Synthesis of Batch Processing Schemes for the Production of Pharmaceuticals and Specialty Chemicals

by

Shahin A. Ali

Submitted to the Department of Chemical Engineering in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Chemical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1999

Author...

Department of Chemical Engineering

November 4, 1998

Certified by...

George Stephanopoulos

A. D. Little Professor of Chemical Engineering

Thesis Supervisor

Accepted by...

Robert Cohen

St. Laurent Professor of Chemical Engineering

Chairman, Committee on Graduate Students

Synthesis of Batch Processing Schemes for the Production of Pharmaceuticals and Specialty Chemicals

by

Shahin A. Ali

Submitted to the Department of Chemical Engineering
on November 4, 1998, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Chemical Engineering

Abstract

While the synthesis of continuous processes has been advanced to significant levels of effectiveness, through pure or hybrid implementations of rigorous optimization-based and heuristic approaches, corresponding progress in the synthesis of batch processing schemes has been lagging behind. Most of the research in the batch area has focused on the design of multipurpose or dedicated batch plants, and the optimal planning and scheduling of batch operations in multipurpose plant configurations.

Unlike a continuous plant, which is composed of a well-defined network of (mostly) single-function unit operations, a batch process is deployed through a series of batch-operating vessels, which accommodate varying sets of physico-chemical transformations of processing materials. As a result, while the synthesis of a continuous plant leads to a structured flowsheet of unit operations, the synthesis of a batch processing scheme leads to a structured sequence of operating steps. Consequently, we have approached the synthesis of batch processing schemes as a problem of synthesis of operating procedures. Such an *operations-centered synthesis* decouples the selection of equipment and allows efficient solution of a rather cumbersome combinatorial optimization problem, while allowing the synthesis of novel processes through the grouping of varying sets of operations in the same equipment.

The methodology we have developed for selecting and ordering the operations to be performed is based on a combination of *Means-Ends Analysis* (MEA) and *Non-Monotonic Planning* (NMP). This approach uses means-ends analysis to detect differences between the current process state and the desired product state. When a difference is detected, we apply a non-monotonic planning methodology, which incorporates a combination of heuristic and quantitative methods to construct a plan consisting of operations capable of resolving the difference. The first step is to determine a set of feasible operations that will eliminate the difference. This set of operations is known as the *Task Category*. The operations are grouped into Task Categories based on the attributes of the tasks they perform. These Task Categories allow us to narrow the number of alternatives that need to be examined. From within the Task Category, a single operation is chosen to be incorporated into the design while all other operations in the set are retained as possible alternatives. In order to determine the applicability of the operation, the preconditions of the operation are assessed against the current process state. If any of the preconditions is violated, operations called "White Knights" are selected and applied before the current operation to alter the current state and remove the precondition violation. The elimination of the precondition violations through the non-monotonic planning routine leads to a feasible ordering of operational steps which describe aspects of the evolving processing scheme. After all the precondition violations have been removed, the sequence of operations that has been generated is applied and a new current process state is generated. The *Means-Ends Analysis with Nonmonotonic Planning* (MEA-NMP) methodology is iteratively applied to resolve remaining differences between the new current states and the final product state. The MEA-NMP approach provides us with base case design, as well as, the search space of feasible alternatives by linking the Task Categories together to form the process *superstructure*. Since the alternatives have been maintained through the identification of the Task Categories, the problem can be formulated as a MINLP and a common MINLP solution strategy can be applied to determine the optimal design.

This thesis describes both the MEA-NMP strategy, as well as, its methodological details. We will discuss how the combinatorial optimization problem, defining the synthesis of batch processing schemes, is decomposed into (a) a logic-based component, defining the feasible processing alternatives, and implemented through the MEA-NMP strategy, and (b) a reduced MINLP formulation whose solution identifies the optimal processing scheme. In addition, we will discuss the detailed mathematical formulation of the batch process synthesis problem using the *finite automaton* and the *State Task Network*, the models employed for the representation of Tasks, the logic that guides the selection of White Knights, and the metrics used for the evaluation of processing alternatives. In the thesis we also outline the computer-aided elements which implement the above ideas within the framework of the *BatchDesign-Kit*, and will illustrate the application for the MEA-NMP strategy on realistic case studies pertaining to the synthesis of processing schemes for the manufacturing of pharmaceuticals. Finally, we discuss the conclusions and contributions attributed to this work and possible directions for future research that have been brought to our attention during the development of this thesis.

Thesis Supervisor: George Stephanopoulos

Title: A. D. Little Professor of Chemical Engineering

Acknowledgements

Before presenting this thesis, it is only appropriate that I dedicate a few lines to those who have influenced and supported me during my endeavor.

When I began my research project, Professor Stephanopoulos told me that I would learn a lot from this project and it would not be a traditional Ph.D. research project. In both regards, he was right. Along with the learning that came from performing research, I was also part of a team which had developed a research project and carried it through to see its commercialization. Through this project I have had the chance to play a number of roles in order to get the job done. This experience has been invaluable and I would like to thank Professor George Stephanopoulos for making me an integral part of it. It will always be a great source of pride for me.

Next, I would like to thank my thesis committee members, Professor James M. Douglas, Professor Charles L. Cooney, Professor Paul I. Barton and Professor C. Michael Mohr, for their guidance. I must give special thanks to Professor Douglas for two reasons, first it was Professor Douglas's Introduction to Chemical Engineering course back at UMass that inspired me to become a chemical engineer, without his inspiration this thesis may well not have been undertaken and second, for making all those trips from UMass to MIT to attend my committee meetings.

While on the subject of UMass I would like to thank my buddies, Vais, Dubé, Bobby B., Platts, Kai, Randy, Hurley and Eric. I have often looked back on all those long nights we spent together in the ChemE room with great fondness, while sitting in the trenches of MIT. I would especially like to thank Dave and Dave, the things you have taught me about life are much more valuable than anything which could be learned in a classroom. To Liz, I'm glad you chose to do your Ph.D in Boston, aside from watching hockey and basketball, I'm happy to have you as a friend.

Thanks to the ERRC consortium and the ONR for sponsoring this research.

There are a number of people at MIT to which I owe thanks. To my office mates past and present I would also like to express my thanks for making the work environment a fun place to be. Thanks to Christine and Orhan for being my computer gurus and to Matthew for filling in the gaps in my knowledge of mathematics. Thanks to Gabrielle for all her help. Thanks to Howard "Mr. Chemical Engineering" Covert for all the questions you have answered for me over the years and for "letting" me graduate before you. Thanks to John for helping me keep my sanity in the weight room.

There are two people who deserve most of the credit for helping me maintain my sanity during this long journey and I could never thank them enough for being such good friends. To Daphne, I want to thank you for supporting and encouraging me over all these years. Through all the good and difficult times you were always there for me and I could never thank you enough. To Talid, well what can I say, a better friend I could never find.

I want to thank some of the new friends I have made in the soccer leagues around Boston. Aside from becoming friends, playing soccer with you was truly enjoyable and provided me with a way to kick back and relieve the stress associated with working on a project of this scope. I should also thank my friends in Western Mass. for their kindness and generosity, as well as for providing me with an escape from life in the city. We have had many good times together and I'm sure there will be many more.

There are a few people who I would like to thank for there support and for showing so much pride in me. To Aunt Christine, Fred and my extended family, Oma, Opa, Erika, Herman, Egon, Heidi, Sigfried, Sonja, Sigfried and all my cousins, thanks for all your caring and pride. Also. thanks to my friends Mark, Steve, Jim, John, Ed, Paul, Tom and Linda, the pride you have shown in me will never be forgotten.

Finally, I would like to thank my mother, father and Sonja, without your support and encouragement I would have never completed this dissertation.

"Our doubts are our traitors, and make us lose
the good we oft might win by fearing to attempt"
- Shakespeare, Measure for Measure

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Within the chemical industry there are two basic modes of chemical production, *batch* and *continuous*. While batch production is by far the more widely used and the much older of the two, continuous processes have, until recently, dominated research and the literature. Over the last 10 – 15 years we have been witnessing a change in this trend. With global markets and a consumer driven market place, the demand of high value-added pharmaceuticals and specialty chemicals has risen sharply. These types of products are naturally suited for batch manufacturing due to factors such as, strict quality constraints, small production volumes, short product life, a large variety of products, and short time-to-market, once the target product has been identified. Researchers have responded to industry's renewed interest in the area of batch processing by increasing their efforts in this area. This trend is clearly visible through the reviews of the literature in the area of batch processing. In 1983, Rippin [1] reviewed material available in the literature concerning batch processing and was able to identify nine references. By 1989, Reklaitis [2] was able to list fifty-one sources in his review and by 1993, in a second review, Rippin [3] was able to list eighty-nine sources. Finally, in 1997, Allgor [4] in his Ph.D. thesis, listed over 100 references dedicated to batch processes. Thus, it is apparent that the academic community is becoming more and more interested in the area of batch processing.

The surge of interest in batch processing has led to work in all aspects of batch processing. There are many studies discussing one or more of these aspects such as, the *scheduling* of batch processes, the use of *intermediate storage*, the *sizing* of processing equipment and the *construction of task networks* given a complete process recipe and set of operations. All types of plants have been considered, from the *single process dedicated* plant, to the *multiproduct* plant, to the *multipurpose* plant. However, while published work in batch process scheduling and the design of batch plants has grown exponentially, there has been very little work in the area of *conceptual design* or *process synthesis* for batch processes.

The objective for the development of any chemical process is to generate a profit for the company. Assuming there is a market for the product, in order to meet this objective, the company must be able to produce the product at the required quality specifications at a minimum cost. While this is a relatively simple objective statement, the task of developing a process with minimum cost for a specific product is challenging. We begin by developing a chemical pathway to produce the desired material, then we must select the appropriate set of operations to carry out the synthesis and purification of the product material and finally, we must select the equipment and allocate the process to a production facility. Each of these steps in the development process can significantly impact the cost of the process and the development of an expression that considers each of these components is necessary if one hopes to develop an optimal process. In addition to developing an accurate expression to capture all of the relevant cost components, there are a number of external constraints which contribute to the cost of a process. Legislation and regulations, which govern the chemical processing industry, provide constraints for the batch process development problem. The chemical industry is required to reduce the environmental impact of the process, while maintaining a robust and safe process. For example, the government imposes a strict set of regulations, which chemical manufacturers must follow. Many of these regulations are set at the boundaries of the facility or process and the methods chosen to meet the regulatory constraints are left to those charged with developing the process. At the same time, the competition within the chemical processing industry is increasing, forcing chemical manufacturers to strive to keep costs low. Thus, we see that aside from developing a process which has a minimum cost, we must also develop a process that meets the many external constraints imposed. As part of this thesis we will discuss the relevant cost components for batch processes, as well as, how the external constraints effect the process development.

The task of determining the minimum cost for a batch process is much more complicated than that for a continuous process. This stems from a fundamental difference between the batch process and the continuous process. The development of continuous processes is generally performed from an *equipment-centered* approach.

Since each operation performed in a continuous process can be mapped one-to-one with the piece of equipment that is capable of performing the task, the equipment-centered approach has traditionally been employed. Batch processes, on the other hand, do not employ this one-to-one mapping between operations performed and single pieces of equipment. In many batch processes, the same piece of equipment is employed to perform many operations. Thus, we see that for batch processes it is natural to develop the process using an *operations-centered* approach. In this operations-centered approach, we focus on the selection of the operations that will produce the desired product from our set of raw materials and only after we have selected the operations, do we then go back and consider the allocation of the operations to specific pieces of equipment. This forms an important part of the basis for this proposed work: *the development of batch processing schemes using an operations-centered approach.*

The development of the batch process proceeds through a series of phases before its implementation, i.e. it goes from chemist to process developer, to process engineer. At each of these phases, the process must be reviewed in order to insure a feasible design and, generally, once a design leaves a particular phase, that portion of the design is considered complete. Minor iterations and backtracking do not substantially change this mode of conception, development and engineering of a batch process. This current approach is not conducive in allowing changes in the previous levels of design. Ideally, the system of process development should allow for the rapid iteration between the project developers, engineers and chemists, facilitating rapid communication between the groups and rapid revisions of design alternatives. The rapid iterative process proposed is equivalent to placing feedback loops into the development process and is one of the main factors driving the scope of the work in this thesis.

1.1 Industrial Relevance for Improved Process Development

Of the areas in the chemical processing industry that employ batch chemical processing, the two that depend on it almost exclusively are the pharmaceutical and the specialty chemical industries. These industries typically produce small quantities of very high value-added chemicals, which require very strict quality control. For example, pharmaceuticals are produced in amounts ranging from a few grams to thousands of kilograms. These amounts are relatively low when compared to bulk chemical production and does not justify the construction and operation of a continuous production facility. In order to make full use of the available equipment, batch chemical processing facilities generally produce a wide variety of products, requiring frequent reconfiguring of the batch plants. Batch processing also allows for strict process and quality control. In the course of our interactions with several pharmaceutical companies, they have indicated to us that there is a strong desire to produce all chemical intermediates and products in a solid form. The production of solids allows for tight control of product quality since crystalline material can be produced with extremely high purity. It also makes the process more robust. Through the construction of processes that contains a number of solid intermediates, purification of the solid form can be done with a high degree of accuracy and prevents disturbances from propagating through the process.

One of the main driving forces for improving process development is to decrease the time-to-market for the product under development. This is especially true within the pharmaceutical industry. At a cost of approximately $1,000,000 per day of development [5], the reason becomes obvious. This figure indicates that even a small decrease in the time it takes to bring a product to market will result in significant savings for the firm. In addition to savings generated through reduced time-to-market, a greater economic potential for pharmaceutical companies lies in drug sales. A number of blockbuster drugs generate sales on the order of one billion dollars per year. This averages to approximately three million dollars per day. Since the patent for a pharmaceutical compound expires after seventeen years and it takes anywhere from three to twelve years to get a drug to market, it is clear that reducing time-to-market has the potential to generate hundreds of millions of dollars in additional revenues for the company. It is our expectation that improving process development for batch processes will significantly reduce the time-to-market for these products and will produce benefits for the users of our method.

In addition to reducing time-to-market, in today's regulatory environment, it is also important to develop environmentally friendly processes. Research has indicated that batch processes can produce anywhere from 10 to 10000 kg of waste per kilogram of product [6]. Aggregating these figures across the entire industry, it is clear that the pharmaceutical industry would benefit greatly from developing processes that produce less waste. It has also become clear that end-of-pipe treatment is not the best way to deal with the waste problem. The best way to eliminate waste is to design the process to not produce the waste in the first place. The elimination of waste through process modification is a fundamental principle within the Environmental Protection Agency (EPA) guidelines for waste management and including the minimization of waste production in the development process will allow us to adhere to this principle from the onset of process development.

In order to determine how to develop better processes and, in turn, reduce the time-to-market for new drugs, we have looked into the development process, specifically the interaction of the chemists and the process development engineers. From our experience, we have found the drug development process to move along the following simplified path. Once a compound has been identified as having some value, it becomes the responsibility of the synthesis chemist to develop a chemical synthesis route for the production of the desired compound. The synthesis chemist is generally free to develop the synthesis route using whatever process and materials are available. We have found that the synthesis is typically developed in an ad hoc manner. The chemist usually selects his favorite solvent and other chemicals or the chemicals that he happens to have handy in the laboratory. During the synthesis process, very little thought is given to the potential difficulties these upstream decisions can cause in the downstream processing. After the chemist has a working synthesis, it is passed on to the process development engineers. This is the crucial stage of the development since many of the decisions made by the chemist are locked into place due to the push to get the product into clinical trials. Once the drug has entered the human trials, especially Phase III Clinical Trials, the process is essentially frozen, since changing the process would require additional filings with the Food and Drug Administration (FDA), which is a very costly and time consuming process. Thus, industry would benefit greatly from a method for screening designs as soon as possible after the decisions are made, so that any modifications can be made before the Investigational New Drug application (IND) is submitted.

The process development process is similar for specialty chemical manufacturers, although for specialty chemical manufacturers, in addition to requiring rapid development, it is extremely important to keep manufacturing costs low since they do not operate with the same profit margins as do the pharmaceutical manufacturers. Also, unless the specialty chemical is a intermediate to a food or pharmaceutical, the specialty chemical manufacturers need not comply with the FDA regulations, which significantly increase the cost of process development for the pharmaceutical manufacturers. In order to keep the process development cost low it is important for the specialty chemical manufacturers to be able to screen a number of alternative processes quickly.

1.2 Thesis Objectives

As we have discussed in the previous section, there are many benefits to be gained through the systematization of batch process development. It is also clear that there is a need within the batch chemical manufacturing industry for this systematization. Thus, the primary objective of this thesis is *to develop a systematic method for use in the synthesis of batch processing schemes*. In this thesis, we have focused primarily on the development of single product processes. Even though many batch processes are performed in multiproduct and multipurpose facilities, we have chosen to begin with the development of single product processes because the single product process is the simplest of the three processes in terms of development. While some researchers view the multiproduct and multipurpose facilities as simply a scheduling problem, we must recognize that to achieve an optimal allocation of multiple processes to a given set of equipment, requires process modification in order to select operations for each process, which are compatible with the equipment set.

Of the phases in process development, we have chosen to focus our attention on the process synthesis phase. We believe that this has been an under-investigated area in the field of batch processing and some of the most substantial improvements to the processes developed, can be made through a careful understanding of the process and the decisions made at the process synthesis level. Most of the previous efforts in batch process development have been in the area of plant design, unfortunately these efforts only have minimal impact on the quality of the processes since, by the time we enter the plant design phase of the development process, most of the key process design decisions have been made and we have very little freedom to effect the process. Based on the phases at which key decisions during the process development, it is clear that the most dramatic impact on the development process will be achieved through the decisions made during the chemical route synthesis phase and the process synthesis phase.

Through our interaction with synthesis chemists and process developers we have found that the process development knowledge is divided among the two groups. The synthesis chemist possesses the knowledge of the detailed chemical process and knowledge of the reasoning behind the selection of the process materials and operations. The process developer possesses knowledge of the translation of the lab scale process into pilot and manufacturing scale processes and knowledge of manufacturing considerations for chemical processing, such as, health and safety issues concerning the process, the types of industrial operations which are available and their ease of implementation. The key to significantly impacting the development process is the systematization of the process development process.

The systematization of the development process will, perhaps most importantly, enhance the communication and information flow between the synthesis chemist and the process developer. By systematizing the development process, especially the process synthesis phase of development, we facilitate the capture and understanding and reasoning behind the decisions made during the development process. This enhanced understanding will only come from the development of strong interactions and feedback between the two groups that have the knowledge necessary to develop the design. The interaction and feedback will allow the chemist to become aware of the impact his decisions have on the process scale-up and the downstream processing. For the process developer, interacting with the chemist at the early stages of chemical route synthesis and throughout the decision making process has the potential to eliminate troublesome scale-up and downstream processing issues through process modification. For example, during the reaction synthesis the chemist may choose a particular solvent because he has it available in the lab. Upon analysis by the process developer, the developer finds that the solvent is a health hazard when used in large scale processing. By providing the process developer with access to the chemical synthesis work while the chemist is still working on the synthesis experiments, the developer can propose to the chemist the selection of a different solvent. Incorporating this feedback loop early on in the process, we avoid the need of stopping development and going back to perform additional experiments, reducing the overall development time. As a result of the early interaction with the chemist, the process developer will be able to develop and construct a larger number of promising alternative processes in the same amount of time. The evaluation of larger numbers of alternatives will undoubtedly result in better processes being developed in terms of reduced cost, as well as, enhanced health and safety with lower environmental impact.

Additionally, the systematic development of processes will enhance the knowledge retained of the company through decision tracking. Currently, much of the knowledge gained from the development process is lost and must be recreated each time a new process is developed. This loss of information is due to the lack of the ability to capture and represent, in a retrievable format, the reasoning behind the decisions made. The current system of process development relies on human assets to accumulate the knowledge of process development and if the employee should retire or leave the firm, a large portion of the decision making reasoning for specific processes is lost. The implementation of a systematic development methodology is an important tool in capturing and maintaining the decision making knowledge. It allows all future employees to go back and reapply the method for process development so that they can follow how the process was developed and understand the reasoning behind the decisions that were made during the original process development. Thus, the knowledge becomes an asset that is not lost when employees leave and is added as a permanent asset of the company.

At the core of systematizing the process of process development is the systematization of process synthesis. In order to accomplish this task it was necessary to rethink the traditional approach to process synthesis and move from the traditional process development practice of using an *equipment-oriented* approach, to an *operations-centered* approach. We have expressed the shift in approach through the development of an *operations-based language*. This shift provides two main benefits. First, the development of an operations-based language facilitates communication between the chemist and the engineer, since the natural language of the chemist is operations-based. Second, the use of an equipment-based approach imposes artificial constraints on the process synthesis by forcing the process synthesis to focus only on tasks which specific pieces of equipment are able to perform. The operations-based approach frees us to select from all feasible operations in order to synthesize the process and only after we have selected the operations, do we worry about the selection of equipment to perform the selected operations.

Once we understand the implications of process development using an operations-centered approach, we can go ahead and develop a systematic approach for the process synthesis phase of process development. Through the analysis of previous work in the area of process synthesis, for both continuous and batch processes, we see that the previous attempts contain a number of attributes, which can be exploited in the development of a systematic approach to batch process synthesis. One concept we have found particularly useful in this regard is *Means-Ends Analysis.* Means-Ends Analysis can be summarized as follows. Given a starting state and a goal state, identify the differences between the two states and select the task or tasks which remove the differences. In addition to the means-ends analysis concept, we have employed the use of a concept known as *nonmonotonic planning.* Nonmonotonic planning is the task of constructing a process plan in a nonlinear manner. In this thesis we have combined these concepts, as well as features from a number of other methodologies, to formulate the *Means-Ends Analysis with Nonmonotonic Planning* (MEA-NMP) process synthesis methodology.

The ultimate goal of any process development methodology is to develop an *optimal* process. While there is still some debate whether one can identify a *globally optimal* process for every chemical product of interest, not to mention define a method for synthesizing an optimal process, the MEA-NMP process synthesis methodology can be used to develop reduced process superstructures which, if it is possible to identify an optimal process, will greatly reduce the number of alternatives which must be examined through standard optimization routines.

1.3 Thesis Outline

We begin the challenge of developing a systematic method for the synthesis of batch processing schemes by defining batch process development. Batch process development covers the development of the process chemistry (reaction route formulation), the synthesis of a manufacturing scale processing plan (process synthesis) and the design of the production facility (plant design). The number of design decisions that need to be made during process development is extremely large and the development process is quite complex. In Chapter 2 we discuss in detail the phases that compose the batch development process and show why the process is divided into these phases. We also discuss the advantages and drawbacks of pursuing this process in phases rather than as a single task. Finally, we show why it is important to systematize the various phases of the development of batch processes as well as discuss how a systematic methodology, such as the *Means-Ends Analysis with Nonmonotonic Planning* methodology, can play a key role in this process.

Once we have an understanding of the batch development process, the next step is to provide a formal mathematical description of the batch process synthesis problem which will constitute the basis for the development of the systematic solution procedure discussed in Chapter 4. In Chapter 3, we begin with the formal representation of the batch process structure through the use of the *State Task Network.* Following the description of the State Task Network, we proceed to formalize the representation of the States and of the Tasks, as well as, the relation between them through a formalism known as *finite automata.* Once we have formalized, mathematically, all the required components, we show that the batch process synthesis problem is actually a *combinatorial optimization* problem and common solution methods for such optimization problems could be used for its solution. Finally, we discuss how *logic* allows us to create a two-tiered optimization problem in which the logic can be used to reduce the complexity of the optimization problem through the systematic reduction of the solution search space.

After presenting the mathematical formalism of the batch process synthesis problem, we describe the synthesis methodology we have developed in order to systematize batch process synthesis. The methodology that we have developed is called the *Means-Ends Analysis with Nonmonotonic Planning* methodology for process synthesis. The MEA-NMP methodology was derived through the adaptation and combination of methodologies originally proposed for the design of continuous processes. As the name indicates, we have adopted aspects from the Means-Ends Analysis approach to continuous process synthesis and the Nonmonotonic Planning strategy originally proposed within Artificial Intelligence and later used to develop continuous process operating procedures. Additionally, we have integrated the hierarchical approach to process synthesis for the synthesis of continuous processes into our methodology. By incorporating the best features of each of these concepts, we have been able to develop a systematic methodology for the synthesis of batch processes.

