019;39(2):146–153.

169. Clash. `http://clash-lang.org`, 2022.

170. SysML.org . SysML Open Source Project. `https://sysml.org/`, 2021.

171. Noran O. UML vs IDEF: An ontology-based comparative study in view of business modelling. In: Proc. 6th In International Conference on Enterprise Information Systems, Porto, Portugal;3:674–682, 2003.

172. Bernus P, Schmidt G. Architectures of information systems. In: Handbook on architectures of information systems:1–9. Springer, 1998.

173. Eriksson HE, Penker M. Business modeling with UML. New York. 2000;12.

174. Friedenthal S, Moore A, Steiner R. A practical guide to SysML: the systems modeling language. Morgan Kaufmann, 2014.

175. Object Management Group (OMG) . SysML Partners: Creators of the SysML. https://sysml.org/sysml-partners/, 2022.

176. Vaneman WK. Evolving Model-Based Systems Engineering Ontologies and Structures. In: INCOSE International Symposium;28:1027–1036. Wiley Online Library, 2018.

177. International Standard Organization (ISO) . OMG Systems Modeling Language (OMG SysML™) Tutorial. https://www.omgsysml.org/INCOSE-OMGSysML-Tutorial-Final-090901.pdf, 2009.

178. Object Management Group (OMG) . Systems Modeling Language (SysML®) v2 Request For Proposal (RFP). https://www.omg.org/cgi-bin/doc.cgi?ad/2017-12-2, 2017.

179. SysML v2 Submission Team . SysML v2 Release. https://github.com/Systems-Modeling/SysML-v2-Release, 2022.

180. Vaneman WK. Enhancing model-based systems engineering with the Lifecycle Modeling Language. In: 2016 Annual IEEE Systems Conference (SysCon):1–7. IEEE, 2016.

181. Gebhart M, Baumgartner M, Oehlert S, Blersch M, Abeck S. Evaluation of service designs based on soaml. In: 2010 Fifth International Conference on Software Engineering Advances:7–13. IEEE, 2010.

182. Brown MT. A picture is worth a thousand words: energy systems language and simulation. Ecological Modelling. 2004;178(1-2):83–100.

183. Logan P, Solutions E, Morris B, Harvey D, Gordon LL, Navy RA. Model-Based Systems Engineering Metamodel: Roadmap for Effective Systems Engineering Process. PhD thesisSapro 2013.

184. World Wide Web Consortium (W3C) . W3. https://www.w3.org/, 2022.

185. Schindel WD. 1.4. 2 What Is the Smallest Model of a System?. In: INCOSE International Symposium;21:99–113. Wiley Online Library, 2011.

186. Lempia D, Schindel B, Hrabik T, McGill S, Graber M. Using visual diagrams and patterns for consistent and complete requirements. In: INCOSE International Symposium;26:415–429. Wiley Online Library, 2016.

187. Kline WA, Schindel WD. Engineering design, a shift from a process to a model-based view. In: 2017 IEEE Frontiers in Education Conference (FIE):1–3. IEEE, 2017.

188. Lopes F, Hervé . Process for Preparation of Atropine. , 2016. WIPO WO/2016/016692.

189. Scallan P. Process planning: the design/manufacture interface. Elsevier, 2003.

190. Schindel W, Peterson T. An Overview of Pattern-Based Systems Engineering (PBSE): Leveraging MBSE Techniques. In: INCOSE Enchantment Chapter Webinar, 2014.

191. McCabe WL, Smith JC, Harriott P. Unit operations of chemical engineering (6th edition). McGraw-Hill New York, 2000.

192. Johnson DB, Merrell BJ, Bounds CG. Albuterol. In: StatPearls [Internet]. StatPearls Publishing, 2022.

193. The American Society of Health-System Pharmacists . Albuterol. https://medlineplus.gov/druginfo/meds/a607004.html, 2023.

194. U.S. Food & Drug Administration . FDA Drug Shortages. https://www.accessdata.fda.gov/scripts/drugshortages/default.cfm, 2023.

195. Becker Z. Bankrupt Akorn Pharma calls it quits and closes all US sites, laying off entire workforce. , 2023.

196. Babad E, Carruthers NI, Jaret RS, Steinman M. A short synthesis of albuterol. Synthesis. 1988;1988(12):966–968.

197. Tann CH, Thiruvengadam T, Chiu J, *et al.* Process for preparing albuterol, acetal, hemiacetal, and hydrates of arylglyoxal intermediates thereof. , 1994. US Patent 5,283,359.

198. Central States Industrial . Cut the Hassle: Hastelloy C-22 vs. C-276. https://www.csidesigns.com/blog/articles/cut-the-hassle-hastelloy-c-22-vs-c-276, 2016.

199. ClinCalc LLC . Albuterol Drug Usage Statistics, United States, 2013 - 2020. https://clincalc.com/DrugStats/Drugs/Albuterol, 2021.