In Chapter 4, we provide a detailed description of how Means-Ends Analysis and Nonmonotonic Planning have been incorporated into a process synthesis methodology. We begin by providing a description of the process synthesis problem and discuss the Means-Ends Analysis concept. Following, we discuss the relevant aspects of planning with respect to process synthesis. As part of the discussion, we present the modeling of the operations and the synthesis process. Finally, we bring all the concepts together to form the Means-Ends Analysis with Nonmonotonic Planning methodology and show how the MEA-NMP methodology can be applied to perform batch process synthesis by means of an example. Chapter 5 contains a case-study developed for a pharmaceutical process synthesized using the MEA-NMP methodology.

Following the presentation of the MEA-NMP methodology, in Chapter 6, we extend the MEA-NMP approach to process synthesis in order to create superstructures, which facilitate the optimization of batch processes. We begin by showing the simultaneous creation of the base-case design and the process superstructure of operations, using the MEA-NMP methodology. The superstructures we develop consist of a reduced set of feasible alternatives to the base-case. It is this superstructure to which combinatorial optimization routines can be applied in order to identify the optimal design. We then present the potential problems that are inherent with attempting to find an optimal design, such as large problem size and complexity and insufficient data, along with the problems that are faced in using the MEA-NMP methodology in conjunction with optimization routines. Along with the presentation of the problems we discuss their implications for process synthesis and discuss the trade-offs that are encountered when attempting to determine an optimal design. Once we understand the problems that are encountered during superstructure optimization, we discuss how the optimization problem is formulated using the MEA-NMP methodology. In the discussion of the formulation of the optimization problem we provide a detailed description of the components of the objective function including raw material costs, equipment costs, utilities costs, overhead and labor costs and waste treatment costs, as well as the constraints on the objective function and the models used in the formulation of the optimization problem. At this point we also discuss the connection between MEA-NMP and logic-based approaches to superstructure optimization. Finally, we review a number of superstructure reduction methods and give an example of alternative processes that can be developed using the process superstructure, along with a discussion of the impact alternatives can have on the superstructure developed using the MEA-NMP methodology.

Once we have completed our discussion of the MEA-NMP methodology, it will be apparent that the MEA-NMP methodology requires significant information support in order to be effective. In order to demonstrate the methodology and the data requirements, we have implemented a computer-aided prototype of the process synthesis tool, called the *Process_Synthesizer,* as a layered application within a complete process development and evaluation tool called *BatchDesign-Kit* (BDK). We have chosen to incorporate the *Process_Synthesizer* within BDK in order to make use of the many features available. We discuss the implementation of the *Process_Synthesizer,* as well as all the features of BDK in Chapter 7.

In Chapter 8, we summarize the work presented in this thesis and present the major conclusions and contributions of this work with respect to the field of batch process development. We also introduce a number of additional problems for future research in areas complementary to that of batch process development.

1.4 References:

1. Rippin, D.W.T., *Review Paper: Simulation of Single- and Multiproduct Batch Chemical Plants for Optimal Design and Operation.* Computers & Chemical Engineering, 1983. 7(3): p. 137-156.
2. Reklaitis, G.V. *Progress and Issues in Computer-Aided Batch Process Design.* in *Foundations of Computer-Aided Process Design.* 1989. Snowmass, Colorado.
3. Rippin, D.W.T., *Batch Process Systems Engineering: A Retrospective and Prospective Review.* Computers and Chemical Engineering, 1993. 17(Suppl): p. S1-S536.
4. Allgor, R.J., *Modeling and Computational Issues in the Development of Batch Processes,* in *Department of Chemical Engineering.* 1997, Massachusetts Institute of Technology: Cambridge. p. 401.
5. Rubin, R.H., *et al., 15.136J, Principles and Practice of Drug Development,* 1997: Massachusetts Institute of Technology.
6. Linninger, A.A., *Waste Production by Batch Processes,* 1996.

Chapter 2

Batch Process Development

What is batch process development? Batch process development is the planning and the design of a batch process in order to produce a valuable material or materials. The goal of process development is ultimately to come up with an optimal process for the production of a given material. Whether or not we can achieve this goal is still the matter of some debate but, we do know for certain that the development of batch processes is performed frequently within industry and currently there is no clear systematic method for the development process.

The batch process development process covers the development of the process chemistry (reaction route formulation), the synthesis of a manufacturing scale processing plan (process synthesis) and the design of the production facility (plant design). The number of design decisions that need to be made during process development is extremely large and the development process is quite complex. In this chapter we will discuss in detail the phases that compose the batch development process and show why the process is divided into these phases. We will also discuss the advantages and drawbacks of pursuing this process in phases rather than as a single task. Finally, we will show why it is important to systematize the various phases of the development of batch processes as well as discuss how a systematic methodology, such as the Means-Ends Analysis Nonmonotonic Planning methodology, can play a key role in this process.

2.1 Phases of Batch Process Development

The complexity of the batch process development problem stems from the many phases associated with the process. The different phases are generally performed by different people, among different groups within a company and many times, the groups are physically located in different areas. The phases of batch process development consist of:

1) Chemical Reaction Route Formulation - The development of a reaction route for the production of the desired compound.
2) Process Synthesis - The development of the processing steps for the production of the desired compound.
   a) Development of the Operating Procedure - Choosing the production tasks that correspond to the lab scale operations along with determining the ordering of these operations and preliminary selection of the operating conditions.
   b) Regulatory Compliance - Evaluation of the materials and operations with respect to all applicable regulations including safety, health and environmental impact.
   c) Process Optimization - The determination of the optimal process, including tasks selection, specification of operating conditions and task ordering.
3) Plant Design - The development of the processing facility for the production of the desired compound.
   a) Equipment Allocation and Scheduling - Assigning the operations to physical pieces of equipment. This includes determining the required sizing of equipment, task merging, parallel processing and the use of intermediate storage. The result is an operating policy for the plant.
   b) Plant Optimization - The determination of the optimal plant structure including operating conditions, task ordering, equipment allocation and scheduling.
4) Total Process Optimization - The determination of the optimal process and plant structure simultaneously.

Before we describe each of the phases and their components in detail, let us first go through a brief outline of the traditional process by which a batch process is developed. We will take our example from the pharmaceutical industry, which produces its products almost exclusively in batch processes.

2.1.1 Traditional Batch Process Development

We begin our example by assuming that we have a compound which has been identified as a desirable drug candidate. Within the pharmaceutical industry, the batch development process takes place during the evaluation of a compound for its safety, efficacy, etc. This is done for two reasons, first is that the pilot plant production runs will be used for the clinical trials of the drug so we must have the ability to product the drug in sufficient quantities for the clinical trials. Second, is that once the clinical trials are complete, we want to go into production as soon as possible. For pharmaceuticals, the Food and Drug Administration (FDA) evaluates the process by which the drug will be produced and the evaluation process takes a considerable amount of time. Also, once the application has been submitted to the FDA for approval, it is very costly to make changes to the process thus, we must have a process ready to be evaluated by the FDA at the time of completion of the clinical trials.

2.1.1.1 Chemical Route Development

The process development begins with the synthesis chemists developing a reaction route for the process. Currently, within the pharmaceutical industry, the synthesis chemists receive very little external input about the development of the reaction route. They are treated with a hands-off attitude by the corporation, so as not to inhibit their creativity. Especially with pharmaceuticals, the value of the product and the time pressure to get the product to market is so high, management is willing to deal with the consequences of the decisions made by the chemists at later stages of development. A phrase frequently heard is "Just give us a reaction route." Thus, the chemist developing a process is free to use an ad hoc approach to synthesize the reaction route, choosing pathways and materials without regard to the downstream implications of these decisions, such as waste management problems and difficult processing steps. This hands-off philosophy is most likely due to the lack of understanding by management that a majority of the decisions made by the chemist are frozen into place for the rest of the development process and all the decisions made by the chemist influence the remaining aspects of process development.

While this attitude is slowly changing, it is still a problem within industry today. Thus, we see that it is important to influence the chemist's decision making process through interaction with the rest of the development team. We will discuss this point in greater detail in Chapter 7. For now, let us continue by assuming that the chemist has generated a reaction route for the production of the desired compound. The traditional approach, at this point, is for the chemist to hand the process over to the process engineers in the form of a process recipe. Illustration 2.1 contains an excerpt from an actual process recipe provided to us in a example pharmaceutical process by a large pharmaceutical company. Notice that this recipe explains step-by-step the method the chemist used to produce the product. Also, notice that the chemist has included important notes about the process. What is not shown in the excerpt is that the chemist has provided us with the main reaction and one of the side reactions that occur in the process along with a list of materials and the amounts required. The details of the route development and the process recipe are discussed later in this chapter.

> *Illustration 2.1*
>
> Case Study III - Chemist's Recipe
>
> Under a nitrogen atmosphere a 3.0M solution of methylmagnesium bromide in ethyl ether (474ml, 1.42mol) was added to tetrahydrofuran (500 ml, sieve-dried to KF=7 mcg/ml with 4Å molecular sieves) in a 3L, 3-necked flask fitted with a mechanical stirrer, thermometer and a 1000ml addition funnel.
>
> > Note:
> >
> > The addition is exothermic; cooling is used to keep the solution temperature below 25C. The reagent cystallizes out of solution during the addition.
>
> The trienone I (200g, 0.97mol) in tetrahydrofuran (400ml) was added from the addition funnel over 1.5hr.
>
> > Note:
> >
> > The temperature of the solution is kept at 15-20C with a cooling bath; the exotherm is easily controlled. The tan suspension of the Grignard becomes lavender in color.
>
> The reaction mixture was aged at room temperature for 1hr with a gentle nitrogen sweep...

Before we continue on with the process development we must ask a key question. What is the potential market demand for this compound? At this point, we need not answer this question in detail but more in a qualitative manner. The market demand forecast will give us a preliminary indication of the level of process development that needs to be preformed. For example, if the demand will be only for a few hundred grams of active ingredient, such as for some therapeutic proteins, we may be able to avoid the planning of a larger scale production facility and meet the product demand with a large lab scale production. On the other hand, if the amount of product to be produced is large, then we will need to begin to evaluate whether we should produce the product in large scale batch production operation, either in a single product plant, if the demand is high enough or should we look to fit the process into an existing facility that is not being used to capacity or should we use a continuous process instead. For pharmaceuticals, issues of quality and consistency of the product generally limit production to the use of batch processes. Even large quantity over the counter drugs like aspirin are produced in a batch mode but advances in the area of small scale continuous processes may make this an attractive option for the pharmaceutical industry so we should maintain this as an option. For now however let us assume that the market demand forecast suggests that the scale is large enough to warrant the development of a single product batch production facility.

The passing of the process on to the process engineers marks the end of phase one and the beginning of phase two of the development process. Now it is the job of the process engineers to synthesize a process that is capable of producing the quantities of product required by the marketing forecast. This is the process synthesis phase of development.

2.1.1.2 Process Synthesis

Process synthesis is defined as the development of the processing steps for the production of the desired material. This consists of the selection and relative ordering of the operations with respect to each other, as well as, the determination of the operating conditions for the tasks which will be performed. The traditional approach used by many pharmaceutical firms is to attempt to directly scale-up the laboratory process provided by the chemist, making minimal changes to the process and incorporating industrial scale operations. Frequently, the process synthesis step is used as the pilot plant development step. The trouble with this approach is that it captures all the inefficiencies of the lab scale process such as, excessive solvent usage and course multi-step separations and it treats the batch process in an ad hoc manner of assigning operations to equipment. Illustration 2.2 is an excerpt from Case Study III provided by a large pharmaceutical company. This excerpt corresponds to the chemist's recipe given in the previous section.

From the excerpt in Illustration 2.2 we see that the process synthesis phase is treated as the pilot plant development phase.

> *Illustration 2.2*
>
> Case Study III - Process Description
>
> Tetrahydrofuran (55.5 kg) and methyl magnesium bromide (3M) in ethyl ether (61.3kg) are charged to ST-203 (100 gal., glass-lined) followed by a tetrahydrofuran (2L) line rinse. Trienone I (25.0 kg)/tetrahydrofuran (44.4kg) solution (previously prepared) is charged to ST-203 via pulsafeeder pump over 3.5hr, while maintaining a batch temperature of 15-20C. Tetrahydrofuran (5kg) is charged as a line rinse to ST-203 and the batch is aged at 25±1C for 3hr.

Generally, the goal of the chemist is to develop a route that maximizes yield and product purity. The chemist fails to consider other issues such as, the difficulty of large scale versions of the operations chosen, safety and health issues associated with large scale processing, environmental impact and waste treatment issues, all of which are major factors in developing a sound process. Thus it falls on the shoulders of the process engineer to take these issues into account when synthesizing the process.

Fortunately, the chemical industry is realizing that in order to remain competitive, they need to develop processes more quickly, which are more robust and more efficient, which also meet all of the stringent new regulatory requirements imposed on chemical manufactures. The traditional approach to process development must be reevaluated in terms of a systematic procedure for process synthesis and process development as a whole. It is clear that the process synthesis task can be improved significantly through a systematic approach and one of the main goals of this thesis is to provide a systematic approach to batch process synthesis. Later, in this section on batch process synthesis, we will describe some of the previous attempts at systematizing the process synthesis task and highlight the difficulties faced in these previous attempts. For now, let us assume that the process engineers have performed the process synthesis task and have selected the operations, structured the ordering of the tasks and provided a preliminary set of operating conditions for the process. This ends phase two of the process development and now we can move on to phase three, the plant design phase.

2.1.1.3 Plant Design

Phase three of the batch process development is the plant design phase. The plant design phase consists of the development of the processing facility for the manufacturing of the desired product. As mentioned previously, this facility can be of the form of a pilot plant up to a large scale multipurpose facility. Most pharmaceutical processes consist of the development of a pilot plant before the development of the actual processing facility but the pilot process usually resembles the final plant only on a smaller scale. Rarely are significant changes made to the process after it leaves the pilot plant.

The plant design phase may take one of two tracks, 1) the retrofitting of the process into an existing facility (assuming the building structure and the utilities are in place) and 2) the grass-roots design of a new facility. Depending upon which track is chosen, different design strategies will be followed. When fitting a process into an existing facility, we want to make the best use of the equipment available to avoid the purchase of additional equipment thus, we are constrained by the list of available equipment. While, in the grass roots design, we are constrained by a capital budget for purchasing the necessary equipment. In either case, the plant design team must determine the type of equipment to be used, the equipment sizing, the best use of intermediate storage and parallel processing units, perform the scheduling and make sure the required utilities are available.

A key part of allocating the process to physical equipment is determining which tasks to perform in the various pieces of equipment. As mentioned previously, for batch processes we can generally perform more than one operation in a single piece of equipment. The allocation of more than on operation to a single piece of equipment is known as *task merging*. As we can see, the equipment allocation process is interrelated with the scheduling of the process. Ideally, from an economic point of view, we would like all equipment to be used 100 percent of the time at full capacity. Clearly, this is not possible due to the differing cycle times of the various operations. Thus, we perform scheduling with intermediate storage and parallel processing in order to try to maximize equipment usage and minimize the process cycle time.

During the design process, the engineer must develop the process with safety considerations in mind. The designer must not develop processes with tasks and equipment configurations which are unsafe. All to often, the evaluation of the safety of a design is left until the design phase is complete and the designer must go back and make modifications to the design in order to address the safety issues. Clearly, it would be better to identify safety issues during the design phase so that the design will include the means to address these concerns. This has the potential for both improving the safety of the design that is developed as well as reduce the development time.

The completion of the design phase leaves us with a complete processing plan including all the processing tasks, the equipment in which the tasks will be performed and the schedule of operation for the process. At this point the process can be given to manufacturing and the processing can begin. Illustration 2.3 is an excerpt take from the Plant Design phase of Case Study III. Not shown is the corresponding flowsheet that was provided. We see from the illustration that the final plant design is almost identical to the pilot plant design. Also, notice that this process has been retrofitted into an existing facility through the use of physical equipment names.

Illustration 2.3

Case Study III - Plant Design

Step Grignard

1) Sequential Operations.
2) CHARGE ST-205 with 111kg WATER.
3) CHARGE ST-205 with 20.0kg NAOAC.
4) CHARGE ST-205 with 14.6kg ACETIC-ACID.
5) CHARGE ST-207 with 44.4kg THF.
6) NOTE: THF is sieve-dried.
7) CHARGE ST-207 with 24.25kg TRIENONE contained in TRIENONE-CRUDE.
8) CHARGE ST-203 with 55.5 kg THF.
9) NOTE: THF is sieve-dried.
10) NOTE: check for KF.
11) CHARGE ST-203 with 61.3kg MEMGBR-ETHER-3M.
12) TRANSFER CONTENTS OF ST-207 TO ST-203.

13) AGE ST-203 FOR 1hr.
14) COOL ST-203 TO 12.5C
15) ...

In the above construction of the phases of batch process development, we see that there is a distinction between the three main phases; reaction route formulation, process synthesis and plant design. Due to the complex nature of the batch process development problem, these distinctions were imposed by past process developers in order to simplify the problem. The disadvantage of the development of each phase independently is that decisions made in each phase impact the options of the developers in the future phases. Notice that at the end of the process synthesis phase and the plant design phase we have an optimization step which was not included in the example since optimization is rarely performed within the pharmaceutical industry. Since the distinction between the process synthesis phase and the plant design phase is artificial, and the two are actually quite interrelated, process development in this manner can only lead to suboptimal processes. The only way to determine an optimal process would be to consider all three phases, the chemical reaction route development, process synthesis and plant design, simultaneously in a large optimization problem as is reflected in phase four above.

Obviously, with current capabilities, the simultaneous optimization of all three phases is impossible. What we propose in this thesis is the rapid and complete exchange of information between the process developers in each phase of development. This would allow the incorporation of feedback loops in the process development among the different phases and should result in better processes being developed. For example, if the process developer in the process synthesis phase of development had access to the development chemists work very early on in the process, the process developer could begin preliminary design evaluations and provide feedback to the development chemist with respect to the chemical route formulation. This is of particular importance with respect to difficult processing steps that may not be difficult for the chemist to perform on the lab bench but provide significant difficulties once scaled up to production. Additionally, with the ever changing regulations regarding chemical usage, if the development team could evaluate the chemical synthesis route early on, they could provide feedback to the chemist and possibly have the chemist evaluate additional routes that avoid or limit the usage of offending chemicals while the chemist is still working on the synthesis route for this process. The importance of this point cannot be over emphasized since the cost of stopping the development once the chemist has passed on the process in order to perform additional experiments can be tremendous. We will revisit this issue in Chapter 7. Now that we have provided a general outline of the batch process development process by way of an example, let us discuss each phase in more detail.

2.1.2 Chemical Reaction Route Formulation

Once an interesting compound has been discovered, it becomes the responsibility of the synthesis chemist to determine a reaction route for the production of that compound. In many cases, these reaction routes include the production of a number of intermediate compounds which must be isolated and become the raw materials for future processing steps. Currently, the synthesis chemist is given very few guidelines for developing the route for the chemical production. What we have been told is that the companies do not wish to impede the creativity of the chemists. The driving force behind this logic stems from the extremely high value of many of these pharmaceuticals and specialty chemicals. There is such a push to get the product to market quickly that there is little concern about the potential problems that are created in the early stages of process development, even though most of the process design decisions are fixed by the decisions made by the chemist during route development.

Since this is the case, in order to have an impact on how the processes are developed, we need to effect how the chemist develops the process. The only way to do this is to provide feedback to the chemist while they are still working on the current project. In the current mode of process development the chemist has finished developing the process for a particular compound and has begun working on another project before the engineers realize that the decisions the chemist has made will have a negative impact on the downstream processing. It would be much more effective and efficient to be able to provide feedback to the chemist while they are still working on the project.

Once the chemist has developed the chemical reaction route, the chemist provides the process engineer with what is commonly known as the *process recipe*. The process recipe includes: the chemical reactions including stoichiometry, the amounts of raw materials, solvents and catalysts used, the operations the chemist has performed in order to generate and isolate the product compound, the conditions at which the operations were performed and the process yields. In addition, we are usually provided with any physical property information the chemist has collected. From our experience, the chemist usually collects information regarding only the product compound. This information usually includes boiling point, melting point and perhaps some assay information such as the elution time for HPLC (high pressure liquid chromatography). Frequently, the chemist provides us with any observations made during the development of the reaction chemistry such as the color of the resulting mixture or observed phase changes. Also, the chemist usually notes any relevant safety information such as, add reactant slowly since there is a significant exotherm.

Additionally, we should note, in many cases the reaction information that the chemist provides is incomplete. The chemist focuses on the product formation, so in many cases the by-products are left unidentified. If the by-products produced are not apparent from the reaction information then the process engineer must either make an educated guess at the by-product identity or resort to what we call *atomic garbage collection,* in which any excess atoms are collected and treated as a single compound in order to close the material balances. With the information provided by the chemist in hand, we can go ahead and begin the process synthesis.

2.1.3 Batch Process Synthesis

The next stage of batch process development is the process synthesis stage. Generally, when we speak of process synthesis, we refer to the selection of the processing tasks that will be used to produce the desired product. Unfortunately, there has been very little research in the area of process synthesis for batch processes. In a review of batch processing Rippin [1] reports: *No systematic approach to this problem has been reported.* The problem stems from the assumption made by most researchers that this information is somehow "given." For example, both of the highly cited review papers by Rippin [2, 3] assume that the process recipe is given and fixed, thus in these reviews Rippin does not acknowledge this as a key area for research and process improvement. In another paper Rippin [4] did, however, pose a challenge that indicates the importance of this problem and has inspired a few researchers to formulate some different views of process synthesis based on the two dominant methods used in process synthesis:

> *...it may be instructive to look at each {method} through the eyes of the other: - could Douglas's physically grounded narrative be expressed in Grossmann's MINLP form, or could the MINLP algorithms be explained in narrative style?*

Reklaitis [5] also acknowledges that this is an area that is in strong need of research, especially in the area of batch process synthesis. Fortunately, this need is carrying over into research, with a number of researchers focusing on systematizing the process of batch process synthesis.

Before we continue with the discussion of the operating procedure development, let us first take a look at some of the previous attempts at developing strategies for process synthesis and operating procedure development.

2.1.3.1 Previous Approaches to Process Synthesis

Over the past twenty years there have been various attempts at formulating methods for the generation of operating procedures, most of which have been in the area of continuous processes. Since it is the methods of process synthesis in which we are interested, let us first examine some of the approaches developed for continuous processes and then move on to those for batch processes.

The earliest attempts at the formulation of operating procedures for chemical processes were generally heuristic or experience based. Engineers would use their experience to generate heuristics, which they applied when developing new processes. One of the first attempts at categorizing and systematizing the synthesis of separation system operating procedures was presented by King [6]. In his book on separation processes, King provides a discussion of the selection of separation processes. In this discussion he categorizes various separation operations into several categories which facilitate comparison and selection of operations:

1) Mechanical processes vs. diffusional processes
2) Equilibrium processes vs. rate-governed processes
3) Energy-separating-agent processes vs. mass-separating-agent processes

Combining these categories with a priority and heuristic structure, King presents what can be considered the first separation system synthesis scheme. While King's scheme provides us with a general guideline to the selection of separation operations, it does not provide a systematic method for the selection of individual separation operations.

At approximately the same time that King published his book, Siirola [7] and Rudd [8] developed a computer system known as AIDES (Adaptive Initial DEsign Synthesizer). The AIDES system was the first system to incorporate a systematic approach in the synthesis of chemical processes. The AIDES system used the concept of *means-ends analysis* to construct designs in a linear manner by identifying differences between the current state and the desired goal state and then identifying tasks that could resolve those differences. It was the first system to attempt to assist the user in making design decisions. Siirola and Rudd were also the first to recognize that chemical process development is more then formalizing the heuristic approaches to design, as Siirola wrote in his Ph.D. thesis [7]:

> *The {AIDES} method is not so much an attempt to formalize observed design practice as it is a proposal of what might be a more logical approach. For example, the emphasis is not on the assembly of equipment, but rather on the identification of tasks which must be performed and the physical principles which might be exploited to perform them.*

AIDES used a combination of heuristics and analysis to develop the operating procedures for continuous processes. The procedures developed by AIDES consisted of phase separation, component separation, heating and cooling operations. The system did not specify specific operations but it did develop flowsheets that corresponded well with some simple examples. For a detailed description of the AIDES system see Siirola [7].

Following in the footsteps of the AIDES system was the BALTAZAR system developed by Mahalec and Motard [9, 10]. BALTAZAR built upon the concepts used in the AIDES system, while expanding its applicability by removing the restriction of preliminary species allocation as required by the AIDES system. A significant contribution of the BALTAZAR system was the concept of *evolutionary search.* The evolutionary search consists of applying a number of rules to the base case design in order to modify and improve the design. It is this evolutionary search which in essence was a foreshadow of the next major advancement in the development of process synthesis methods.

The next major advancement in the area of operating procedure synthesis, came with the development of the *hierarchical approach* to process synthesis by Douglas [11, 12]. The hierarchical approach is a much more formal statement of the evolutionary search proposed by Mahalec and Motard. Douglas's hierarchical approach suggests that the process design should proceed in a hierarchical evolutionary manner, moving from an abstract design to a detailed design. The hierarchical approach is a depth-first design procedure in order to construct a base case design as rapidly as possible while maintaining the alternatives for each design decision. The goal of this methodology is to quickly develop a design while at each level of abstraction, evaluating the economics of the process in order to determine if it is worth continuing with the design. The logic here is that if a design is not economical at a high level of abstraction, adding more detail to the design will only make the design less economical and we should terminate uneconomical designs as early as possible. The maintenance of the alternatives for each design decision allow the developer to go back after the completion of the base case design, develop alternatives and compare alternative designs. The hierarchical approach was implemented by Kirkwood [13, 14] in a system known as PIP (Process Invention Procedure) and was further extended by Han in a system known as ConceptDesigner [15, 16].

Alternatively, during the same time period, a competing school of thought had developed suggesting that process synthesis should be performed through the formulation and solution of mathematical programming problems. Umeda et al [17-19] first proposed formulating the process synthesis problem as a nonlinear programming problem (NLP) using continuous variables. The approach by Umeda et al was a combined evolutionary optimization approach and required an initial flowsheet in order to begin the optimization. Following the work by Umeda et al and after some substantial work in the application of MINLP optimization to the scheduling problem [20], Papoulias and Grossmann [21], and Grossmann [22] realized that the process synthesis problem could not be handled without the inclusion of the integer decisions and formulated the process synthesis problem as a mixed-integer linear programming (MILP) problem. Formulating the process synthesis problem using linear programming was insufficient in order to capture the behavior of the process. Also, solution methods for solving optimization problems were limited.

The next advancement came with the development of algorithms, Duran and Grossmann [23], that could solve certain classes of mixed-integer nonlinear programming (MINLP) problems. While the Generalized Benders Decomposition (GBD) algorithm [24] had been developed more than twenty years earlier and this algorithm was capable of resolving classes of MINLP problems, it was not until Duran and Grossmann [23] extended the GBD to form the Outer Approximation (OA) method that these algorithms sparked substantial interest in the field of chemical process development. Kocis and Grossmann [25, 26] exploited the revived interest in the ability to solve MINLP problems and developed MINLP formulations for process synthesis problems. The work by Kocis and Grossmann led to the development of a process synthesis system based formulating and solving MINLP problems. The system known as PROSYN was developed by Kravanja and Grossmann [27, 28] and has had limited success in solving small scale specially constructed process synthesis problems.

While there have been other attempts at developing systematic methods for process synthesis based on thermodynamics [29-31], there are essentially two schools of thought on process synthesis, a) the hierarchical decomposition approach as promoted by Douglas and b) the mathematical programming approach as promoted by Grossmann and others [32-34]. Most approaches since the development of the basic approaches developed by Douglas and Grossmann have been variations and improvements upon these fundamental approaches. More recently there has been some consolidation of the two schools of thought and there have been a number approaches that have been developed combining elements from the two different approaches. This consolidation of the two approaches began with the application of logic to MINLP problems in process synthesis in order to capture some of the heuristic knowledge in the design process to reduce the size of the optimization problem. Raman and Grossmann [35-37] were among the first to incorporate logic in an attempt to solve MILP and MINLP problems more efficiently. This work was extended further by Viswanathan and Grossmann [38] and Turkay and Grossmann [39]. Daichendt and Grossmann [40] have gone a step further and attempted to integrate a hierarchical decomposition procedure that is similar to that of Douglas's, with an MINLP formulation and solution procedure in order to reduce the complexity of the resulting MINLP problem.

The work of this thesis falls under the combined approach category, although we have also gone a step further and capitalized on basis of the work that has preceded the hierarchical decomposition and the MINLP approaches. As will be shown in the following chapters of this thesis, in addition to exploiting the hierarchical decomposition and the MINLP solution procedures, we make extensive use of the means-ends analysis approach of the AIDES system as well as the concept of *nonlinear planning* which was adopted from the area of operating procedure development in chemical process control [41-44]. The combined approach that we have developed is the next step in the search for a complete design methodology.

The first instance of the development of a systematic approach to batch process synthesis was that of Iribarren [45], which essentially used Douglas's hierarchical design procedure to first construct a continuous process and then convert the continuous process into a batch process. During approximately the same time period Grossmann and coworkers [22, 26-28, 35, 36, 38, 39, 46-48] have attempted to develop systematic approaches to process synthesis based on MINLP formulation. Even though their work was mostly applied to continuous processes, the MINLP formulation has been applied to batch processes as well. More recently, researchers [49, 50] have proposed basing batch process synthesis on *Mixed-Integer Dynamic Optimization* (MIDO) in order to more effectively capture the dynamic aspects of batch processes. Charalambides, Shah and Pantelides [32] have mathematically formalized the State Task Network representation in order to perform process synthesis based on dynamic optimization. All of these approaches have had limited success due to the complexity of the problem.

Daichendt and Grossmann [40] have attempted to meet the challenge posed by Rippin through the development of an integrated approach which exploits aspects from Douglas's hierarchical decomposition method to reduce the complexity of the resulting MINLP optimization problem through the creation of smaller subproblems that are optimized independently and then integrated to form a complete design.

Finally, the work presented in this thesis also focuses on systematizing batch process synthesis. We have not only expoited the methods of Douglas and Grossmann, we have also incorporated aspects of the means-ends analysis approach proposed by Siirola [7], the concept of nonlinear planning brought to chemical engineering by Lakshmanan and Stephanopoulos [41, 42] as well as incorporating our own twist to the problem through the development of the operations-based synthesis [51-53]. The approach we have developed with be explained in detail later in this thesis.

The process synthesis problem that we and others have been trying to solve is stated as follows:

Given:

The chemist's recipe which includes

a) The chemical reactions
b) The laboratory processing steps
c) The operating conditions for those steps
d) The raw materials
e) The product specifications including intermediate product specifications
f) Additional materials used in the process including solvents, catalysts and non-reacting quench materials
g) Physical property information that was collected
h) Notes on operations such as the presence of a large exotherm or the color of a mixture

Determine:

The operating procedure which consists of the optimal set of

a) Processing tasks

b) The relative ordering of those processing tasks

c) The operating conditions for the tasks

Subject to the goal of minimizing production costs including raw material costs, utility costs, waste treatment costs and any costs due to health or safety implications.

We can see from the problem statement that this is a complex problem and we have not yet considered the additional complexity that would be brought in through the equipment allocation that will take place in the next phase of development.

The components of process synthesis that were listed above indicate a complex series of steps that, ideally, should be integrated in order to develop an optimal design. Currently, however, the system is far too complex to formulate as a single problem, thus, finding a truly optimal process with regard to all the elements of process development is, with today's capabilities, impossible. What we suggest is the development of a systematic method for the formulation and reduction of the problem based on the development of a base case design and a method to limit the search space of alternatives to only the feasible alternatives for each design decision. Thus, we can formulate criterion that do not necessarily identify the optimal design but do identify designs that meet the criterion we set for the process. In the remainder of this section we will discuss the individual components of the batch process synthesis.

2.1.3.2 Development of the Operating Procedure

The development of the operating procedure is the core of the process synthesis stage of the batch process development. The development of the operating procedure consists of three elements:

1) Operation Selection - The selection of production scale operations to achieve the same results as described in the process recipe provided by the chemist.
2) Operation Ordering - The determination of the ordering of the operations that have been selected.
3) Operating Condition Selection - The determination of the operating parameters for the operations to be used in the process.

It should be obvious that the three elements of the operating procedure development are not independent. The selection of the operations influences the ordering of the operations since there may be a number of different paths that achieve the desired goal state and based on the results of the previously selected operations, alternate operations may be selected and ordered differently. The operating conditions may influence the selection of the operations by altering the composition of the materials that are produced by any given operation.

We can try to overcome this interdependency by setting a goal prioritization for selecting the operations based on attempts to achieve the desired goals. An example of this is the use of the hierarchy of differences proposed by Siirola [54] in order to prioritize the differences that should be resolved during the process synthesis. This hierarchy will be discussed in detail in Chapter 4. We can also combine the goal prioritization with rules for selecting operation such as, *first choose operations that do not require the use of a mass separating agent.* Another approach for selecting operations is the combined use of rules and physical property assessment. Jaksland and Gani [30, 31] have developed a system based on the use of physical properties for the synthesis of separation systems. The major drawback of methods like this is the lack of physical property data and the inability to predict reliably physical properties, especially for solids but methods like this will become more useful as property estimation methods advance.

From the methods used to identify operations, the ordering of the operations will be imposed on the process. Another method for influencing the resulting process is that we can set the desired outputs from each operation and derive operating conditions that will produce the specified results. If an operation cannot be performed at the operating conditions required to generate the desired result, the operation must be removed from consideration.

Earlier we have described the methods used by others to perform process synthesis. In Chapter 4, we will discuss the approach developed for this thesis called the Means-Ends Analysis Nonmonotonic Planning approach and show how it exploits, through integration, many of the strategies discussed above.

2.1.3.3 Regulatory Compliance

The issue of regulatory compliance is becoming ever more important in the chemical processing industry. The regulation of chemical processes takes places at many levels. In the US there are Federal, State and Local guidelines that govern health, safety and environmental impact issues which chemical manufactures must follow. The traditional approach to regulatory compliance has been to develop the process, design the plant and then retrofit the process with modifications such that the process meets all the applicable regulatory guidelines. For environmental impact issues, this usually implied extensive end-of-pipe treatment for the waste materials that were produced by the process and for safety issues this involved adding layers of safety devices to the process.

More recently, however, there has been a realization by industry that regulatory issues are extending well beyond the scope of simply making design modifications to the plant. There is a need to develop processes with the regulatory issues in mind from the onset of development. Due to the sky rocketing costs of waste treatment, the idea is that if regulatory issues are taken into account early on in process development, we can design processes that are safer and more environmentally friendly, while at the same time making the process more efficient and more cost effective. Currently though, there has been very little work in the area of incorporating these issues early in the development process.

This author could only find one work which could be used to look at health, safety and environmental impact properties during reaction route synthesis. A system called SMART (Solvent Measurement, Assessment and Revamping Tool) developed by Modi et al [55] aides the developer in the selection of suitable solvents to be used in the process based on various properties of the solvents. This will undoubtedly be an area of increasing research as corporations move to limit the use of offending chemicals.

There has been some emphasis on implementing waste minimization concepts into the process synthesis stage. The US Environmental Protection Agency has noted that synthesizing processes that produce less waste is the most desirable waste minimization option. Douglas [56] has extended his hierarchical methodology and shown how it can be effective in evaluating alternative processes in order synthesis cleaner processes through a waste minimization approach. Stephanopoulos and coworkers [51-53, 57-59] have made waste minimization a key concept in the development of the *BatchDesign-Kit* system. Aside from the works by Stephanopoulos et al and Douglas, we have only been able to find a small number of other works that discuss waste minimization at the process synthesis level. Cohen and Allen [60] discuss waste minimization through the development of highly efficient processes. Flower et al [61] focus on setting targets and developing processes that meet those targets. Ahmad and Barton [62, 63] have focused on pollution prevention through the reduction of solvent usage through recovery and recycle by integrating these tasks into the process. Stefanis et al [64, 65] have taken a *Life Cycle Assessment* (LCA) approach to minimizing environmental impact and have formulated a design and scheduling methodology based on LCA. Others [66-68] have taken a very domain specific approach to synthesis, and have focused on designing systems with minimal waste production for the given type of system. As we can see there is a serious need for research into waste minimization at the process synthesis level.

As for process synthesis concerning health and safety issues, we have only been able to identify two works that directly address these issues and reflect the importance of their consideration at the process synthesis level. Stephanopoulos et al [51, 58] discuss the screening of materials based on material properties and regulatory rules during the synthesis level and suggest modifying the process in order to minimize the use of the offending chemicals. Heikkila et al [69] have developed a set of rules that are guided by properties of various compounds in the process to help the developer during process synthesis. They also suggest performing a safety analysis on the various alternative processes by calculating some safety indices for the process and then selecting the safest process based on the value of the resulting indices. Again, there is much work to be done in incorporating health and safety issues within the process synthesis phase of design.

2.1.3.4 Process Optimization

We define process optimization as the determination of the optimal process, including task selection, specification of operating conditions and task ordering. While there has been a large amount of work in the area of process optimization in the area of process synthesis, most of the work has generally been of the form of superstructure optimization through the formulation of an MINLP optimization problem. A *superstructure* is a representation of the synthesis problem that contains possible alternative solutions to the synthesis problem embedded within it. The most general superstructure would contain all possible alternative components for the construction of all possible solutions to the problem embedded within it. It is the superstructure that defines the search space of alternatives for the optimization problem. The trouble with superstructure optimization is the complexity of the problem. The resulting optimization problems are generally too complex to be solved using common MINLP solution strategies. Grossmann [22, 70] has pioneered the work in the area of process optimization using MINLP formulations.

The complexity issues regarding the MINLP optimization lead naturally to extensions to this research in the area of superstructure reduction. Grossmann et al [35-39] and others [46, 71, 72] have attempted to develop systematic methods for reducing the process superstructure before applying an optimization solution routine. Most of this work has been based on the application of propositional logic into process synthesis in order to incorporate heuristics and constraints into the MINLP formulation to reduce the search space of alternatives. More recently there has been an attempt at combining hierarchical decomposition and mathematical programming in order to reduce the complexity of the resulting superstructures, Diachendt and Grossmann [40].

In Chapter 6 we describe the role of process optimization in detail and explore how the MEA-NMP methodology developed in this thesis can be exploited in order to develop a reduced superstructure for optimization.

2.1.4 Plant Design

Most of the work in batch process development has been in the area of plant design. Plant design is defined as the development of the processing facility for the production of the desired compound. Traditionally, the plant design phase of process development follows the process synthesis phase. In most cases the work within the area of plant design assumes that the process recipe is given and fixed.

The single product plant design problem has been adapted from the one posed by Yeh and Reklaitis [73] and is as follows:

Given:

a) The amount of product to be produced
b) The available production time
c) The size/duty factors for each task (capacity requirement for that task per unit of final product)
d) The relationships between the processing time of each task and the batch size
e) The status (stable or unstable) of the intermediates produced in each task
f) A list of available equipment types

Determine:

a) The task allocation to the type of equipment
b) The equipment sizing
c) The location of intermediate storage, if necessary
d) The location of parallel process equipment, if necessary
e) The task allocation to the physical equipment
f) The processing schedule

Such that the cost function for the process is minimized.

The traditional goal of batch plant design has been to minimize processing costs for producing the desired product at given specifications, mainly through the minimization of equipment costs. Most of the early approaches attempted to minimize equipment costs by attempting to optimize equipment size. The first paper to consider this design problem for batch plants was by Ketner [74]. In his paper, Ketner addresses the issue optimal equipment sizing for a single product plant through the formulation of an optimal sizing problem. It was ten years before researchers again picked up this line of work. Loonkar and Robinson [75, 76] solved simple sizing problems and extended the formulation to include multiproduct plants and semi-continuous equipment. In following works researchers have investigated the inclusion of discrete equipment sizes and scheduling into the problem formulation, Sparrow et al. [77], optimal design of mutliproduct plants, Grossmann and Sargent [20] and Knopf et al [78], the optimal design of multipurpose plants, Suhami and Mah [79], the issue of intermediate storage, Karimi and Reklaitis [80], optimal scheduling [81] optimal retrofit Vaselenak et al [82], and optimal design Yeh and Reklaitis [73] and Barrera and Evans [83]. This is a representative list of works in the area of batch plant design and a number of comprehensive reviews have been published by Rippin [1, 3] and Reklaitis [5]. A more recent review has been performed by Allgor [84] as part of his Ph.D. thesis. From this list of work and the reviews we see that the plant design problem has had a considerable increase in attention especially over the past 15 years. Now let us discuss the components of the plant design problem and how they impact the batch process development.

2.1.4.1 Equipment Allocation and Scheduling

The plant design problem is the problem of determining an operating policy for the plant. In order to develop an optimal design we must simultaneously allocate the process to physical pieces of equipment and schedule the production using the equipment. This requires determining the required sizing of the equipment, task merging of multiple tasks into a single pieces of equipment, the use of intermediate storage and the use of parallel processing tasks. The problem is further complicated when we attempt to design *multiproduct* and *mutlipurpose* facilities.

Batch plants can be categorized into three main types

1) The single product plant - a plant that is used exclusively to produce a single product.
2) The multiproduct plant - a plant which produces more than one product in the form of single product campaigns. Every batch of a given product uses the same equipment configuration.
3) The multipurpose plant - a plant in which multiple products are produced and a given product need not follow the same processing path from one batch to the next. Equipment items are shared between processes during the same production campaign.

It should be apparent from the definitions of the various types of plants that the single product plant is the simplest to design since we need only consider the needs for a single product thus reducing the complexity of the problem. We see that the multiproduct and multipurpose plants are considerably more difficult when considering the sizing of equipment, the use of intermediate storage and the scheduling.

The equipment allocation problem consists of assigning the operations that have been selected for producing the desired product to physical pieces of equipment. Based on the amount of material and the tasks to be performed, we must either design specific pieces of equipment for grass root facilities or attempt to fit the process into an existing facility with predefined equipment. This problem is considerably more complex for batch processes than for continuous processes. For continuous processes there is usually a one-to-one mapping of operations to the equipment which can perform the operation so the criterion for selecting the equipment for a continuous process is generally the capital cost and operating costs for competing pieces of equipment. For example, if two different types of continuous filters can be used to perform the same filter operation, we would select the filter that has the lowest combined capital and operating costs. For batch processes on the other hand, we generally perform many operations using the same piece of equipment. This is known as *task merging*. Determining which operations should be performed in the same piece of equipment, based on the capabilities of the equipment and the resulting process economics, results in a complex optimization problem.

In order to develop an optimal plant design we cannot decouple the equipment allocation problem from the scheduling problem. In order to schedule the equipment usage we must determine which tasks are to be performed in the various pieces of equipment and the required size of the equipment. Also, due to the different cycle times for the various tasks, we may find the optimal usage of equipment may require the addition of intermediate storage and parallel processing. It should be apparent form the description of the problem that this problem lends itself to solution through optimization and many of the works cited above provide formulations and solutions to the plant design optimization problem.

2.1.4.2 Plant Optimization

As previously mentioned the plant optimization problem consists of the determination of the optimal batch plant structure including task ordering, equipment allocation, scheduling and the determination of operating conditions. There have been a number of solution methods proposed for the solution of the batch plant optimization problem. The solution methods include heuristic based, mathematical programming and combined approaches. The reviews cited above provide references to many of the works in this area up to 1993. An additional search of the literature has shown that the solution methods and formulations the are being developed still rely on combinations of heuristics and mathematical programming in order to solve the plant design problem [40, 85-89]. Just as in the case with the process synthesis optimization, the heuristics are used to reduce the complexity of alternatives by eliminating regions of the search space and then either MILP or MINLP programming is used to determine the solution.

An alternate approach using screening models to reduce the superstructure of alternative design structures has been proposed by Daichendt and Grossmann [47, 90]. The screening models provide a lower bound on the plant cost through the simultaneous consideration of changes to the process structure, the operation of the tasks and the allocation of equipment. Allgor et al [84, 91, 92] have extended the screening model concept to also incorporate certain process synthesis aspects such as the selection of solvents and reagents from a predefined list of candidates. In addition to its capability of reducing the process superstructure, the method proposed by Allgor et al has been shown to be used to generate design targets and a base-case design during the early stages of process development. Another important contribution of the method of Allgor et al, is that the screening models can be used as a lower bounding *Master problem* (See Chapter 3 for definition of Master problem) in the first rigorous decomposition procedure for solving MIDO problems [49]. The main drawback of the method of Allgor et al is that the Master problem is derived using *physical laws and engineering insight*, thus must be derived on a case-by-case basis [49], which may require substantial effort.

2.1.5 Total Process Optimization

Ideally, we would like to develop an optimal batch manufacturing scheme. The development of an optimal batch manufacturing scheme would require the simultaneous determination of the optimal process and optimal plant structure. As we have previously mentioned, each of these tasks individually leads to a very complex optimization problem and the combined optimization problem that considers all components of the process synthesis and the plant design is a great deal more complex. To date, there have been no attempts at the formulation or solution of the total process optimization as a flat MINLP. Only the various subsystems have be formulated and optimized but the independent optimization of the subsystems does not imply an overall optimal solution.

2.2 A Systematic Approach to Process Development

From the descriptions of the phases of process development given in this chapter, we see that there is a need for a systematic approach to process development. With the current approach we will never be able to achieve an optimal design since considering each phase independently is insufficient for determining the optimal process. Perhaps the most important aspect of systematizing the batch process development will be the increased awareness of the implications of the decisions made early on in the design process and how they influence the designs that are eventually developed.

This thesis focuses on systematizing the process synthesis phase of the batch process development. A systematic approach to process synthesis will not only make the development of plant designs easier by allowing the tracking of the decisions that were made and why they were made but, also will allow the process engineers to review more processes and begin to provide the synthesis chemists with feedback at an earlier stage of development. In Chapter 4 we will discuss the Means-Ends Analysis Nonmonotonic Planning approach to process synthesis and describe in detail how it systematizes the process of process synthesis. Then in Chapter 6 we will discuss how this systematic approach to process synthesis can have an impact on design. Finally in Chapter 7, we will show how MEA-NMP can be integrated with the BatchDesign-Kit system for process development and demonstrate how the combined system can have an influence over all three phases of batch process development.

2.3 References:

1. Rippin, D.W.T., *Design and Operation of Multiproduct and Multipurpose Batch Chemical Plants. - An Analysis of Problem Structure.* Computers & Chemical Engineering, 1983. **7**(4): p. 463-481.
2. Rippin, D.W.T., *Review Paper: Simulation of Single- and Multiproduct Batch Chemical Plants for Optimal Design and Operation.* Computers & Chemical Engineering, 1983. **7**(3): p. 137-156.
3. Rippin, D.W.T., *Batch Process Systems Engineering: A Retrospective and Prospective Review.* Computers and Chemical Engineering, 1993. **17**(Suppl): p. S1-S536.
4. Rippin, D.W.T. *Introduction: Approaches to Chemical Process Synthesis.* in *Foundations of Computer-Aided Process Design.* 1989. Snowmass, Colorado.
5. Reklaitis, G.V. *Progress and Issues in Computer-Aided Batch Process Design.* in *Foundations of Computer-Aided Process Design.* 1989. Snowmass, Colorado.
6. King, C.J., *Separation Processes.* McGraw-Hill Chemical Engineering Series. 1971, New York: McGraw-Hill. 850.
7. Siirola, J.J., *The Computer-Aided Synthesis of Chemical Process Designs*, in *Chemical Engineering.* 1970, University of Wisconsin. p. 128.
8. Siirola, J.J. and D.F. Rudd, *Computer-Aided Synthesis of Chemical Process Designs.* Industrial & Engineering Chemistry Fundamentals, 1971. **10**(3): p. 353-362.
9. Mahalec, V. and R.L. Motard, *Procedures for the Initial Design of Chemical Processing Systems.* Computers and Chemical Engineering, 1977. **1**: p. 57-68.
10. Mahalec, V. and R.L. Motard, *Evolutionary Search for an Optimal Limiting Process Flowsheet.* Computers and Chemical Engineering, 1977. **1**: p. 149-160.
11. Douglas, J.M., *A Hierarchical Decision Procedure for Process Synthesis.* AIChE Journal, 1985. **31**: p. 353.
12. Douglas, J.M., *Conceptual Design of Chemical Processes.* McGraw-Hill Chemical Engineering Series. 1988, New York: McGraw-Hill Publishing Company. 601.
13. Kirkwood, R.L., *PIP - Process Invention Procedure: A Prototype Expert System for Synthesizing Chemical Process Flowsheets*, in *Chemical Engineering.* 1987, University of Massachusetts Amherst: Amherst, MA. p. 188.
14. Kirkwood, R.L., M.H. Locke, and J.M. Doglas, *Prototype Expert System for Synthesizing Chemical Process Flowsheets.* Computers and Chemical Engineering, 1988. **12**(4): p. 329-343.

15. Han, C., *Human-Aided, Computer-Based Design Paradigm: The Automation of Conceptual Process Design*, in *Chemical Engineering*. 1994, Massachusetts Institute of Technology: Cambridge, MA. p. 212.

16. Han, C., J.M. Douglas, and G. Stephanopoulos, *Agent-Based Approach to a Design Support System for the Synthesis of Continuous Chemical Processes.* Computers and Chemical Engineering, 1995. **19**(Suppl): p. S63-S69.

17. Umeda, T., A. Hirai, and A. Ichikawa, *Synthesis of Optimal Processing System by an Integrated Approach.* Chemical Engineering Science, 1972. **27**: p. 795.

18. Umeda, T., A. Shindo, and A. Ichikawa, *Process Synthesis by Task Assignment.* Chemical Engineering Science, 1974. **29**: p. 2033-2040.

19. Umeda, T. and A. Ichikawa, *A Rational Approach to Process Synthesis: An Extensive Use of Task Assignment Concept.* Chemical Engineering Science, 1975. **30**: p. 699-707.

20. Grossmann, I.E. and R.W.H. Sargent, *Optimum Design of multipurpose Chemical Plants.* Ind. Eng. Chem. Process Des. Dev., 1979. **18**(2): p. 343-348.

21. Papoulias, S.A. and I.E. Grossmann, *A Structural Optimization Approach in Process Synthesis. Part III. Total Processing Systems.* Computers and Chemical Engineering, 1983. **7**: p. 723.

22. Grossmann, I.E., *Mixed-Integer Programming Approach for the Synthesis of Integrated Process Flowsheets.* Computers and Chemical Engineering, 1985. **9**: p. 463.

23. Duran, M.A. and I.E. Grossmann, *An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs.* Mathematical Programming, 1986. **36**(3): p. 307-339.

24. Geoffrion, A.M., *Generalized Benders Decomposition.* Journal of Optimization Theory and Applications, 1972. **10**: p. 237-260.

25. Kocis, G.R. and I.E. Grossmann, *Global Optimization of Nonconvex Mixed-Integer Nonlinear Programming (MINLP) Problems in Process Synthesis.* Ind. Eng. Chem. Res., 1988. **27**: p. 1407.

26. Kocis, G.R. and I.E. Grossmann, *A Modeling and Decomposition Strategy for the MINLP Optimization of Process Flowsheets.* Computers and Chemical Engineering, 1989. **13**: p. 797.

27. Kravanja, Z. and I.E. Grossmann, *PROSYN-An MINLP Process Synthesizer.* Computers and Chemical Engineering, 1990. **14**: p. 1363.

28. Kravanja, Z. and I.E. Grossmann, *Recent Developments in PROSYN-A topology MINLP Synthesizer.* Computers and Chemical Engineering, 1994.

29. Balakrishna, S. and L.T. Biegler, *A Unified Approach for the Simultaneous Synthesis of Reaction, Energy, and Separation Systems.* Industrial & Engineering Chemistry Research, 1993. **32**(7): p. 1372-1382.

30. Jaksland, C., *Separation Process Design and Synthesis Based on Thermodynamic Insights*, in *Chemical Engineering*. 1996, Technical University of Denmark: Lyngby, Denmark. p. 196.

31. Jaksland, C. and R. Gani, *An Integrated Approach to Process/Product Design and Synthesis Based on Properties-Process Relationship*. Computers & Chemical Engineering, 1996. **20**(Suppl. A): p. S151-156.

32. Charalambides, M.S., N. Shah, and C.C. Pantelides. *Optimal Batch Process Synthesis*. in *AIChE Annual Meeting*. 1993. St. Louis, Missouri.

33. Biegler, L.T., I.E. Grossman, and A.W. Westerberg, *Systematic Methods of Chemical Process Design*. Prentice Hall International Series in the Physical and Chemical Engineering Sciences. 1997, Upper Saddle River, NJ: Prentice Hall PTR. 796.

34. Floudas, C.A., *Nonlinear and Mixed-Integer Optimization*. Topics in Chemical Engineering, ed. K.E. Gubbins. 1995, New York: Oxford University Press. 462.

35. Raman, R. and I.E. Grossmann, *Relation Between MILP Modelling and Logical Inference for Chemical Process Synthesis*. Computers & Chemical Engineering, 1991. **15**(2): p. 73-84.

36. Raman, R. and I.E. Grossmann, *Symbolic Integration of Logic in Mixed-Integer Linear Programming Techniques for Process Synthesis*. Computers & Chemical Engineering, 1993. **17**(9): p. 909-927.

37. Raman, R. and I.E. Grossmann, *Modelling and Computational Techniques for Logic Based Integer Programming*. Computers & Chemical Engineering, 1994. **18**(7): p. 563-578.

38. Viswanathan, J. and I.E. Grossmann, *Symbolic Logic, Optimization and Process Synthesis*, . 1994, The Engineering Design Research Center, Carnegie Mellon University: Pittsburgh, PA.

39. Turkay, M. and I.E. Grossmann, *Logic-Based MINLP Algorithms for the Optimal Synthesis of Process Networks*. Computers and Chemical Engineering, 1996. **20**(8): p. 959-978.

40. Daichendt, M.M. and I.E. Grossmann, *Integration of hierarchical decomposition and mathematical programming for synthesis of process flowsheets*. Computers and Chemical Engineering, 1997. **22**(1-2): p. 147-175.

41. Lakshmanan, R. and G. Stephanopoulos, *Synthesis of Operating Procedures for Complete Chemical Plants - I. Hierarchical, Structured Modelling for Nonlinear Planning*. Computers & Chemical Engineering, 1988. **12**(9/10): p. 985-1002.

42. Lakshmanan, R. and G. Stephanopoulos, *Synthesis of Operating Procedures for Complete Chemical Plants - II. A Nonlinear Planning Methodology*. Computers & Chemical Engineering, 1988. **12**(9/10): p. 1003-1021.

43. Lakshmanan, R., *Synthesis of Operating Procedures for Complete Chemical Plants*, in *Chemical Engineering*. 1989, Massachusetts Institute of Technology: Cambridge. p. 195.

44. Lakshmanan, R. and G. Stephanopoulos, *Synthesis of Operating Procedures for Complete Chemical Plants - III. Planning in the Presence of Qualitative, Mixing Constraints.* Computers & Chemical Engineering, 1990. **14**(3): p. 301-317.

45. Iribarren, O.A., *Batch Chemical Process Design*, in *Chemical Engineering*. 1985, University of Massachusetts Amherst: Amherst, MA. p. 221.

46. Floudas, C.A. and I.E. Grossmann. *Algorithmic Approaches to Process Synthesis: Logic and Global Optimization.* in *Fourth International Conference on Foundations of Computer-Aided Process Design.* 1994. Snowmass, Colorado: CACHE-AIChE.

47. Daichendt, M.M. and I.E. Grossmann, *Preliminary Screening for the MINLP Synthesis of Process Systems. I. Aggregation and Decomposition Techniques.* Computers and Chemical Engineering, 1994. **18**: p. 663.

48. Grossmann, I.E., *Mixed-Integer Optimization Techniques for Algorithmic Process Synthesis*, in *Process Synthesis*, J.L. Anderson, Editor. 1996, Academic Press: Boston. p. 172-246.

49. Allgor, R.J. and P.I. Barton, *Mixed-Integer Dynamic Optimization I: Problem Formulation.* Computers & Chemical Engineering, 1998.

50. Barton, P.I., *et al.*, *Dynamic Optimization in a Discontinuous World.* Industrial & Engineering Chemistry Research, 1998. **37**(3): p. 966-981.

51. Linninger, A.A., *et al.*, *Synthesis and Assessment of Batch Processes for Pollution Prevention.* AIChE Symposium Series, 1995. **90**(303): p. 46-53.

52. Linninger, A.A., *et al.*, *Generation and Assessment of Batch Processes with Ecological Considerations.* Computers and Chemical Engineering, 1995. **19**(Suppl): p. S7-S13.

53. Linninger, A.A., *et al. BatchDesign-Kit: A Comprehensive Computer-Aided Design Framework with Ecological Considerations for Batch Process Development.* in *AIChE Annual Meeting.* 1996. Chicago, IL.

54. Siirola, J.J., *Strategic Process Synthesis: Advances in the Hierarchical Approach.* Computers & Chemical Engineering, 1996. **19 Supplement**(ESCAPE-6).

55. Modi, A., *et al.*, *Rapid Plant-Wide Screening of Solvent for Batch Processes.* Computers & Chemical Engineering, 1996. **20**(Suppl. A): p. S375-S380.

56. Douglas, J.M., *Process Synthesis for Waste Minimization.* Industrial & Engineering Chemistry Research, 1992. **31**(1): p. 238-243.

57. Linninger, A.A. and G. Stephanopoulos. *Modern Waste Management Using a Technology Assessment Tool.* in *AIChE Summer.* 1995. Boston, MA: AIChE.

58. Linninger, A.A., S.A. Ali, and G. Stephanopoulos, *Knowledge-Based Validation and Waste Management of Batch Pharmaceutical Process Designs.* Computers and Chemical Engineering, 1996. **20**(Suppl): p. S1431-S1436.

59. Linninger, A.A., *et al.*, *Pollution Prevention for Production Systems of Energetic Materials.* Waste Management, 1997. **17**(2-3): p. 165-173.

60. Cohen, Y. and D. Allen, *An Integrated Approach to Process Waste Minimization Research.* Journal of Hazardous Materials, 1992. **29**(2): p. 237-253.

61. Flower, J.R., S.C. Bikos, and S.W. Johnson, *A Modular Approach to Synthesis of Cleaner Processes.* Computers & Chemical Engineering, 1995. **19**(Suppl): p. S45-S50.

62. Ahmad, B.S. and P.I. Barton, *Solvent Recovery Targeting for Pollution Prevention in Pharmaceutical and Specialty Chemical Manufacturing.* AIChE Symposium Series, 1994. **90**(303).

63. Ahmad, B.S. and P.I. Barton. *Synthesis of Batch Processes with Integrated Solvent Recovery and Recycling.* in *AIChE Annual Meeting.* 1995. Miami.

64. Stefanis, S.K., A.G. Livingston, and E.N. Pistikopoulos, *Minimizing the Environmental Impact of Process Plants: A Process Systems Methodology.* Computers & Chemical Engineering, 1995. **19**(Suppl): p. S39-S44.

65. Stefanis, S.K., A.G. Livingston, and E.N. Pistikopoulos, *Environmental Impact Considerations in the Optimal Design and Scheduling of Batch Processes.* Computers & Chemical Engineering, 1997. **21**(10): p. 1073-1094.

66. El-Halwagi, M.M., *Synthesism of Reverse-Osmosis Networks for Waste Reduction.* AIChE Journal, 1992. **38**(8): p. 1185-1198.

67. Crabtree, E.W., M.M. El-Halwagi, and R.F. Dunn, *Synthesis of Hybrid Gas Permeation Membrane/Condensation Systems for Pollution Prevention.* Journal of the Air & Waste Management Association, 1998. **48**(7): p. 616-626.

68. Srinivas, B.K. and M.M. El-Halwagi, *Optimal Design of Pervaporation Systems for Waste Reduction.* Computers & Chemical Engineering, 1993. **17**(10): p. 957-970.

69. Heikkila, A.M., M. Hurme, and M. Jarvelainen, *Safety Considerations in Process Synthesis.* Computers & Chemical Engineering, 1996. **20**(Suppl. A): p. S115-S120.

70. Grossmann, I.E. *MINLP Optimization Strategies and Algorithms for Process Synthesis.* in *Foundations of Computer-Aided Process Design.* 1989. Snowmass, Colorado.

71. Friedler, F., *et al.*, *Graph-Theoretic Approach to Process Synthesis: Axioms and Theorems.* Chemical Engineering Science, 1992. **47**(8): p. 1973-1988.

72. Friedler, F., *et al.*, *Graph-Theoretic Approach to Process Synthesis: Polynomial Algorithm for Maximal Structure Generation.* Computers and Chemical Engineering, 1993. **17**(9): p. 929-942.

73. Yeh, N.C. and G.V. Reklaitis, *Synthesis and Sizing of Batch/Semicontinuous Processes: Single Product Plants.* Computers & Chemical Engineering, 1987. **11**(6): p. 639-654.

74. Ketner, S.E., *Minimize Batch Equipment Cost.* Chemical Engineering, 1960. **22**(6/7): p. 121-124.

75. Loonkar, Y.R. and J.D. Robinson, *Minimization of Captial Investment for Batch Processes.* Ind. End. Chem. Process Des. Dev., 1970. **9**(4): p. 625-629.

76. Robinson, J.D. and Y.R. Loonkar, *Minimization of Captial Investment for Multiproduct Batch-plants.* Process Technology International, 1972. **17**(11): p. 861-863.

77. Sparrow, R.E., G.J. Forder, and D.W.T. Rippin, *The Choice of Equipment Sizes for Multiproduct Batch Plants. Heuristics vs. Branch and Bound.* Ind. Eng. Chem. Process Des. Dev., 1975. **14**(3): p. 197-203.

78. Knopf, F.C., M.R. Okos, and G.V. Reklaitis, *Optimal Design of Batch/Semicontinuous Processes.* Ind. Eng. Chem. Process Des. Dev., 1982. **21**(1): p. 79-86.

79. Suhami, I. and R.S.H. Mah, *Optimal Design of Multipurpose Batch Plants.* Ind. Eng. Chem. Process Des. Dev., 1982. **21**(1): p. 94-100.

80. Karimi, I.A. and G.V. Reklaitis, *Intermediate Storage in Noncontinuous Processes Involving Stages of Parallel Units.* AIChE Journal, 1985. **31**(1): p. 44-52.

81. Takamatsu, T., I. Hashimoto, and S. Hasebe, *Optimal Scheduling and Minimum Storage Tank Capacities in a Process System with Parallel Batch Units.* Computers & Chemical Engineering, 1979. **3**: p. 185-195.

82. Vaslenak, J.A., I.E. Grossmann, and A.W. Westerberg, *Optimal Retrofit of Multiproduct Batch Plants.* Ind. Eng. Chem. Res., 1987. **26**: p. 718-726.

83. Barrera, M.D. and L.B. Evans, *Optimal Design and Operation of Batch Processes.* Chem. Eng. Comm., 1989. **82**: p. 45-66.

84. Allgor, R.J., *Modeling and Computational Issues in the Development of Batch Processes*, in *Department of Chemical Engineering*. 1997, Massachusetts Institute of Technology: Cambridge. p. 401.

85. Voudouris, V.T. and I.E. Grossmann, *Mixed-Integer Linear-Programming Reformulations for Batch Process Design with Discrete Equipment Sizes.* Computers & Chemical Engineering, 1992. **31**(5): p. 1315-1325.

86. Voudouris, V.T. and I.E. Grossmann, *Optimal Synthesis of Multiproduct Batch Plants with Cyclic Scheduling and Inventory Considerations.* Industrial & Engineering Chemistry Research, 1993. **32**(9): p. 1962-1980.

87. Voudouris, V.T. and I.E. Grossmann, *MILP Model for Scheduling and Design of a Special Class of Multipurpose Batch Plants.* Computers & Chemical Engineering, 1996. **20**(11): p. 1335-1360.

88. Subrahmanyam, S., *et al.*, *Application of Distributed Computing to Batch Plant Design and Scheduling.* AIChE Journal, 1996. **42**(6): p. 1648-1661.

89. Lee, H.K. and I.B. Lee, *A Synthesis of Multiproduct Batch Plants Considering Both In-Phase and Out-of-Phase Modes.* Computers & Chemical Engineering, 1996. **20**(Suppl. A): p. S195-S200.

90. Daichendt, M.M. and I.E. Grossmann, *Preliminary Screening for the MINLP Synthesis of Process Systems. II. Heat Exchanger Networks.* Computers and Chemical Engineering, 1994. **18**: p. 679.

91. Allgor, R.J., L.B. Evans, and P.I. Barton, *Screening Models for Batch Process Development: I. Design Targets for Reaction/Distillation Networks.* Chemical Engineering Science, 1997. **Submitted, December 1997**.

92. Allgor, R.J., L.B. Evans, and P.I. Barton, *Screening Models for Batch Process Development: II. Case Studies.* Chemical Engineering Science, 1997. **Submitted, December 1997**.

Chapter 3

Formalization of the Batch Process Synthesis Problem

In Chapter 2 we discussed the process of batch process development. Our discussion focused primarily on the general aspects of the development process. In this chapter, we will discuss the formalization of the *batch process synthesis problem*, which will constitute the basis for the development of the systematic solution procedure, discussed in Chapter 4.

We will begin with the formal representation of the batch process structure through the use of the *State Task Network*. Following the description of the state task network in Section 3.1, in Section 3.2 we will proceed to formalize the representation of the States and of the Tasks as well as the relation between them through the formalism of the *finite automata*. Once we have formalized, mathematically, all the required components, in Section 3.3 we will show that the batch process synthesis problem is actually a combinatorial optimization problem and common solution methods for such optimization problems could be used for its solution. Finally, in Section 3.4 we will discuss how logic allows us to create a two-tiered optimization problem in which the logic can be used to reduce the complexity of the optimization problem through the systematic reduction of the solution search space.

3.1 Development of the State Task Network

As we have seen in Chapter 2, the batch process synthesis problem is a problem of identifying a sequence of operations to transform a given set of raw materials into a desired product with given specifications. How can this sequence of operations be represented?

A process design is essentially a recipe for the generation of the product material from the raw materials. The recipe includes the tasks that need to be performed, the conditions at which these tasks must be carried out and the ordering of the tasks. The linking of the tasks in a meaningful order and the description of the results of the application of the tasks to the materials is known as a *task network* or *transition network.*

The concept of a task or transition network has been used extensively in the area of process design. Its foundations are based on those of a cooking recipe, since cooking is, perhaps, the most familiar batch process.

> *Illustration 3.1*
>
> When one bakes a cake or prepares a meal for the first time, he generally follows a specific recipe. The recipe provides the detailed step by step instructions for preparing a dish. If we look closer at the recipe, it is constructed of a set of tasks or operations (add, mix, pour, bake, etc.) that transform the raw materials (eggs, flour, milk, cocoa etc.) into the final product (double-layer chocolate cheese cake). The recipe also states the equipment that we should use to perform the tasks (mix two eggs in a large bowl using a mixer) as well as provides a description of the intermediate states during the process (the cake batter should be smooth and have the consistency of honey).
>
> If we look at a recipe in terms of a sequence of chemical processing operations, the analogy is obvious. We take a given set of raw materials (Trienone, Methylmagnesium bromide) and through a set of operations (mix, react) in specialized equipment (jacketed-reactor, crystallizer), we obtain the desired product (Carbinol).

Recipes are precisely what the chemist generates when developing the chemical reaction synthesis route. Based on the idea of chemical recipes, Reklaitis [1] developed for batch processes what he called "*recipe networks.*" The recipe network attempts to describe the batch process in terms of a recipe i.e. a sequence of processing tasks, rather than as a network of unit operations, which is the typical form for the representation of a continuous process. Thus, a recipe network, i.e. a specific batch process, is represented as a simple block diagram where the blocks represent the tasks and the arcs represent the ordering of the tasks. In continuous processes, there is a direct correspondence between the operations and the equipment, while in batch processes many operations can be performed in a single piece of equipment. The recipe network attemptes to capture the task structure. The drawback of the recipe network is that it is rather ambiguous in its description of the process, using a simple line an block graphical description for the process thus, not providing a clear visual description of the process.

To overcome these weaknesses, Kondili et al [2] extended the idea of the recipe network and developed a representation for a batch process, called *State Task Network (STN).* A STN is a graphical description of a process recipe that captures the key features of the process i.e. the states of the materials at each point in the process, and the tasks required to transform a given state into the next state or states. The STN is a very useful representation for the batch process synthesis since it can be used to provide an abstract or highly detailed view of the process depending upon the level of detail of the task set we are using. The STN representation uses two types of nodes, *state nodes* and *task nodes*. The state nodes represent the statc of materials in the system and the task nodes represent the processing operations that transform the materials from one state to the next. For example, assume we have a process mixture composed of THF, water and Carbinol. The state of the mixture consists of composition information (40% THF, 50% water and 10% Carbinol), intensive properties such as temperature and pressure and physical characteristics such as phase information (an aqueous phase and a liquid phase containing dissolved carbinol). On the other hand, the task *heat* changes the state of a the process mixture from T to T + x. The new state also reflects any changes, depending upon the properties associated with the material's component compounds, such as the creation of a third phase due to the volatility of THF and water from the example material above. Figure 3.1 provides an example of a generic State Task Network. The circles in the figure represent the states and the tasks are represented by rectangles. An example of an actual STN for Plant 1 in the Fenvalerate production process [3] is given in Figure 3.2.

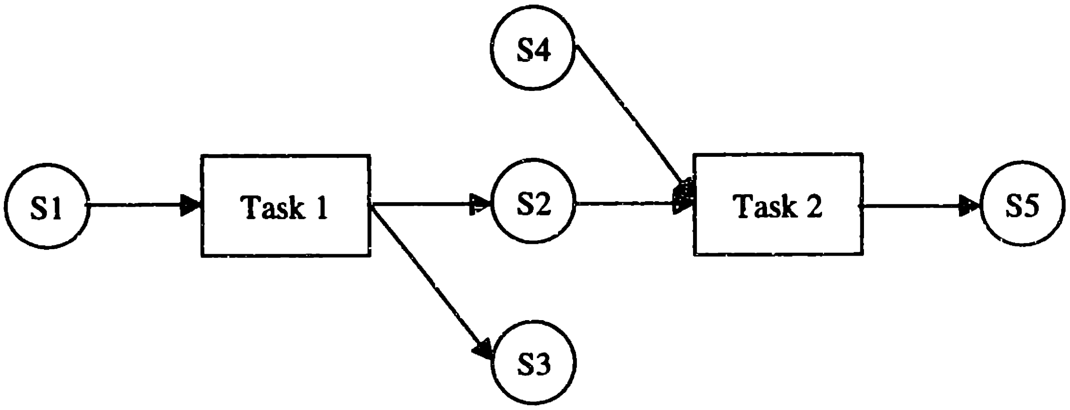


Figure 3.1 State Task Network

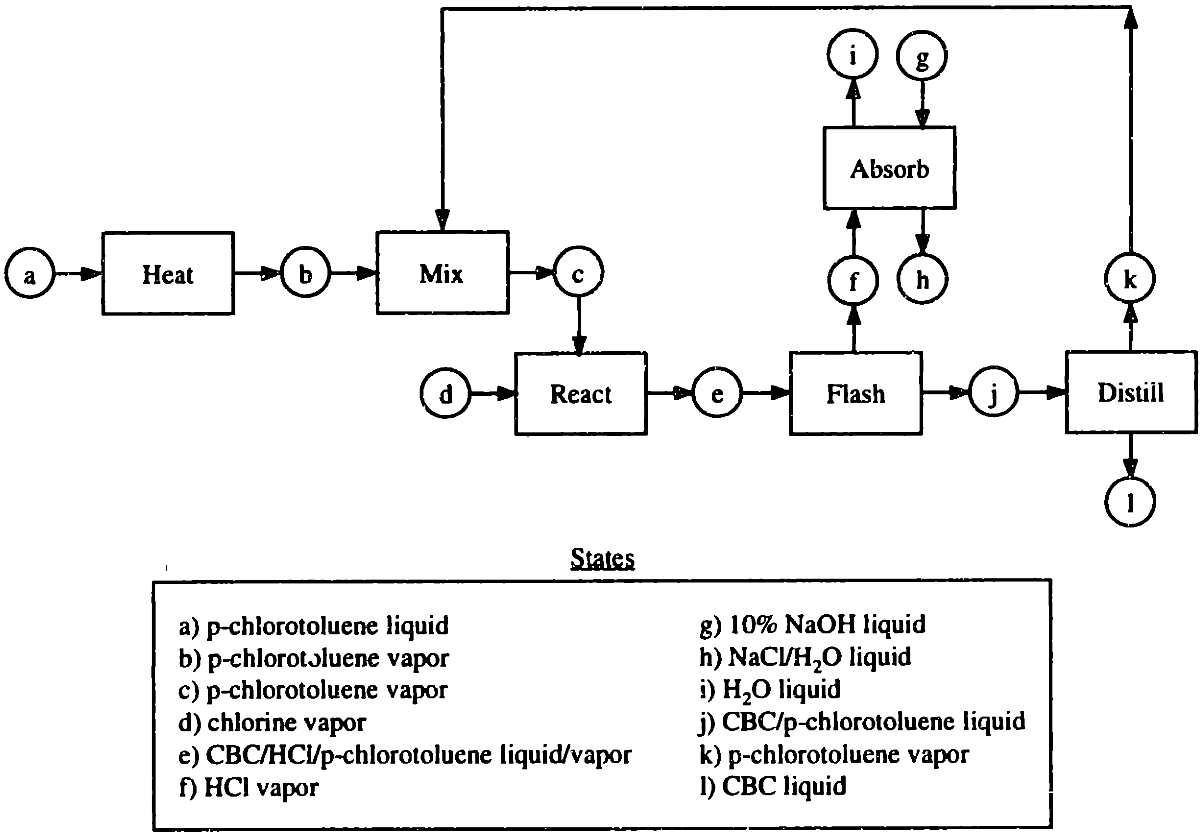


Figure 3.2 STN for Plant 1 of the Fenvalerate Production Process

Kondili et al [2] defined two rules for the construction of an STN:

1) A task has as many input (output) states as the number of different types of input (output) materials.
2) Two or more streams entering the same state are necessarily of the same quality. If mixing of different streams is involved in the process, then this operation should form a separate task.

It should be clear from the definitions of a state and a task that these rules are required to ensure consistency within the STN.

An important feature of the STN is that it does not limit in any way the level of detail at which the tasks must be modeled. This permits the construction of STNs at multiple levels of abstraction [4].

> *Illustration 3.2*
>
> Suppose that during the synthesis of a separation scheme for a given material mixture, we know that in order to perform the separation successfully a solvent replacement step is necessary. At this point we also know that we have a number of tasks and combinations of tasks that can perform the *solvent replacement* but we don't have the properties necessary to decide which tasks are feasible. We can insert an abstract solvent replacement task into the STN and specify the desired output states. Later, we can return to the abstract *solvent replacement* task and replace it with a specific task or a sequence of specific tasks that carry out the solvent replacement.

From the above discussion we see that the STN represents only states and tasks, it says nothing about the equipment which are used to perform these tasks. This is a very important feature of the STN concept, which allows the STN to be very flexible in its description of the process. The STN concept provides the explicit representation tools necessary for an unambiguous description of a batch process. A final advantage of using an STN to describe a batch process is that it provides a natural basis for the mathematical formalization of the batch process synthesis problem. The mathematical formulation of the batch process synthesis problem spawned from the STN is discussed in the next section.

3.2 STNs and Finite Automata: The Mathematical Formulation

While the STN is a convenient representation for the process, with states representing the state of a material at a particular point in the process and tasks representing a physico-chemical method for the transition from one set of states to another set of states, we must have the ability to construct models for each of these components. We must be able to model a material in a given state, as well as model the task that performs the transition, in order to generate the output state(s). With this in mind, it should be obvious that at the heart of the batch process synthesis problem is a question of modeling. In order to successfully perform batch process synthesis, we must have a means for modeling the process as well as a means for modeling the construction of the process.

In order to construct a useful and efficient model for batch process synthesis we would like the model to be simple, yet highly descriptive. One model that meets these requirements is the model of *finite automaton*, and we have used it to provide the formal mathematical basis for expressing the STN.

3.2.1 The Finite Automaton

A finite automaton [5] is defined as a 5-tuple $(S, \Sigma, \delta, s_0, F)$, where

1) S is a finite set called the states
2) Σ is a finite set called the alphabet
3) $\delta: S \times \Sigma \rightarrow S$ is the transition function
4) $s_0 \in S$ is the start state
5) $F \subseteq S$ is the set of accept states

We see from this definition of the finite automaton that we have all the corresponding components to the STN, described in the previous section. The state corresponds to the state of the process materials, the alphabet corresponds to the properties of the material which describe the state, the transition function corresponds to the task, the start state corresponds to the set of raw materials and the set of accept states corresponds to the desired product state. Let us now expand the discussion of each of the five elements that compose the definition of a finite automaton.

3.2.1.1 Finite Automaton: The States

The finite automaton provides us with a finite set of elements called the states. In batch process development the set of possible states for the system is theoretically infinite. Fortunately, for any given process, we do not need to consider all possible states. We must only consider those states that are achievable and relevant to the process. Thus, we also have a finite set of states, which are bounded by the tasks and the operating conditions. Within batch process synthesis, the states from the finite automaton correspond to the state of the materials within the system at different points in time. The state of a material is defined by the temperature, pressure, concentration of its component compounds and total amount. In order to be effective in solving the process synthesis problem, we should have the ability to model the materials used in the process. The more detailed the model the better and more accurate we can make our designs. To identify all the required operations, we must be able to determine the compositions and compound distributions among various phases. While the finite automaton does not help us in our construction of the models of the material states, it does emphasize the importance of having a distinct and accurate material model.

3.2.1.2 Finite Automaton: The Alphabet

The alphabet is the set of allowable input symbols to the system and can be thought of as the values of the set of variables that can fully characterize the state of the system. In batch process synthesis, the alphabet is a set of properties and attributes that describe the various states, which are recognized by the different tasks. For example, a subset of the alphabet used in batch process synthesis includes temperature, pressure and composition. Later we will show how we will use a concept known as the *preconditions* to identify a subset of the alphabet for each task.

3.2.1.3 Finite Automaton: The Transition Function

The *transition function* of a finite automaton manipulates the values within the alphabet thereby effecting a change to the system, moving it from one state to another. Within batch process synthesis, the transition functions are the tasks which transform the input states into the output states. The transition functions read the elements of the alphabet that describe the input states and then the transition function generates the output states and uses elements of the alphabet to describe the output states. For example, the *heat* operation operates over an input state reading its temperature, T. The output state contains the same material but the temperature has been changed to T + x. Along with the temperature change, there could be changes to other attributes of the materials, for instance the number of phases present could have increased to two depending upon the volatility of the compounds that are present. It should be clear from the description of the transition function that the transition function must have the ability to interact with the material model.

3.2.1.4 Finite Automaton: The Start State

As with any process we need to have a starting point from which we begin the development. Both the finite automaton and the STN start with an explicit declaration of a starting state. The start state, s_0, can be thought of as a representative state that includes the set of all possible states of all the raw materials that can produce the desired product. For batch process synthesis, we narrow this definition to say that s_0 represents the state of the key raw material. Given the key raw material and its state, the construction of the process can proceed. We should note that even though we define s_0 as the start state in the process that is being developed, it may not be the actual starting state. For example, beginning with the key raw material, we may decide that before performing the reaction, this key raw material must be dissolved in a solvent. Due to the properties of the key raw material it may be advantageous to add the key raw material to the solvent. We see that the actual starting state for the process would be the solvent state, to which the key raw material is added. Thus, we have one state from which we begin the design process and another state from which the process starts. The state of the key raw material will be referred to as the *start* state.

3.2.1.5 Finite Automaton: The Accept States

The set of accept states F, includes the states of the desired outcome. The accept states provide the synthesis with direction and the termination criterion, i.e. stop the synthesis when an accept state is reached. For batch process synthesis, F may include only one state, the state of the desired product, or it may also include any other states that the process should achieve, such as a state that contains specifications on a by-product material.

The power of the finite automaton model should not be lost on its simplicity. It provides the necessary language and facilities to completely describe the batch synthesis problem. Within this general description, the various types of models, used to describe the batch process synthesis problem and solution, can be articulated. It also fits well with the STN structure proposed above. The finite automaton provides the means to describe the process by placing an emphasis on the models while the STN provides a descriptive graphical representation.

The states and the transition functions will be formalized independently and then combined in the spirit of the finite automaton, to provide a complete formalization of the batch process synthesis problem. Charalambides et al [6] developed a mathematical formalism for batch processes, the description of the batch process design problem. We have used this formalism in order to provide a precise mathematical definition of the batch process synthesis problem, as described in the following paragraphs.

3.2.2 The Mathematical Formalization of a State

The mathematical formalization of a state is based upon the construction of a mathematical model to represent the state of a material. It is important that a *state* and a *material* not be confused. A material is composed of both a *state* and an *amount*. A state is independent of the amount of material. For example, lets assume we have 1 kg of a mixture of NaCl and water. The state of the material consists of the temperature, pressure, composition and phase. For this example lets assume we have standard conditions and a composition of 1% NaCl. Thus, the solution is not saturated, so we have a single liquid phase. The material includes all the state information as well as some measure of the amount of the material that is present. In this example the mass of the material was given.

Within batch processing a state represents the state of a material in the process. In order to formally describe a state we must first determine the desired attributes for our material model. A material can consist of multiple components and multiple phases. The state of the material is determined by the component compounds of the material as well as its intensive properties. In order to construct a mathematical formulation for the state of the material we need to accurately represent the properties of the material. Through a simple mathematical expression we represent a state as a function of its properties.

3.2.2.1 Mathematical Formalization: The State

Let s_i denote a specific state, D denote the set of all intensive properties and Q denote the set of component compounds. Then a state, s_i can be represented as,

$$s_i = s_i(D_i, Q_i) \tag{3.1}$$

where D_i represents the values for the elements of the set of intensive properties for state s_i and Q_i represents the set of component compounds.

3.2.2.2 Mathematical Formalization: Bounded Properties

Generally, the values of the variables describing the state are bounded to lie within a specific range of values. This may be due to material quality issues, such as avoiding the thermal degradation of a component, or process operation considerations, maintaining a compound in solution. The bounded properties are formally represented as:

$$d_{ij}^{\min} \le d_{ij} \le d_{ij}^{\max} \quad \forall i,\ j = 1,2,\cdots \tag{3.2}$$

where the index i represents the state and j represents an individual property.

3.2.2.3 Mathematical Formalization: The Material

As we stated earlier, a material is expressed in terms of a state and an amount. From the above description of a state we can derive a formal statement of a material. If we let m denote a material then we see in Equation 3.3 that a material is a function of both the state and an amount,

$$m(s_i, B_i) \quad \forall i \tag{3.3}$$

where B_i represents the amount of state i. Illustration 3.3 provides an explicit example of a material and its components.

> *Illustration 3.3*
>
> Lets assume that we have a mixture of ethanol and water. The state information is:
>
> Composition - 20% Ethanol, 40% Water
>
> Temperature - 20C
>
> Pressure - 1atm
>
> From the fundamental state properties and the physical properties of the component compounds other properties can be obtained such as phase and heat capacity.
>
> The amount is usually expressed in terms of mass or moles. Through a combination of the state and the amount information, we can completely characterize the material and obtain values for extensive properties such as volume.

Now that we have formalized the state and the material, we can move on to formalize the tasks which effect changes upon the states.

3.2.3 Task Formalization

The formalization of the tasks implies the development of explicit formal models. These models are the transition functions of the finite automata. We know that each task, o_n, takes place over a finite amount of time. Let us call this finite amount of time, the *duration of the task* and denote it using the symbol τ_n.

3.2.3.1 Mathematical Formalization: The Transition Function

The task models of batch operations are generally described through mixed systems of differential-algebraic equations. These systems can be expressed through the general form,

$$f_n(x_n(t), \dot{x}_n(t), y_n(t), u_n(t), v_n) = 0 \quad \forall n, t \in [0, \tau_n] \tag{3.4}$$

where $x_n(t)$ represents the vector of differential variables and $y_n(t)$ represents the vector of algebraic variables. In this formulation, $u_n(t)$ represents the vector of control variables and v_n represents the time-invariant parameters for items such as equipment specifications. The general expression for the transition functions does not capture the limits of the applicability of these models. In order to model physical systems, we must constrain the transition functions to operate over a limited region of state values. Thus, we must develop models for these constraints.

3.2.3.2 Mathematical Formalization: The Constraints

The value of the state variables and thus the extent of transitions are limited by safety and operability constraints. These constraints can be viewed as limits on the operating conditions of the particular task and are applicable throughout the duration of the task. For example, the temperature of the reaction mixture may need to be held above a certain value in order to keep one of the components dissolved in solution. These types of constraint models (path constraints) have the general form:

$$h_n(x_n(t), \dot{x}_n(t), y_n(t), u_n(t), v_n) \le 0 \quad \forall n, t \in [0, \tau_n] \tag{3.5}$$

Additionally, there are point constraints that only apply to the task at a finite set of times $\{t_{n\lambda}\}$. These constraints may be either equality or inequality constraints and have the general form:

$$\begin{aligned} g_n^E(x_n(t_{n\lambda}), \dot{x}_n(t_{n\lambda}), y_n(t_{n\lambda}), u_n(t_{n\lambda}), v_n) = 0 \\ \forall n,\ t_{n\lambda} \in [0, \tau_n], \lambda = 1, 2, \cdots \end{aligned} \tag{3.6}$$

$$g_n^I(x_n(t_{n\lambda}), \dot{x}_n(t_{n\lambda}), y_n(t_{n\lambda}), u_n(t_{n\lambda}), v_n) \le 0 \qquad \forall n, t_{n\lambda} \in [0, \tau_n], \lambda = 1,2,\cdots \tag{3.7}$$

An example of a point constraint is the termination of the collection of a product material during a batch distillation, once the still temperature exceeds a specified value.

3.2.3.3 Mathematical Formalization: Bounded Parameters and Variables

The time-invariant parameters and the operating condition variables are generally bounded to lie within a certain range. Additionally, the operating condition variables can depend upon the values of the time-invariant parameters. Examples of time-invariant parameters for a distillation operation are the reflux ratio and the number of trays in the column. An example of an operating condition variable that is a function of a time-invariant parameter is the heat supplied to a jacketed reactor. The amount of heat supplied to the reactor depends on the size of the heat transfer area of the reactor and the heat transfer coefficient. Equation 3.8 provides the bounded time-invariant representation, while Equation 3.9 represents the bounded operating-condition variables as a function of the time-invariant parameters.

$$v_n^{\min} \le v_n \le v_n^{\max} \quad \forall n \tag{3.8}$$

$$u_n^{\min}(v_n) \le u_n(t) \le u_n^{\max}(v_n) \quad \forall n, t \in [0, \tau_n] \tag{3.9}$$

Now that we have provided a formal mathematical description of the materials, states, task and constraints, we are ready to show how the formalization extends to the relation between the individual components of the formalization.

3.2.4 State-Task Relation

Let S_n^{in} denote a set of input states to a task, o_n, where S_n^{in} is a subset of S. Each task or transition function transforms the set of states, S_n^{in}, into a new set of states, S_n^{out}, which is also a subset of S. The properties of the output states, $s_i \in S_n^{out}$, produced by task, o_n, are functions of the final values of the task variables. The state intensive variables are represented by equations of the form

$$d_{ni} = F_n(x_n(\tau_n), \dot{x}_n(\tau_n), y_n(\tau_n), u_n(\tau_n), v_n) \quad \forall n, i = 1,2,\cdots \qquad (3.10)$$

The amount of the material at state, s_i, after task, o_n, is completed, depends on the final values of the task variables as shown in Equation 3.11,

$$B_{ni}^{out} = \tilde{F}_n(x_n(\tau_n), \dot{x}_n(\tau_n), y_n(\tau_n), u_n(\tau_n), v_n) \quad \forall n, \quad i = 1,2,\cdots \qquad (3.11)$$

where, B_{ni}^{out} represents the amount of state, s_i, produced by task, o_n.

The intensive and quantity variables can also be related to each other and to additional time-invariant parameters. The relation between intensive and quantity variables is expressed through the formulation of constraints. An example of a relation between quantity variables is the specification of a feed ratio of raw materials. The general mathematical formulation of these constraints is given in Equation 3.12.

$$G_n\left(v_n, \{D_i, B_{ni}, s_i \in S_n^{in}\}, \{D_i, B_{ni}^{out}, s_i \in S_n^{out}\}\right) \le 0 \quad \forall n, i \qquad (3.12)$$

The set of equations and inequalities (3.1) through (3.12) provides the mathematical formalism for all of the components that underlie the STN description of a batch process and the interaction between the components through the formalization of transition functions and constraints. We can now go ahead and use the mathematical formalization to show that the batch process synthesis problem is actually an optimization problem and the solution to the optimization problem provides the formal generation of an STN.

3.3 The Optimization Problem for Batch Process Synthesis

From the mathematical formalization of the components in the previous sections we see that the components depend on a large number of variables and parameters, each of which must be specified in order to complete the design. Along with the specification of all the variable and parameter values, we have the overriding problem of selecting the tasks, operating conditions, ordering of the tasks and equipment. All of these choices are subject to both overall process constraints, such as the product specifications, as well as localized constraints, e.g. the volume of material going into a tank cannot exceed the capacity of the tank. The problem of determining the optimal set of operations, operating conditions, ordering of operations and equipment from a large set of possible alternatives is by definition a combinatorial optimization problem. The goal of this section is to describe the type of combinatorial optimization problem that we have in batch process synthesis and suggest possible solution methods.

Further analysis of the batch process synthesis problem shows that it contains both integer decision variables, such as whether or not to include a task in the process, and continuous variables, e.g. variable representing material flows. Additionally, from the study of the operational model, we know that the performance equations of most of the tasks will be nonlinear. Generally, the objective for the process synthesis problem is to produce the desired product to meet the given specifications at a minimum cost. Of course, there may be a number of additional objectives, which may be incorporated as simultaneous objectives, or as constraints on the stated primary objective. For now, lets assume the objective is the minimization of cost of production, and we will revisit the construction of the objective function later. From these attributes, it stems naturally that the batch process synthesis problem can be formulated as a *Mixed-Integer NonLinear Programming* Optimization problem (MINLP). Since batch processes are inherently dynamic, as we have mentioned in Chapter 2, more recent work has begun to focus on formulating the problem as a *Mixed-Integer Dynamic Optimization* problem [7]. The problem can only be formulated as a MINLP if we assume we can discretize the time domain of the system. Some authors have indicated that the solution of the discretized problem using traditional MINLP solution strategies merely results in process improvement and may not determine the optimal process [7]. Since the work of this thesis does not depend on whether the problem is solved as a MINLP or an MIDO, and since this is an issue still under debate, let us continue our discussion using the more common approach of the MINLP formulation. Grossmann and coworkers [8] have suggested various approaches to solve process synthesis problems through the application of a wide variety of optimization routines to MINLP formulations of the problem. For the discussion of the MINLP solution methods we will use a common generic MINLP problem representation articulated by both Floudas [9] and Grossmann [10]:

$$\begin{aligned} &\min_{x,y} Z = f(x,y) \\ &s.t.\ h_j(x,y) = 0 \quad j \in J \\ &\quad g_j(x,y) \leq 0 \quad j \in J \\ &\quad x \in X \subseteq \Re^n \\ &\quad y \in Y \end{aligned} \tag{3.13}$$

In this formulation we assume that f, h and g are convex, differentiable functions. Later we will discuss the implications if any of the, f, h or g, is nonconvex. Here x and y are the continuous and discrete variables, respectively. The set X is assumed to be a compact set, e.g. $X = \{x \mid Dx \leq d, x^L \leq x \leq x^U\}$; the discrete set Y is a set of integer points, $Y = \{y \mid y \in \mathbf{Z}^m, Ay \leq a\}$. For process synthesis problems, the values of the elements of y are limited to 0 and 1, i.e. $y \in \{0,1\}^m$.

In the remainder of this section we will provide a short review of a number of the most common solution methods for MINLP problems and present some of the difficulties arising from the use of optimization routines to solve batch process synthesis problems. We will also show how logic can be exploited in order to make the solution of such optimization problems more efficient. In Chapter 6, we will show how the scope of extremely large process synthesis optimization problems can be reduced through the Means-Ends Analysis with Nonmonotonic Planning methodology.

3.3.1 Solution Algorithms for MINLP Problems

At this point we will discuss some of the most common solution methods for MINLP problems. The fact that MINLP problems are NP-hard makes them some of the most difficult problems to solve. Fortunately, there are a number of solution methods that can provide efficient, more or less, approaches for this class of problems. Two comprehensive sources of the various solution methods for solving MINLP problems in chemical engineering are those by Floudas [9] and Grossmann [10]. In this section, we review four of the main approaches for solving MINLP problems: *Branch and Bound* (BB) [11], *Generalized Benders Decomposition* (GBD) [12], *Outer Approximation* (OA) [13] and *Extended Cutting Plane* (ECP) [14].

3.3.1.1 Tree Search: Branch and Bound

The Branch and Bound (BB) strategy [11] is a form of the class of search methods known as *tree search methods*. Tree search methods are methods which can solve problems whose space of possible solutions (i.e. search space) can be represented using a tree structure. Before we describe the BB strategy in detail, let us first describe some of the general characteristics of tree search methods.

Tree Search: The Tree

Frequently chemical production processes can be represented using a tree structure. A tree structure consists of a root node, intermediate nodes and terminal nodes. The tree structure provides us with a convenient representation for classes of solutions that can be generated. The root node represents a class containing all possible solutions to the MINLP problem. Each intermediate node represents a subclass of the root node and contain all remaining possible solutions that are available after particular discrete decisions have been fixed. The terminal nodes mark the completion of a solution path. Figure 3.3 contains an example of a binary tree. Notice that each arc is labeled with a y variable equal to a zero or one. These y variables are the discrete decision variables and correspond to the binary decisions made while developing the process.

Within a tree structure the root node is considered to be at the zeroth-level. The generation of the intermediate nodes within the tree is called *branching*. Each subsequent branching produces nodes one level higher i.e. the first branching is called Level-1, the second Level-2, and so on. The example in Figure 3.3 has 4 levels including the root node. Also, in Figure 3.3 we have chosen a binary tree, but we should note that it is possible to construct trees that have higher levels of branching. It should also be clear from this example that, since all nodes emanate from the root node, the root is the highest solution class, containing all possible solutions.

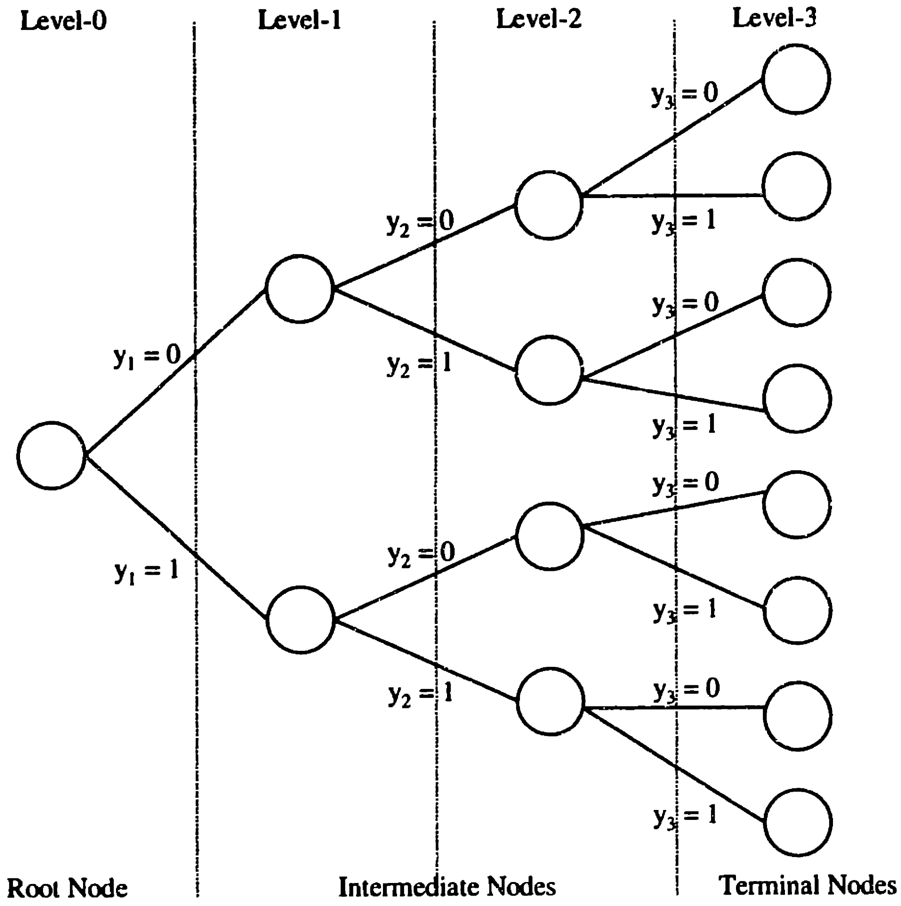


Figure 3.3 Binary tree

The binary tree provides a convenient method for evaluating the effect decisions can have on the outcome of a process. The decisions can be viewed as integer values representing the decision made. In this case, we must either select one branch or the other. For process synthesis, we are generally interested in minimizing the cost. We assume that the branches represent the cost of an additional operation and the nodes represent the cost of the process including all the operations along the path leading to that node. The cost associated with the processing tasks for each branch along a particular path is determined by summing all the costs along the particular path leading to that node. For example, if we assume that each node represents an operation, we can estimate the cost for that process, assuming that we also have the material costs, by adding the cost for each operation to the material cost. The method generates partial costs for a process by summing up the costs for the branches up to a particular point in the tree [15]. This partial cost is the element which is used to generate the lower bound on the problem. We develop the process based on the lowest partial cost at a particular point in the search.

Tree Search: The Properties

The tree structure of the problem provides us with the property that the cost increases monotonically along each path of the tree. The monotonic increase in cost implies two additional properties [15]:

1) For an intermediate node in the tree, its partial cost is a lower bound on the cost of any successor nodes. Each additional node causes the process to incur additional cost.
2) For a terminal node, its total cost is an upper bound to the cost of the original problem because it represents a feasible integral solultion. The terminal node defines a solution that may or may not be optimal.

These properties allow the pruning of any node in the tree for which the lower bound on the cost is greater than or equal to the current upper bound. Thus, we no longer search any path whose cost is greater than the upper bound. It should be obvious that this can significantly reduce the search time versus a complete enumeration of all paths.

We can generally represent the optimization of batch process synthesis problems through a tree structure. Each node in the tree represents an operation and its corresponding models. Figure 3.4 contains an example of the tree representation of the alternatives for a small batch process. In the figure, the tree represents the production of a single product. We see that all processes stem from the root node which in this case, we could call "processes for the production of X." Also, notice that depending upon the different decisions made, there are different production paths. In this example, there are a total of eleven alternative processes that could be formulated.

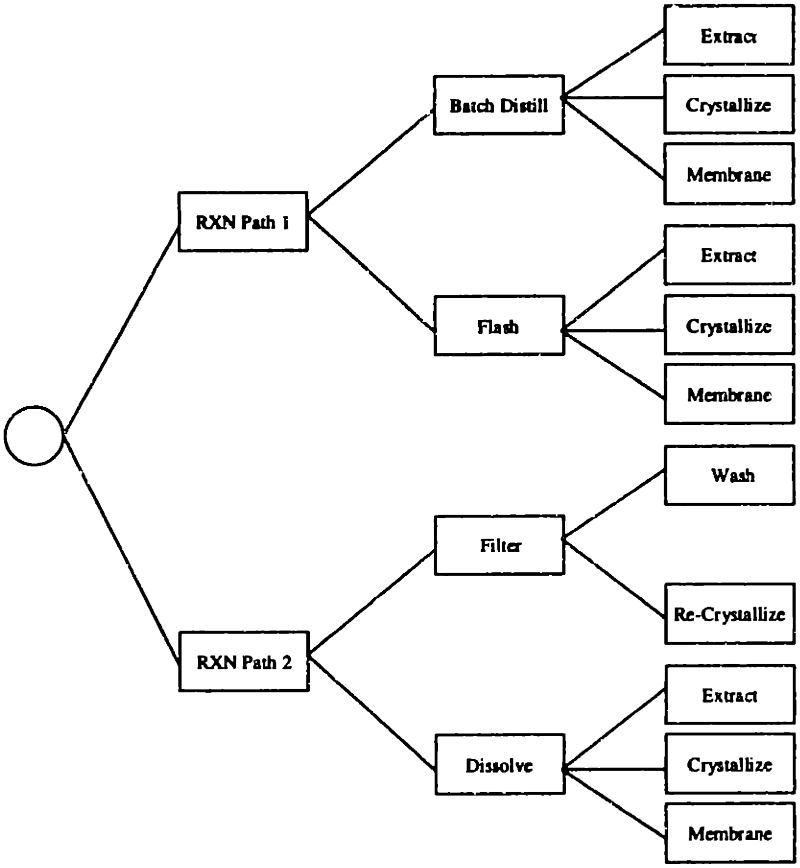


Figure 3.4 Batch Process Task Tree

Clearly, complete enumeration of the tree is not practical except for small trees. In order to solve large problems, we must use selective enumeration of alternatives. There are two basic methods for evaluation of the nodes, *depth-first* search and *breadth-first* search.

In depth-first search, we begin at the root node, and evaluate its cost. We then chose one branch and evaluate the cost of the next node along that branch. This process is repeated until a terminal node is reached. After reaching a terminal node, we backtrack to the last node that has a branch, which has not been examined, and evaluate the next node along that branch. If the cost of including that node in the plan is lower than the previous completed path we continue with the development along that path, otherwise we backtrack to the next node that has an unexamined branch and continue the development in a depth-first manner. This process of backtracking and then developing the process to the next terminal node is repeated until we terminate the search or we have enumerated all alternatives.

In the breadth-first search, we begin at the root node and evaluate the first node on each branch leading from the root node. We then select the node with the lowest cost and evaluate all its successor nodes. Again, we select the node with the lowest cost and repeat the process until we achieve a terminal node that has a cost less than all other intermediate terminal nodes.

In general, depth-first search strategies are used for large problems since the number of nodes that must be stored is generally smaller than for the breadth-first search. Biegler et al [15] (p504-506) provides an excellent examples of depth-first and breadth-first tree searches for a separation scheme. Now that we have discussed the properties of implicit enumeration strategies, let us move on and discuss the BB strategy.

Branch and Bound: Strategy

For the BB strategy, the nodes of the tree have a different representation. The nodes represent a degree of relaxation. The cost at each node corresponds to the degree of relaxation of the problem, thus the cost is not the sum of the costs along each path. Each additional node represents a more constrained optimization problem so that the solution of the successor nodes must be greater than or equal to the predecessor nodes (in a minimization context).

The BB strategy begins with the *relaxation* of the integer constraints. This implies solving a NonLinear Programming (NLP) problem and allowing the variables that were constrained to be integers to take on continuous values. If the discrete variables in the solution to the relaxed problem take on integer values, then we have found the optimal solution and the search is stopped. If any discrete variable, y_I, takes on a non-integer value, then we must construct a new relaxation problem which consists of two subproblems, one which contains an additional upper bound constraint, $y_i \leq \alpha^k{}_i$ and one which contains an additional lower bound constraint, $y_i \geq \beta^k{}_i$. The determination of the two subproblems is called *branching* and the NLP problem is called a node. The NLP relaxation has the following form [10],

$$\begin{aligned}
&\min_{x,y} Z_{LB}^{k} = f(x,y) \\
&s.t.\ h_j(x,y) = 0 \ \ j \in J \\
&\quad g_j(x,y) \le 0 \quad j \in J \\
&\quad x \in X, \quad y \in Y_R \\
&\quad y_i \le \alpha_i^k \quad i \in I_{FL}^{k} \\
&\quad y_i \ge \beta_i^k \quad i \in I_{FU}^{k}
\end{aligned} \tag{3.14}$$

where Y_R is the continuous relaxation of the set Y, and I^k_{FL}, I^k_{FU}, are the index subsets of the integer variables y_i, $i \in I$, which are restricted to lower and upper bounds, α^k_i, β^k_i, at the kth step of a branch and bound enumeration procedure. Note that $\alpha^k_i = \lfloor y^l_i \rfloor$, $\beta^k_i = \lceil y^m_i \rceil$, $l < k$, $m < k$, where y^l_i, y^m_i, are noninteger values at a previous step.

Branching eliminates a region of the discrete search space and ensures that none of the feasible solutions are excluded. Each of the subproblems is then solved as a continuous relaxation. This process is continued until an integer feasible solution is found. Once such a solution is found, the value of the objective function becomes the upper bound on the minimum value of the objective function. This allows us to remove all the nodes that have objective function values larger than the upper bound. We also remove nodes whose objective function does not have a feasible solution. The branching process continues until all the nodes have either been removed or expanded to terminal nodes. The optimal solution is then the problem that has an integer feasible solution with the lowest value of the objective function.

It should be clear from the description above that BB should only be used if the NLP subproblems are inexpensive to solve or if the problem is small. Additionally, the NLP subproblems should represent a tight relaxation in order to perform the implicit enumeration efficiently. Typically, this is not so for MINLP problems so BB performs poorly and is not used [9]. In the discussion above we did not mention how one should select the branching variables. According to Floudas [9] "Since robust methods for selecting the branching variables are not available, a common practice is to establish a set of priorities provided by the user."

3.3.1.2 Outer Approximation

The OA strategy by Duran and Grossmann [13] was developed to reduce the computational expense involved in solving a large number of NLP subproblems as required by the BB method. OA and GBD are "decomposition" algorithms that search the discrete space in a different manner to BB. In practice, they work much better than BB, primarily due to the poor NLP relaxation of BB. The following discussion of the OA method follows closely the description given by Grossmann [15].

The fundamental idea underlying the OA strategy is "to solve an alternating sequence of NLP and MILP master problems."[15] The NLP subproblems are derived by fixing the binary variables at some selected value. Then an optimization of the objective function over the continuous variables is performed to obtain an upper bound to the original MINLP problem. The NLP subproblem for a fixed y^k has the following form [10],

$$\begin{aligned} &\min Z_U^k = f(x, y^k) \\ &s.t.\ g_j(x, y^k) \leq 0 \quad j \in J \\ &x \in X \end{aligned} \tag{3.15}$$

where the binary variables, y, are separable. As Floudas [9] points out, this formulation assumes that the "nonlinear equalities can be eliminated algebraically or numerically." For the case where this NLP subproblem has a feasible solution, the solution provides an upper bound to the original objective function.

In the case where the NLP problem is infeasible, x^k is determined by solving the following problem [10],

$$\begin{aligned} &\min u \\ &s.t.\ g_j(x, y^k) \leq u \quad j \in J \\ &x \in X, \quad u \in \Re^1 \end{aligned} \tag{3.16}$$

The MILP master problem is solved to obtain a global linear approximation to the MINLP problem. The master MILP problem has the form [10],

$$\begin{aligned}
&\min Z_L^k = \alpha \\
&s.t.\, \alpha \ge f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\
&g_j(x^k, y^k) + \nabla g_j(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \le 0 \quad j \in J^k \\
&k = 1, \ldots, K, \quad x \in X, \quad y \in Y, \quad \alpha \in \Re^1
\end{aligned} \tag{3.17}$$

where Z_L^K yields a valid lower bound to the original optimization problem for the case where the problem is convex and separable. The master problem underestimates the objective function and overestimates the nonlinear feasible region. Also, as part of the master problem, "the linear approximations of the nonlinear equations are relaxed as inequalities."[15] "The MILP master problem accumulates the different linear approximations of the previous iterations to produce an increasingly better approximation of the original MINLP problem. At each iteration the master problem predicts new values of the binary variables y and a lower bound to the objective function Z." [15] The OA method is said to have converged when the lower bound and upper bound are within a specified tolerance, ε. An excellent step by step description of the OA strategy is given by Grossmann (pp. 764-766) [15]. Finally, it should be noted that the OA strategy requires convexity of objective function and the constraints to ensure a global optimum. The issue of nonconvex functions will be discussed later in this chapter.

3.3.1.3 Generalized Benders Decomposition

The GBD method is the predecessor to Grossmann's OA method. As with the OA method, the GBD method operates by solving a sequence of NLP subproblems and MILP master problems. The difference between the GBD method and the OA method lies in the definition of the MILP master problem. The GBD method only considers the active inequality constraints and the problem is limited to the discrete variables with their corresponding continuous variables, i.e. the set $x \in X$ is disregarded.[10] GBD is also slightly less restrictive than OA since GBD does not require separability of the binary variables. The GBD master problem is generated by projecting the original problem in the y-space. The projection is performed by deriving a Lagrangian function parameterized in the discrete variables for each subproblem. Equation 3.18 gives the projected master problem [16],

$$\begin{array}{ll} \min Z_L^k = \alpha & \\ s.t. \alpha \geq f(x^k,y^k) + \nabla_y f(x^k,y^k)^T (y - y^k) & \\ +(\mu^k)^T [g(x^k,y^k) + \nabla_y g(x^k,y^k)^T (y - y^k)] \quad k \in KFS & \\ (\lambda^k)^T [g(x^k,y^k) + \nabla_y g(x^k,y^k)^T (y - y^k)] \leq 0 \quad k \in KIS & \\ y \in Y, \quad \alpha \in \Re^1 & \end{array} \tag{3.18}$$

where KFS is the set of feasible subproblems and KIS is the set of infeasible subproblems. In each iteration the solution of the master problem provides a new set of binary variables to be used in the next NLP subproblem. As with OA, GBD converges when the upper bound generated from the solution to the NLP subproblem and the lower bound given by Z_L^k are within a specified tolerance.

Since the OA method and the GBD method are so similar, we would like to compare the two methods. Duran and Grossmann [13] show that the lower bounds given by the OA method are always greater than or equal to the lower bounds of GBD. Thus, the OA method requires fewer solutions of NLP subproblems and MILP master problems. While this is clearly a benefit of the OA method, the MILP master problems for the OA method are more expensive to solve. If the NLP subproblems of the GBD are inexpensive to solve, the GBD method may converge faster. As with the OA strategy, GBD can only be guaranteed to converge to a global optimum if the convexity conditions mentioned in the discussion of the OA strategy hold.

3.3.1.4 Extended Cutting Plane

The ECP method is a *sequential linear programming* method. It uses a series of MILP problems, equivalent to the master problem within GBD and OA, to solve the MINLP problem. The EPC method linearizes the objective function and constraints about some initial point (x^k,y^k). The MILP is then solved to obtain a new solution (x^{k+1},y^{k+1}). The original constraints are then evaluated at (x^{k+1},y^{k+1}) and the most violated constraint is determined through $g_m(x^{k+1},y^{k+1}) = \max[g_j(x^{k+1},y^{k+1})]$. If the evaluation of the most violated constraint lies within some tolerance, ε, the optimal solution is

(x^{k+1},y^{k+1}). If not, we relinearize this constraint about the point (x^{k+1},y^{k+1}) and add this constraint to the list of constraints for the previous MILP. We then solve the MILP subject to the new set of constraints. This procedure is repeated until the tolerance, ε, is below the prespecified value. The ECP strategy works to obtain convergence of the discrete and continuous variables simultaneously, thus, a drawback of the ECP method is that it may require a large number of iterations to obtain convergence due to the simultaneous convergence of the discrete and continuous variables.

3.3.1.5 Other Methods

There are a number of other methods and extensions to the MINLP solution methods described above to improve the applicability and efficiency of the methods. Most of these methods are variations of the above methods. For a detailed description of these extensions see Grossmann [10] and Floudas [9]. One extension of the MINLP methods that is of particular interest for the solution of process synthesis problems is the use of logic to represent the MINLP problem. The next section describes the logic-based approach.

3.3.2 Logic-Based MINLP Representation

From the above discussions we see that MINLP problems are combinatorial and due to the large number of alternatives for process synthesis problems, the size of combinatorial problem can be very large. These large scale problems and the fact that these problems are NP-complete, makes the determination of an optimal solution very difficult. In order to reduce the search space for the optimization we must reduce the number of possible alternatives within the process. One representation we can use to reduce the search space of possible alternatives is the logic-based representation.

3.3.2.1 Propositional Logic

The logic-based representation is especially suited to modeling of constraints and heuristics and is grounded in propositional logic. "A proposition is any logical expression and consists of a set of clauses P_i, $i = 1,...,r$ that are related by the logical operators OR, AND, IMPLICATION."[17] Propositional logic is a logic that is based upon truth of statements. The values of a proposition can only be true or false. It should be clear that this makes propositions a natural choice for representing integer decisions. Generally, true is assigned the value 1 and false is assigned to 0. For example, each decision of whether or not to include an operation in a particular process plan can be modeled by an integer decision variable i.e. a value of one indicates including the operation in the plan while a value of zero indicates not including the operation in the plan.

Propositional logic provides a useful tool for capturing qualitative knowledge. By modeling the constraints and heuristics using logic, logical inference can be applied to reduce the search space for the solution. Raman and Grossmann [17] have shown "qualitative knowledge that can be expressed in propositional logic form has an equivalent representation as linear equations and inequalities." This makes propositional logic extremely useful in the formulation of MINLP problems since it is often easier to state the constraints as propositional logic statements and then convert them to their mathematical equivalent. Another important feature of the propositional logic representation is that uncertain information can also be handled. This feature can be especially useful in the representation of heuristics. In the following discussion of propositional logic we will use the notation used by Raman and Grossmann [17] in order to maintain consistency.

The conversion of propositional statements into linear mathematical form is facilitated through the use of a few basic equivalent relations. Table 3.1 from Biegler et al [15] provides the equivalent relations to convert arbitrary propositional logic expressions into linear constraints.

Logical Relation	Comments	Boolean Expression	Representation as Linear Inequalities
Logical OR		$P_1 \vee P_2 \vee \ldots \vee P_r$	$y_1 + y_2 + \ldots + y_r \geq 1$
Logical AND		$P_1 \wedge P_2 \wedge \ldots \wedge P_r$	$y_1 \geq 1$ $y_2 \geq 1$ $\vdots$ $y_r \geq 1$
Implication	$P_1 \Rightarrow P_2$	$\neg P_1 \vee P_2$	$1 - y_1 + y_2 \geq 1$
Equivalence	P_1 if and only if P_2 $(P_1 \Rightarrow P_2) \wedge$ $(P_2 \Rightarrow P_1)$	$(\neg P_1 \vee P_2) \wedge (\neg P_2 \vee P_1)$	$y_1 = y_2$
Exclusive OR	Exactly one of the variables is true	$P_1 \underline{\vee} P_2 \underline{\vee} \ldots \underline{\vee} P_r$	$y_1 + y_2 + \ldots + y_r = 1$

Table 3.1 Constraint Representation of Logic Propositions and Operators [15] p.515

The fundamental component of a propositional statement is the literal which "is a single variable that can assume either of two values, true or false."[17] The logical relations are used to relate the literals. For example, if we have some expression E_1, whose value is true and another E_2, whose value is false, the value of the expression created through the application of the logical OR operator is true i.e. $E_1 \vee E_2 \Rightarrow$true. The value of the resulting expression created through the use of the logical AND operator is false i.e. $E_1 \wedge E_2 \Rightarrow$false. The truth tables defining the logical relations can be found in any discrete mathematics text book.

There are two possible representations for propositional logic, *conjunctive normal form* (CNF) and *disjunctive normal form* (DNF). A *conjunction* is a series of elements connected by AND operators and a *disjunction* is a series of elements connected by OR operators. CNF is a conjunction of disjunctive clauses. For example, let Q denote a generic clause, then CNF is denoted by $Q_1 \wedge Q_2 \wedge \ldots \wedge Q_s$. A disjunctive clause is a disjunction of literals represented by $P_1 \vee P_2 \vee \ldots \vee P_r$. In contrast to CNF, DNF is a disjunction of conjunctive clauses represented by $Q_1 \vee Q_2 \vee \ldots \vee Q_r$, where the conjunctive clause is represented by $P_1 \wedge P_2 \wedge \ldots \wedge P_s$.

The first step of converting logical expressions into its equivalent mathematical expression is to convert the logical expression into CNF. This procedure was formalized by Clocksin and Mellish [18] and the following three step procedure for converting logical expressions into mathematically equivalent expressions was given by Raman and Grossmann [17]. Beginning with a logical proposition:

1) Replace the implication by its equivalent disjunction: $P_1 \Rightarrow P_2 \Leftrightarrow \neg P_1 \vee P_2$
2) Move the negation inward by applying DeMorgan's Theorem:
$\neg(P_1 \wedge P_2) \Leftrightarrow (\neg P_1 \vee \neg P_2)$
$\neg(P_1 \vee P_2) \Leftrightarrow (\neg P_1 \wedge \neg P_2)$
3) Recursively distribute the "OR" over the "AND" by using the following equivalence: $(P_1 \wedge P_2) \vee P_3 \Leftrightarrow (P_1 \vee P_3) \wedge (P_2 \vee P_3)$

Once the proposition has been expressed in CNF, it is then straightforward using the associative and distributive laws to transform the expression from CNF to DNF. Also, using Table 3.1 it is simple to convert an expression from CNF or DNF to its mathematical form. Illustration 3.4 provides an example of a constraint from a batch process, which is converted into a logical expression and then linear mathematical constraints.

Illustration 3.4

The process ordering constraint we wish to convert is *if crystallization is selected then filtration and drying must be selected.* The conversion is performed in four steps. First we transform the constraint into a logical proposition by letting P_C = select crystallization, P_F = select filtration and P_D = select drying. The logical expression then becomes the following implication: $P_C \Rightarrow P_F \wedge P_D$. We then remove the implication by replacing it with its equivalent disjunction: $\neg P_C \vee (P_F \wedge P_D)$. Next we distribute the OR over the AND and obtain: $(\neg P_C \vee P_F) \wedge (\neg P_C \vee P_D)$. Finally, we assign the corresponding 0-1 variables and using Table 3.1 we generate the linear mathematical constraints: $y_C - y_F \leq 0$ and $y_C - y_D \leq 0$.

Illustration 3.4 shows how to convert certain logical propositions into their equivalent linear mathematical form. Propositional logic can also be useful in the conversion of uncertain information as contained in heuristics into their equivalent linear mathematical form. Heuristics, along with being uncertain, also have the ability to contradict each other and this must also be taken into account. In order to use heuristic rules, the heuristic is modeled as a propositional logic clause as discussed above. Post [19] has proposed modeling the potential violation heuristics as logical relations constructed of the form: Clause OR V. This is just the simple statement that either the cause is true or it is being violated i.e. V is true. Since we are using V as a literal then it must also take only 0-1 values. For example: the constraint given in the above illustration would then be expressed as $\neg P_C \vee (P_F \wedge P_D) \vee V$ and the linear mathematical constraints would then become: $y_C - y_F - v_1 \leq 0$ and $y_C - y_D - v_1 \leq 0$.

But, as we know, different heuristics have different levels of importance. Raman and Grossmann [17] have developed a method to incorporate heuristic knowledge into the formulation. We can embed the importance of the heuristics into the logical structure by applying a non-negative penalty w, for the violation of a given heuristic. The more important or certain the heuristic, the higher the penalty for its violation. This allows the formulation of an MILP incorporating uncertain knowledge whose objective function first attempts to satisfy all the logical relationships (Z=0) and if that is not possible, then to find a solution that minimizes the total penalty. The formulation of the MILP is given in Equation 3.20 [17].

$$\begin{aligned} & \min Z = w^T v \\ & s.t.\ Ay + v \geq a: \quad \text{heuristics} \\ & \qquad By \geq b: \quad \text{logical facts} \\ & \qquad y \in \{0,1\}^n, \quad v \geq 0 \end{aligned} \tag{3.19}$$

The major drawback of assigning penalties to the heuristics is that the penalties add an element of subjectivity to the formulation.

3.3.2.2 Logical Inference

The use of logic allows us to perform logical inference. Inference can be quiet powerful in allowing us to eliminate regions from the search space and make the solution of the optimization more efficient. Grossmann and coworkers [20-22] have developed both *symbolic* and mathematical logical inference methods for optimization problems.

The application of symbolic inference can be best applied to the branch and bound algorithm. Both the DNF and the CNF representations can be applied to the BB method but as will be discussed, certain criteria should be met in order to determine which representation to apply. In order to fully exploit the benefits of the logic representations, the BB method implements the logic during branching in an attempt to perform logical chaining in order to reduce the number of branches that need to be examined. Raman and Grossmann [20] have performed an analysis of the symbolic solution using BB. The description of this method is summarized below.

Let us first examine the effect of logical inference on the branching. Once the constraints for a given node have been converted into either DNF or CNF, we can use logic to infer some information about the structure of the problem. In order to branch most efficiently, we would like to prune fruitless branches as early as possible during the search for a solution. One scheme is to search among the branches that will lead most quickly to integer solutions. Within the DNF representation "the variable that has the highest priority for branching is the one that occurs the least number of times in non-negated or negated form in the disjunctive representation of the logic at the current node of the tree."[20] This variable is chosen for branching because disjunctive logic dictates that it has the lowest number of feasible solutions. At the same time, as implied by DNF, this variable has the potential to eliminate the largest number of clauses, thus reducing the size of the equation to be solved in terms of the remaining variables. Using the CNF representation, the highest priority variable for branching is the variable that occurs in the largest number of short clauses. Here "the greater the number of short clauses the branching variable appears in the greater the amount of variables that are likely to be fixed at the resulting node."[20]

Through logical chaining, both the DNF approach and the CNF approach use logical inference to fix as many of the other binary variables as possible. "The CNF based approach utilizes unit resolution [18] for the purpose of making inferences, while the DNF approach simplifies the current Boolean equation, and searches for variables that occur with the same sign in all the clauses and setting them to that value at the node."[20] For formal descriptions of the DNF approach and the CNF approach see Raman and Grossmann [20].

This formal strategy for pruning branches exploits what is called the *domino effect* of binary decisions. In many cases once we make a decision, that decision propagates its effect through the plan and necessarily fixes other decisions in the plan, just as in a row of dominos, if they are spaced properly when the first is knocked over its effect is propagated through the row and eventually they are all knocked over. By mathematically constructing the relations between the decisions through the use of CNF and DNF entire regions of the search space can be eliminated. In Chapter 4 we discuss the use of Means-Ends Analysis and Nonmonotonic Planning and show how these concepts relate to the logic based identification of the feasible search space.

It is important to note that the DNF approach and the CNF approach are better suited for different types of problems. The DNF approach should be used for problems in which the number of feasible design solutions is much less than the number of combinations of binary variables, while the CNF approach can handle large numbers of feasible alternatives but generally generates a larger number of branchings. The DNF approach provides another advantage over the CNF approach. The procedure for determining the fixed variables is simpler that that of the CNF approach.

3.3.2.3 Logical Formulation of the MINLP

Building on the ideas presented above, Raman and Grossmann [21] have shown that the MINLP problem can be formulated as a generalized disjunctive program. The problem then has the form:

$$\min Z = \sum_i c_i + f(x)$$

$$s.t.\ g(x) \le 0$$

$$\begin{bmatrix} Y_i \\ h_i(x) \le 0 \\ c_i = \gamma_i \end{bmatrix} \vee \begin{bmatrix} \neg Y_i \\ B^i x = 0 \\ c_i = 0 \end{bmatrix} i \in D \tag{3.20}$$

$$\Omega(Y) = \text{True}$$

$$x \in \Re^n, c \ge 0, Y \in \{\text{True}, \text{False}\}^m$$

Here c_i and x are the continuous variables, Y_i represent the Boolean variables, $\Omega(Y)$ represents the logical relations involving the Boolean variables and D represents the set of disjunctions. The objective function and g may be linear or nonlinear.

In process synthesis, Yi = True corresponds to the existence of an operation, thus the equations and constraints for that operation are applied. For $\neg Y_i$ = False the continuous variables relating to that operation are set to zero as well as its cost function. Using the generalized disjunctive program formulation Turkay and Grossmann [22] have developed a logic based version of the OA method and the Benders Decomposition for solving MINLP problems. An excellent description of these methods is given by Turkay and Grossmann [22] and Floudas and Grossmann [23].

3.3.3 Nonconvexities

Previously we have stated that all the above methods assume that the objective functions and constraints meet a convexity condition. Nonconvexities can cause the optimization algorithms to converge to extremely suboptimal solutions or not to converge at all. Since process synthesis problems frequently contain nonconvexities we must look at some techniques for dealing with nonconvex functions.

Nonconvex functions cause difficulties in two ways. First is that the NLP subproblems that are essential to the solution techniques described above, may not have a unique local optimal solution. "Second, the master problem does not guarantee a valid lower bound, Z_L^K or a valid bounding representation with which the global optimum may be cut off."[10] As Grossmann points out [10], in order to deal with these problems we can use global optimization methods that operate on problems with special structures [23] or use a heuristic strategy to reduce the effect of the nonconvexities. Grossmann [10] provides a nice description of the application of heuristic methods for attempting to eliminate the nonconvexities from the master problem.

While these methods can address some MINLP problems with nonconvex functions, this is still an area that continues to provide many interesting research challenges. The interested reader should refer to Floudas and Grossmann [23].

3.4 A Two-Tiered Approach to Process Synthesis

In the previous sections of this chapter we have formalized the batch process synthesis problem through the construction of the State Task Network as well as through the mathematical developments of all the components which make up the formal batch process synthesis problem. We have shown through the mathematical formulation that the batch process synthesis problem is a large complex combinatorial optimization problem and provided a review of some of the most common optimization techniques. In this section we will outline a two-tiered approach to process synthesis which will exploit logic based synthesis in order to reduce the size of the optimization problem and exploit the optimization techniques for the determination of a feasible design.

3.4.1 Logic Based Determination of Feasible Solutions

In Section 3.3.2, we have shown that the process synthesis problem can be formulated as a logic-based MINLP problem. In this section we will describe how logic can be used at an even higher level in order to formulate what is known as a process superstructure of feasible alternatives. Once we have formulated the superstructure we can apply the techniques developed in this chapter to further reduce the search space of alternatives and develop a feasible and possibly close to optimal design.

The Means-Ends Analysis with Nonmonotonic Planning Approach is a logic-based approach to process synthesis. Through the construction of a base case design, a process *superstructure* is created simultaneously, to which optimization routines can be applied to identify the most promising design. The process superstructure that is created through the MEA-NMP methodology is a superstructure of feasible alternatives. A superstructure is a structure that contains all possible interconnections between tasks. From the superstructure all possible configurations could be obtained through selective removal of connections between tasks. The connections represent binary decisions where including the connection is represented by a one and the removal of a connection is represented by a zero.

The identification of the superstructure through the MEA-NMP approach is based on difference identification and a logical grouping of tasks based on properties that relate the tasks. The task groupings, called task categories, allow the maintenance of the set of operations that are alternatives for the specific processing task being performed. Each operation that is identified during the development of the base case design corresponds to a node in an STN representing the process. Correspondingly, each node in the STN can be replaced with a task category of tasks instead of a single task. It is this STN of task categories that becomes the process superstructure. Figure 3.5 provides an example of a generalized superstructure constructed from the all alternative tasks within the task categories. Notice that the general structure of the network in Figure 3.5 corresponds to the structure of an STN, where the states that lie on the arcs between the tasks have been removed. It should be clear from Figure 3.5 that all possible permutations of tasks could be generated from the superstructure.

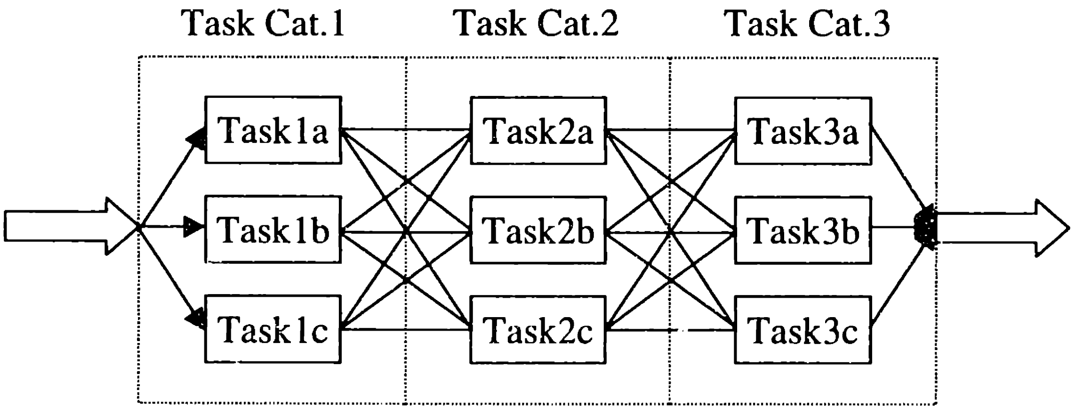


Figure 3.5 Superstructure Construction from Task Categories

So far we have said nothing about the details of the methodology used to select the operations or about the selection of the ordering of operations. These are details of the MEA-NMP methodology and are discussed in Chapter 4.

3.4.2 Optimization: The Selection of the Best Solution

The logic guides the progression of the simultaneous development of both the base case design and the process superstructure and optimization facilitates the determination of the best solution from the superstructure. The optimization routines we have discussed in this chapter provide us with insight into the formulation of the resulting optimization problem from the process superstructure. Even though the MEA-NMP methodology allows us to construct a reduced superstructure when compared to the superstructure of all process alternatives, the resulting optimization problem is still quite large. Fortunately, as we have shown, we can use the formalisms within propositional logic to convert heuristics and constraints into mathematical equivalent forms. This combined with the logic based MINLP optimization routines allow us to further reduce the number of alternatives that need to be examined and hopefully allows us to identify the best solution.

The superstructure of task categories leads naturally to the formulation of a Mixed Integer NonLinear Programming (MINLP) problem. Grossmann and coworkers [8], [24], [22], [15] have extensively investigated the area of superstructure optimization through MINLP. The following is the general formulation of the mixed-integer optimization model [15],

$$\begin{aligned} &\min_{x,y} Z = C(x,y) \\ &s.t. \quad h(x) = 0 \\ &\qquad\ \ g(x,y) \le 0 \\ &\qquad\ \ x \in X \quad y \in \{0,1\}^m \end{aligned} \tag{3.21}$$

where C is a cost function, x is a vector of continuous variables which represent flows, operating conditions, etc., y is a vector of 0-1 variables which denote the potential existence of tasks. The h(x) = 0 equations represent utility and material balances, while

$g(x,y)\leq 0$ represent specifications and physical limits. So we see from this formulation that the superstructure can be formalized as an MINLP.

Of course, as we have mentioned in Chapter 2, a recently emerging body of work is based on formulating the optimization problem as a Mixed-Integer Dynamic Optimization problem based on the inherently dynamic nature of batch processes. The logical construction of the superstructure does not influence the methods used for solving the resulting optimization. Thus, the emerging MIDO problem formulations are also applicable within the two-tiered approach to process synthesis. For a description of the formulation of the MIDO problem the interested reader should refer to the paper by Allgor et al [7].

Once we have completed our discussion of the MEA-NMP methodology in Chapter 4, we will move on and show how the MEA-NMP methodology can be extended and used to create the reduced superstructure of feasible alternatives. Chapter 6 will provide the detailed discussion of how the MEA-NMP approach generates the superstructure of feasible process alternatives. In the same chapter we will also discuss how, once we have the superstructure, we can go ahead and apply optimization techniques to identify the best solution.

3.5 References:

1. Reklaitis, G.V. *Perspectives of Scheduling and Planning of Process Operations.* in *PSE'91*. 1991. Montebello, Canada.
2. Kondili, E., C.C. Pantelides, and R.W.H. Sargent, *A General Algorithm for Short-Term Scheduling of Batch Operations - I. MILP Formulation.* Computers & Chemical Engineering, 1993. **17**(2): p. 211-227.
3. Seymour, C.B., *Conceptual Design of Dedicated Batch Versus Continuous Processing in Multireaction Step Processes*, in *Deparment of Chemical Engineering*. 1995, University of Massachusetts Amherst: Amherst, MA. p. 394.
4. Allgor, R.J., *et al.*, *Optimal Batch Process Development.* Computers & Chemical Engineering, 1996. **20**(6/7): p. 885-896.
5. Sipser, M., *Introduction to the Theory of Computation*. 1997, Boston: PWS Publishing Company. 396.
6. Charalambides, M.S., N. Shah, and C.C. Pantelides. *Optimal Batch Process Synthesis.* in *AIChE Annual Meeting*. 1993. St. Louis, Missouri.
7. Allgor, R.J. and P.I. Barton, *Mixed-Integer Dynamic Optimization I: Problem Formulation.* Computers & Chemical Engineering, 1998.
8. Kocis, G.R. and I.E. Grossmann, *A Modeling and Decomposition Strategy for the MINLP Optimization of Process Flowsheets.* Computers and Chemical Engineering, 1989. **13**: p. 797.
9. Floudas, C.A., *Nonlinear and Mixed-Integer Optimization*. Topics in Chemical Engineering, ed. K.E. Gubbins. 1995, New York: Oxford University Press. 462.
10. Grossmann, I.E., *Mixed-Integer Optimization Techniques for Algorithmic Process Synthesis*, in *Process Synthesis*, J.L. Anderson, Editor. 1996, Academic Press: Boston. p. 172-246.
11. Beale, E.M.L., *Integer Programming*, in *The State of the Art in Numerical Analysis*, D. Jacobs, Editor. 1977, Academic Press: London. p. 409-448.
12. Geoffrion, A.M., *Generalized Benders Decomposition.* Journal of Optimization Theory and Applications, 1972. **10**: p. 237-260.
13. Duran, M.A. and I.E. Grossmann, *An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs.* Mathematical Programming, 1986. **36**(3): p. 307-339.
14. Westerlund, T. and F. Petterson, *A Cutting Plane Method for Solving Convex MINLP Problems*, . 1992, Process Design Laboratory, Abo Akademi.
15. Biegler, L.T., I.E. Grossman, and A.W. Westerberg, *Systematic Methods of Chemical Process Design*. Prentice Hall International Series in the Physical and

Chemical Engineering Sciences. 1997, Upper Saddle River, NJ: Prentice Hall PTR. 796.

16. Quesada, I. and I.E. Grossmann, *An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems.* Computers and Chemical Engineering, 1992. **16**(10/11): p. 937-947

17. Raman, R. and I.E. Grossmann, *Relation Between MILP Modelling and Logical Inference for Chemical Process Synthesis.* Computers & Chemical Engineering, 1991. **15**(2): p. 73-84.

18. Clocksin, W.F. and C.S. Mellish, *Programming in Prolog.* 1981, New York: Springer-Verlag.

19. Post, S. *Reasoning with Incomplete and Uncertain Knowledge as an Integer Linear Program.* in *Proc. Avignon 87: Expert Systems and Their Applications.* 1987. Avignon, France.

20. Raman, R. and I.E. Grossmann, *Symbolic Integration of Logic in Mixed-Integer Linear Programming Techniques for Process Synthesis.* Computers & Chemical Engineering, 1993. **17**(9): p. 909-927.

21. Raman, R. and I.E. Grossmann, *Modelling and Computational Techniques for Logic Based Integer Programming.* Computers & Chemical Engineering, 1994. **18**(7): p. 563-578.

22. Turkay, M. and I.E. Grossmann, *Logic-Based MINLP Algorithms for the Optimal Synthesis of Process Networks.* Computers and Chemical Engineering, 1996. **20**(8): p. 959-978.

23. Floudas, C.A. and I.E. Grossmann. *Algorithmic Approaches to Process Synthesis: Logic and Global Optimization.* in *Fourth International Conference on Foundations of Computer-Aided Process Design.* 1994. Snowmass, Colorado: CACHE-AIChE.

24. Kravanja, Z. and I.E. Grossmann, *PROSYN-An MINLP Process Synthesizer.* Computers and Chemical Engineering, 1990. **14**: p. 1363.

Chapter 4

Means-Ends Analysis and Nonmonotonic Planning in Batch Process Synthesis

As discussed in Chapter 2, the batch process development task is a complex task composed of a variety of components like reaction pathway development, conceptual design, equipment allocation, process optimization and scheduling. While most of these areas have been studied extensively, the area of conceptual design has remained perhaps the only area, that has yet to experience the development of a widely accepted systematic methodology. The goal of this thesis is to develop a methodology for the conceptual design and synthesis of batch processes. Along with the development of the reaction pathway, the conceptual design stage is at the core of batch process development.

We began the development of the batch process synthesis methodology by examining previous attempts at the development of process synthesis methodologies, both continuous and batch. Using the experience of the previous research efforts in this area we have developed a methodology that envelops the best features of these previous attempts.

The methodology that we have developed is called the *Means-Ends Analysis with Nonmonotonic Planning* (MEA-NMP) methodology for process synthesis. The Means-Ends Analysis with Nonmonotonic Planning methodology was derived through the adaptation and combination of methodologies originally proposed for the design of continuous processes. As the name indicates, we have adopted aspects from the Means-Ends Analysis approach, which was first applied to the process synthesis of continuous processes by Jeffery Siirola [1], and the Nonmonotonic Planning concept [2], which has its roots in Artificial Intelligence (AI), to formulate our methodology. Additionally, we have integrated the hierarchical approach to process synthesis promoted by James Douglas [3, 4] for the synthesis of continuous processes. By incorporating the best features of each of these concepts, we have been able to develop a systematic methodology for the synthesis of batch processes.

In this Chapter, we provide a detailed description of how Means-Ends Analysis and Nonmonotonic Planning have been incorporated into a process synthesis methodology. We begin by providing a description of the process synthesis problem in Section 4.1. In Section 4.2 we discuss the Means-Ends Analysis concept. Following in Section 4.3, we discuss the relevant aspects of planning with respect to process synthesis. An important aspect of the process synthesis methodology is the modeling of the operations and the synthesis process, aspects which are discussed in Section 4.4. Once we have completed the discussion of the modeling and process operations, we bring all the concepts together to form the Means-Ends Analysis with Nonmonotonic Planning methodology, which is described in detail in Section 4.5. Finally, we show how the MEA-NMP methodology can be applied to batch process synthesis by means of an example in Section 4.6.

4.1 The Process Synthesis Problem

In Chapter 3, we discussed in detail the formulation of the batch process synthesis problem with regard to the construction of a task network as well as the mathematical formulation. If we examine the details of Chapter 3, we see that we can boil the process synthesis problem down into two general tasks: the construction of a superstructure of alternative processes and the identification of the optimal process from within the superstructure. While this may seem like an over simplified view of the process synthesis problem, it does highlight the two main difficulties: First, how do we go about generating the superstructure of feasible process alternatives and second, how do we select the optimal process from the superstructure through the solution of the associated mixed integer nonlinear programming problem.

One of the first questions we ask ourselves when attacking a problem of this nature is: how do we go about constructing plans without performing a generate and test style search over the entire space of possible alternatives? Heuristic methods have been rather coarse, optimization methods inefficient and general problem solvers very abstract.

The methodology proposed here is a merger of these three methods. A domain specific MEA-NMP planner will provide the primary plan construction through the construction of partial plans and the use of planning islands [5]. This phase will generate the outline of the plan and keep track of possible alternatives. Heuristics provide a method for guiding the planning if the planner gets "stuck". The final phase will be the use of an optimization routine to search through the space defined by the planner and identify the "best" alternative at each node, thus constructing the "best" plan.

4.2 Means-Ends Analysis

Means-Ends Analysis (MEA) can be described in terms of its root word components. In essence MEA is identifying the "means" to bring about a desired "end." The concept of Means-Ends Analysis is fairly simple. Given an initial state of the world and a final state of the world, determine the differences between the two states and identify a task or set of tasks that can derive the final state from the initial state. Here, when we say "state of the world" we imply the world relevant to our processing system. The MEA concept has been attributed to the work of Aristotle around 340 BC [6]. It has been applied to problem solving ever since, but was first formally introduced into the field of Artificial Intelligence by Newell, Shaw and Simon [7] in their paper detailing the development of the *General Problem Solving* program. Simon [8] gives a nice description of MEA within the General Problem Solving paradigm.

MEA works well for systems that can be formulated in terms of objects and operators. Although the work was performed 38 years ago the basic concept has not changed and it is worth presenting an excerpt from Newell et. al. [7]:

> "An operator is something that can be applied to certain objects to produce different objects. The objects can be characterized by the features they possess, and by the differences that can be observed between pairs of objects. Operators may be restricted to apply to only certain kinds of objects; and there may be operators that are applied to several objects as inputs, producing one or more objects as output."

The very simple description given above still applies to the work being performed today. We are still working in an environment that is made up of operators and objects, while the models of the operators and objects have changed. A discussion of the operators used within the MEA environment is given in Section 4.4.

Applying MEA to complex systems, a single operator can rarely transform the initial state into the goal state. Thus, sequences of operators that move the system from the initial state to the goal state need to be constructed. This construction of a sequence of operators is called *synthetic construction.* Figure 4.1 provides a schematic of the synthetic mode of construction using MEA. From the schematic it becomes clear that there may be a number of different plans that can generate the goal state from the initial state.

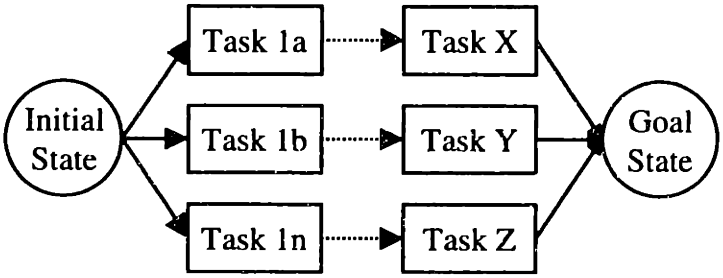


Figure 4.1 MEA - Synthetic Mode

MEA can also be applied in a retro-synthetic manner. Here we begin with the goal state, GS, and identify an operator that contains the action that can generate GS. The *preconditions* of this operator then become the new goal state and the process is continued until the preconditions of an operator match the initial state. The preconditions of an operator are the conditions that the state of the system must meet in order for the operator to be feasible. Figure 4.2 contains a schematic of the *Retro-synthetic* mode of MEA.

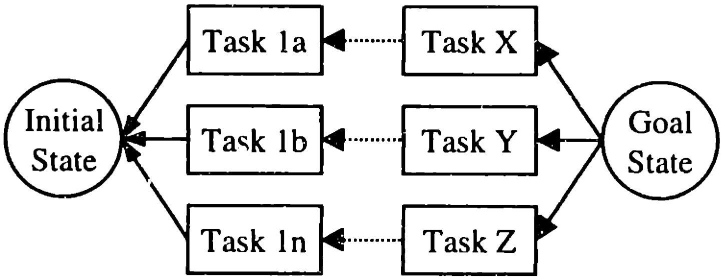


Figure 4.2 MEA - Retro-synthetic Mode

Of course, we can envision a system where we apply both the synthetic mode and the retro-synthetic modes simultaneously for task identification. In this case we are looking for convergence of tasks somewhere in the middle of the plan development. This scenario is depicted in Figure 4.3 and is called the *Hybrid* mode of MEA.

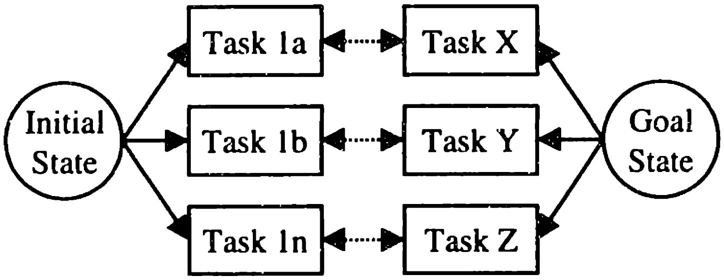


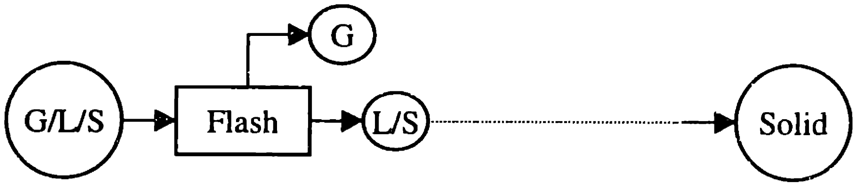
Figure 4.3 MEA - Hybrid Mode

While MEA is a simple and straightforward method for constructing plans, it has an obvious drawback. Since MEA operates in a *linear* or *monotonic* manner, there is no way to guarantee that we are progressing toward either the goal state for the synthetic mode or the initial state when working in the retro-synthetic mode. This means that a planning algorithm that relies completely on MEA may never terminate. In addition to the previous drawback, a direct application of MEA can result in a significant amount of *backtracking*. Backtracking occurs when, after a series of steps the method fails to find an applicable operator. In order to continue with the synthesis of the plan, the planner must back up to the last branching point and select a new direction to continue planning. This has been shown to be an inefficient manner of plan construction [2]. We will see in the following sections how we can overcome the drawbacks of MEA.

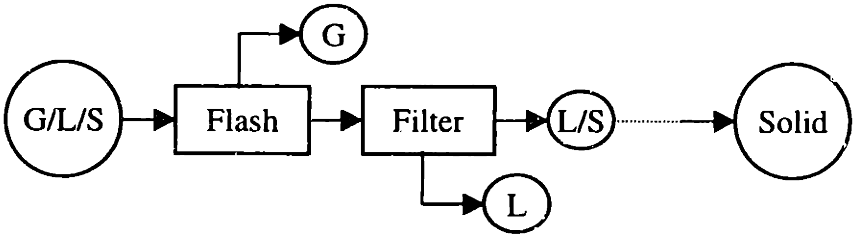
It should be obvious that Means-Ends Analysis is a method for constructing task networks or plans. In the next section we will discuss the two main categories of planning and their advantages and disadvantages. Before we leave this section lets show how the application of MEA to process synthesis would proceed by means of an example.

Illustration 4.1

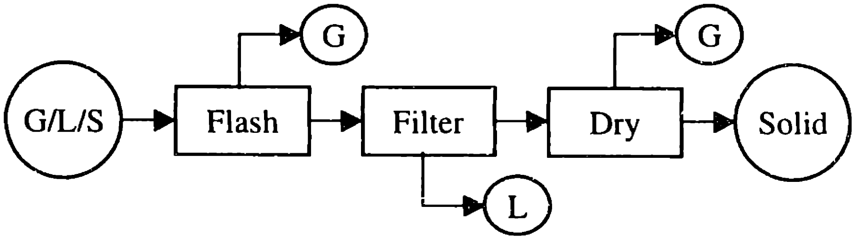
Assume we are given a mixture that contains a vapor phase, a liquid phase and a solid phase and our goal state is to produce the solid at a very high purity. Application of the MEA approach would be as follows. First, we must identify the type of difference between the current state i.e. the given mixture and the goal state. Here the difference is one of phase. Now we must select an operation that can remove this difference. We can first look to our abstract operators and select separate, since we clearly wish to separate the solid from the gas and the liquid. Since we know of no single operator that can overcome this difference we must begin to select operators that move us closer to the desired goal state. Let us first select the *Flash* operation. (For now let us assume that we "know" how to select the operations, how the operations should be selected will be described later in this chapter.) We then apply the *Flash* operation and generate the output state.



Once the new state has been generated we check for differences between the new state and the goal state. Again, we have a phase difference and we select and apply another operator to remove this difference. In this case it is the *Filter* operation.



This is repeated until either we reach an infeasible operation in which case we must backtrack and look for another route or we reach the goal state. In our simple example we show the final operation to produce the desired goal state.



4.3 Planning

During process synthesis, the goal for the developer is to develop a set of operating procedures to produce the desired product at the desired specifications. The developer studies the information provided by the chemist and then attempts to select the processing steps and operating conditions that will perform the required tasks. The developer is creating a plan for the process. What the developer does during the process synthesis activity is plan the ordering of the operations and plan what the operating conditions should be.

The synthesis of batch process designs is a planning operation. Conceptual design is defined as the selection of process operations and the interconnections between these operations to transform a set of raw materials into the final product. Planning is defined as constructing action sequences to achieve a goal. By comparing these two definitions we see that conceptual design is naturally a planning task. In this section we examine how planning is defined and then go on to discuss the two main types of planning, linear planning and nonlinear planning.

4.3.1 The Planning Concept

Winston [9] defines a plan as " a prescription for a sequence of actions that, if followed, will change the relations among objects so as to achieve a desired goal." Thus, planning can be defined as the act of determining the prescription described by Winston. We can expand on the concept of planning by examining three key ideas behind planning as proposed by Russell and Norvig [6]: (1) through planning we can "open up" the representation of states, goals and actions, (2) the planner is free to add actions to the plan wherever they are needed, rather than in an incremental sequence starting at the initial state, and (3) most parts of the world are independent of most other parts. The first key suggests that since planning algorithms generally require the use of a formal language for its descriptions, the planner is able to make "direct connections between states and actions" [6]. The second key implies that the planning can take place based on the order of importance of goals or any other criterion and does not need to proceed in a sequential manner. This has the potential for improving efficiency. Since the planner is not forced to move in a particular sequence, the planner can implement methods to avoid backtracking. The third key allows the planner the freedom to identify subgoals and construct *partial plans* in order to achieve those goals in a so-called divide-and-conquer strategy. The term partial plan will be defined later in this chapter.

Planning has traditionally been a key component within research areas of AI. There have been numerous attempts at creating efficient and effective planning algorithms. A long time goal of AI researchers has been the development of a domain-independent planning algorithm. Chapman [2] was able to develop theorems "that suggest that efficient general purpose planning with more expressive action representations is impossible." This implies that in order to use planning effectively we should focus on the development of domain-dependent planners. This is not to say that the lessons learned and the progress made during the study of the domain-independent planners should be ignored. On the contrary, a lot can be learned about the development of planning algorithms from this previous work. Chapman [2], Lakshmanan [10], Winston [9] and Russell and Norvig [6] all provide reviews of domain-independent planning algorithms and techniques. The interested reader should consult these sources. From this body of work in the field of planning, planning can be grouped into two categories, linear (monotonic) planning and nonlinear (nonmonotonic) planning.

For any planning problem there are three possible outcomes:

(i) Success - where a feasible plan is found.

(ii) Failure - when a complete enumeration of the search space has been performed and no feasible plans have been found.

(iii) Nontermination - the plan continues to grow through the addition of operators but never finds a solution.

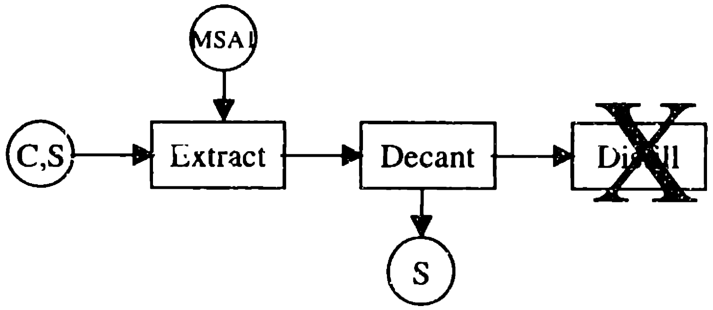
Obviously, nontermination is the worst possible result since we have no way of knowing if the planner is not going to terminate. We may think the planner is involved in a complex problem and requires a large number of operators in order to formulate the solution plan. The only way to handle this situation is to establish a priori some termination criterion to force the termination of the planning operation. Since nontermination is a possible result from any planning algorithm we must be aware of this possibility when developing our planning methodology.

Two of the main categories of planning algorithms are the *linear* or *monotonic planning* algorithms and the *nonlinear* or *nonmonotonic planning* algorithms. The following sections describe both monotonic and nonmonotonic planning algorithms and discuss the advantages of the nonmonotonic planning algorithm in the area of chemical process design. Lakshmanan pp. 35-37 [10] gives a very nice chemical engineering example that illustrates the difference between linear planning and nonlinear planning.

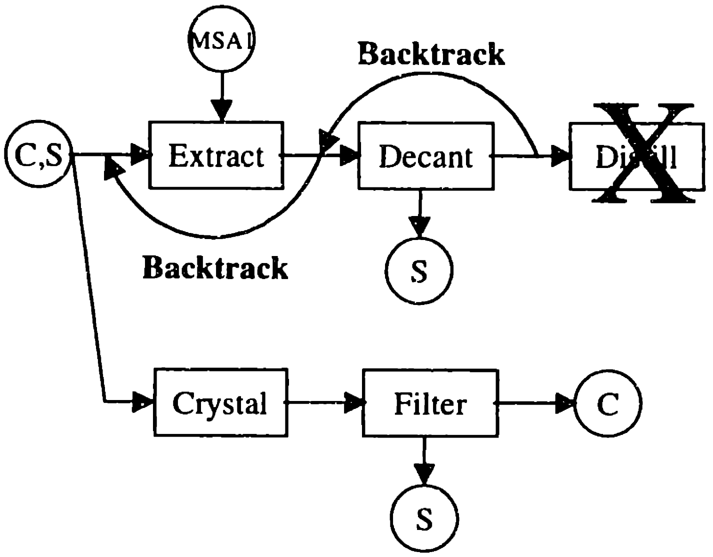
4.3.2 Linear or Monotonic Planning

The simplest method of planning is linear planning. Linear planning is the construction of a plan beginning at either an initial state or at the goal state and applying operators in a step-by-step manner in order to achieve either the goal state or the initial state, respectively. It should be obvious that in most cases the planner will not be able to identify a single operator which can bring the system to the goal state. Thus the plan is constructed in a step-by-step manner, which should move the system closer to the desired goal with each step. At every point in the planning process, all variables are assigned values that correspond to objects in the planning world and the ordering of all the operators that have been selected is fixed. For complex systems it is difficult to quantify exactly how close we are to the goal state, thus we cannot guarantee that we are actually moving toward the goal state. In addition, a major cause of inefficiency for linear planners is that when an operator is found to be infeasible. At this point the system must backtrack to a point in the plan where an alternate decision can be made and the planning can be continued. This repeated backtracking and planning is essentially a trial and error routine. Illustration 4.2 provides an example of backtracking during a linear planning operation.

> *Illustration 4.2*
>
> Given a mixture containing a compound, C, dissolved in a solvent, S, achieve the goal of separating C from S. A linear planner would construct a plan in the following manner. First one could perform an extraction using MSA1 to remove the compound from its original solvent into MSA1. Following the extraction a "decant" operation could be performed to separate the phase containing C from S. Finally we may want to distill the mixture to separate C from MSA1 but we find the distill operation is infeasible.



At this point we must backtrack to a point where a branching opportunity exists and then continue the planning operation.



From the above illustration, we see that due to backtracking, linear planning can be a time consuming process that is quite inefficient. In the worst case scenario, a linear planner would generate all possible structures to the point of infeasibility until a plan is discovered that is feasible. In the following section we will show how a nonlinear planner can avoid the enumeration just described.

4.3.3 Nonlinear or Nonmonotonic Planning

The concept of nonlinear planning is attributed to Sacerdoti [11]. His idea was that while a plan is being constructed, the plan does not have to specify fully the order of execution of its steps. This idea by Sacerdoti has spawned the field of nonlinear planning.

Nonlinear planning is the construction of plans through the refinement of *partial plans*. A partial plan is an incomplete plan which leaves some information unspecified. This unspecified information could be either the order of the operators in the plan i.e. it has been determined that a plan requires an azeotropic separation but the exact location of this operation has yet to be determined within the processing plan, or it could be the identification of a category of operators which can be used to achieve the desired goal instead of a specific operator, i.e. for the same azeotropic separation, it is known that the azeotropic mixture must be separated but it has yet to be determined the precise operations for performing the separation. "A single partial plan actually describes a large number of completions." [2] For different values of the unspecified information, alternative complete plans can be developed. For example, depending upon how the previous operations have modified the state of the current mixture, an azeotropic split using a mass separating agent may be chosen over a pressure distillation giving us two possible completions.

A major advantage of nonlinear planning over linear planning is that the use of nonlinear planning can help to reduce the need for backtracking. Since information can be left unspecified, the method of *constraint posting with backtracking* developed by Chapman [2] can be employed.

4.3.3.1 Constraint Posting Without Backtracking

Chapman [2] first implemented constraint posting without backtracking in his domain-independent nonlinear planning program called TWEAK. Constraint posting without backtracking functions as follows. Instead of backtracking when a constraint violation, called a *clobberer*, is detected; identify an operator, called a *white knight*, that would negate the preconditions leading to the constraint violation. The white knight is applied before the operation that causes the constraint violation, this is called *promotion* of the white knight. This leads to the posting of a new constraint determining the temporal ordering of the operations in the plan i.e. the white knight must be applied before the clobberer. Illustration 4.3 provides a graphical description of the elimination of a precondition violation and the resulting temporal ordering of the white knight with respect to the clobberer.

Illustration 4.3

Let us begin this example by assuming that we have selected an operation to perform a generic task and call it the primary alternative. Before we can apply this operation we must first check the preconditions for the operation to ensure its feasibility. Let us now assume that one of the preconditions for this operation is violated as depicted in Figure 4.4.

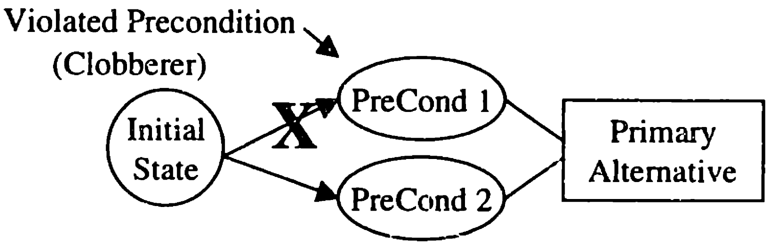


Figure 4.4 Precondition Violation

If we were performing the planning using a linear planner, this operation would be labeled as infeasible and would be removed from consideration, thus eliminating a potential plan. The nonlinear planner, however, attempts to identify an operation that will change the state of the system, removing the condition that lead to the precondition violation. This operation is placed before the primary alternative as shown in Figure 4.5, generating a new state, which now does not cause a precondition violation leading to feasibility of the primary alternative.

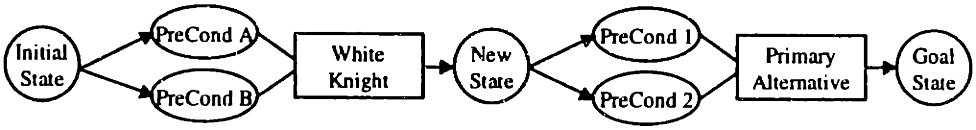


Figure 4.5 Negation of Precondition Violation by the Promotion of a White Knight

We see that we have avoided the inefficient backtracking of linear planning and perhaps more importantly, we have generated a plan that may otherwise have been undiscovered.

Another form of promotion of operators which also results in constraint posting without backtracking is the promotion of an infeasible operation. If we are unable to identify a white knight operation, before we remove an operation from consideration, we attempt to promote the infeasible operation in order to determine if reordering of the operations will remove the precondition violation. The infeasible operation is placed before the preceding operation in the plan and its preconditions are evaluated. In order to use this type of promotion the preconditions of the promoted operation must not be violated or violations must be removed through white knight identification and the resulting postconditions from the application of the promoted operation must not cause the preconditions of the demoted operation to be violated. An example of the promotion of an infeasible operation is given in Illustration 4.4.

Illustration 4.4

We begin this example with a partially developed plan consisting of two operations which have been applied. We have identified the primary alternative operation and we must evaluate the preconditions. This is shown in Figure 4.6.

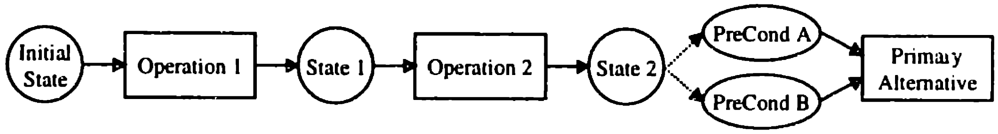


Figure 4.6 Plan with Identification of the Primary Alternative

Upon evaluating the preconditions of the primary alternative operation we find a precondition violation and let us assume that we cannot identify a white knight to resolve the precondition violation. We then promote the primary alternative to be before Operation 2 in the plan as shown in Figure 4.7.

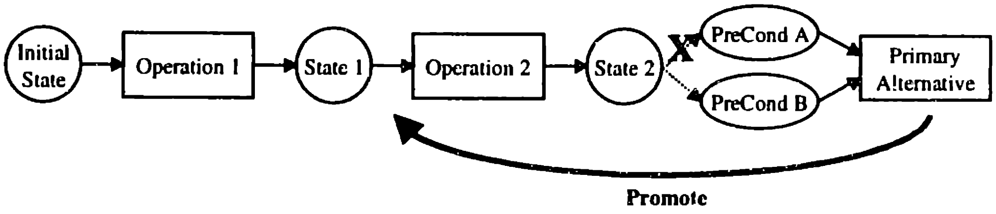


Figure 4.7 Promotion of the Primary Alternative

We must now evaluate the preconditions of the primary alternative operation using State 1 as the current state as shown in Figure 4.8.

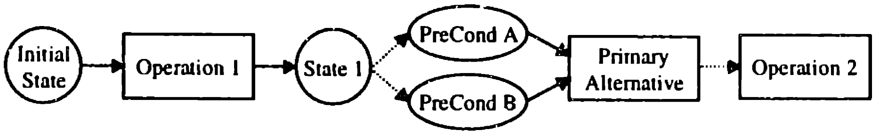


Figure 4.8 Evaluation of the Preconditions

Assuming there are now no longer any precondition violations, we apply the primary alternative operation and generate the new current state which we then use to evaluate the preconditions of Operation 2. What we have done is demote Operation 2 with respect to the primary alternative operation as shown in Figure 4.9.

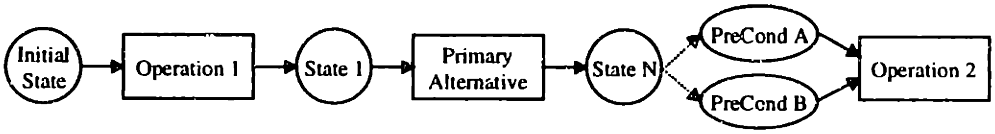


Figure 4.9 Evaluation of Operation 2 Preconditions

If there are no precondition violations for Operation 2 we can apply Operation 2 and continue with the planning as depicted in Figure 4.10.

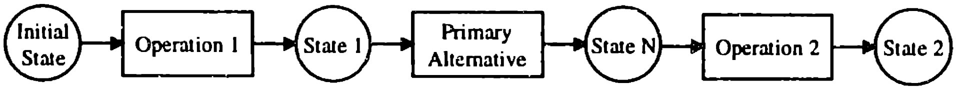


Figure 4.10 Application of Operation 2

4.3.3.2 Planning Islands

Another major advantage of nonlinear planning is that it allows for the identification of *planning islands* [5]. A planning island is an operator or a category of operators that must be included in the plan in order for the plan to be able to reach the goal state. Since nonlinear planning allows for the incomplete specification of the plan, depending upon attributes of the initial state and the goal state, planning islands can be identified a priori. Siirola [5] refers to these attributes as *critical features*. Some examples of critical features include azeotropic mixtures, solid eutectics and temperature sensitive components. The corresponding planning islands are azeotropic separation, solid/solid separation and low temperature separations. From the examples of the planning islands, it should be clear that planning islands are also examples of partial plans since they are specified with incomplete information. The use of planning islands will increase the efficiency of planning since any plan that does not contain the island can be discarded as infeasible. The planning islands act to reduce the search space of feasible alternatives while also allowing us to maintain a set of predefined strategies for overcoming the critical features. For example, to overcome distillation boundaries one could perform a) an extraction, b) crystallization, c) add a mass separating agent or d) change the system pressure.

Now that we have discussed the main modes of planning, before we can completely describe the Means-Ends Analysis methodology we need to discuss the modeling and the operators used by the planning algorithms. The modeling and the operators strongly influence the quality of the plans that are developed, as will be discussed in the following section.

4.4 Modeling and Operators

At the heart of domain-dependent planning are the operators and the models upon which they are based. The modeling performed by a planning program is composed of modeling of the state of the system and modeling of the planning operators along with their effect on the system. Lakshmanan and Stephanopoulos [12] provided a list of four modeling tasks that a computer program must have for automatically generating operating procedures. This list has been adapted to the generation of conceptual process designs.

1. Model the state of the system – A planning program must reason about the initial, intermediate and final states of a plan. The first basic modeling requirement is the representation of the state of the system for which planning is being carried out. This was discussed in Chapter 3.
2. Model the operators - A plan is implemented by carrying out a series of actions (such as distill, decant, crystallize) upon objects (materials) in the system. The models for carrying out the actions are known as operators. A planning algorithm must contain models of these operators.
3. Model the effects of operator application - Actions, which are carried out by the operators result in a perturbation of the state of a system, necessitating special forms of models to represent these effects.
4. Model the planning methodology - Finally, all of the information contained in the facilities above must be manipulated by some inferencing machinery which is responsible for constructing the sequences of operations which define a plan.

The modeling of the operators and the effects of the operators are interrelated with the modeling of the state of the system. The modeling of the planning methodology is the integration and coordination of the previous three modeling tasks in order to form a complete planning scheme. The four modeling tasks form the basis of the development of the MEA-NMP planning methodology. The modeling of the state of the system was discussed in Chapter 3. The remaining tasks are discussed in this chapter.

The development of a planning methodology for process synthesis depends heavily upon the ability to identify the appropriate operator models. The plan will be constructed using the operators and it is the models of the operators that provide the necessary information to the planner in order to perform the planning operation. The operator models in the planning paradigm are the "steps" that move the system from one "situation"(state) to the next. Two major criteria for determining the appropriate operator models are (i) correctness/completeness of the models and (ii) the tractability/efficiency of the planning methodology [12]. The two criteria imply that the complexity of the planning methodology is directly related to the type of operator models employed.

Below we summarize the three main types of operator models that have been traditionally used in planning; the STRIPS operator, the *conditional* operator and the *functional* operator. It will be shown that the functional operator is most applicable to the domain-dependent planning of batch processes.

4.4.1 STRIPS Operator

The simplest operator and the one used by most domain-independent planners is the STRIPS operator [13]. The STRIPS operator was a key element of the STRIPS (Stanford Research Institute Problem Solver) problem solving project. The STRIPS project was an attempt at creating a program "that attempts to find a sequence of operators in a space of world models to transform a given initial world model into a model in which a given goal formula can be proven true." While the STRIPS project failed to fully achieve its goal, one element from the STRIPS project did make an impact on the field of planning and that is the STRIPS operator.

The STRIPS operator is a description of an action, consisting of a finite set of *preconditions* and a finite set of *postconditions*. The preconditions are statements that must be true of the world before the action can be performed and postconditions are statements that are guaranteed to be true after the action corresponding to that operator has been carried out. There are three restrictions placed on the form of the preconditions and the postconditions:

1. Preconditions and postconditions must be expressed as propositions that have a content, which is a tuple of elements, and can be negated.
2. The elements can be variables or constants, and there can be infinitely many of these.
3. Functions, propositional operators and quantifiers are not allowed.

A schematic representation of the STRIPS operator is given in Figure 4.11.

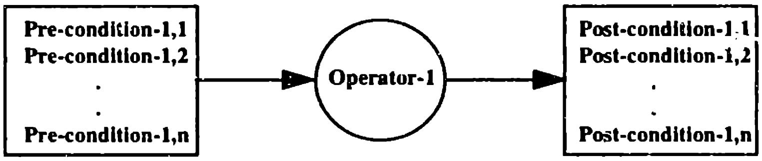


Figure 4.11 Schematic of the STRIPS operator [12].

The major advantage of the STRIPS operator is its simplicity. If all the actions of the system can be expressed as independent, non-interacting operators then the planning problem becomes a problem of determining a consistent ordering of the operators. The correctness and completeness of domain-independent planners have been proven by Chapman [2], thus planning using STRIPS operators can be proven to be correct and complete. Chapman also provides a polynomial time algorithm for planning using the STRIPS operator showing that planning using STRIPS operators proceeds in polynomial time.

The major disadvantage of the STRIPS operator model is that the behavior of most systems is functionally dependent on the situation in which the action is carried out. This violates the third fundamental condition of the STRIPS operator. Also, there is no way to model the interaction between operators using the STRIPS operator. Many practical applications require interaction between the operators since the actions of one operator many influence the actions of other operators in the system. It should be clear from the above description that for a process modeling system, the STRIPS operator is insufficient. The main problem is a necessity to violate restriction number three. For example, modeling a distillation operation would require an infinite number of STRIPS operators since we would need to generate an operator to match every possible state of the feed stream to the resulting output steams. Clearly the postconditions of a distillation are functionally dependent on the input state as well as the operating conditions of the column.

4.4.2 Conditional Operator

The next major class of operators is the conditional operator. The conditional operator has two sets of postconditions, the *if-true postconditions* and *the if-false postconditions*. If all of the preconditions of the operator are satisfied, then the if-true postconditions represent the state of the system. If any one of the preconditions is not satisfied then the if-false postconditions represent the state of the world. A schematic of the conditional operator is given in Figure 4.12. The use of conditional operators in planning was examined by Chapman [2]. From this work Chapman developed the Intractability Theorem:

> *Intractability Theorem* – The problem of determining whether a proposition is necessarily true in a nonlinear plan whose action representation is sufficiently strong to represent conditional actions, dependency of effects on input situations, or derived side-effects is NP-hard.

The proof of the Intractability Theorem can be found in [2]. From the Intractability Theorem it is clear that no polynomial time algorithm is known for checking whether a proposition is necessarily true in a given situation.

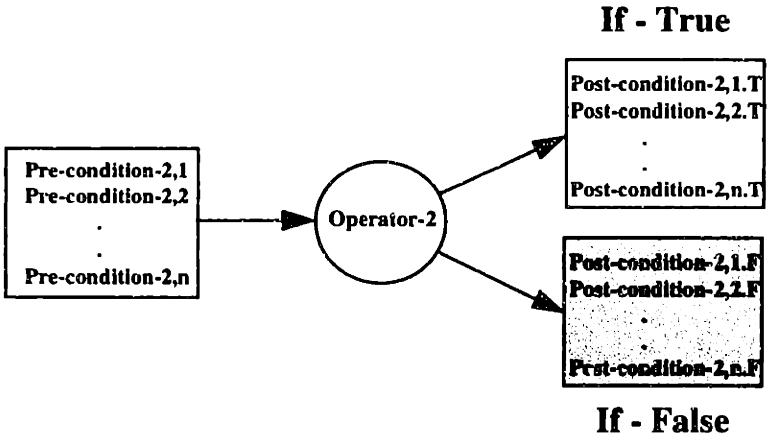


Figure 4.12 Schematic of the conditional operator [12].

As with the STRIPS operator, the conditional operator is also insufficient for use in detailed process modeling. While it may find some applicability for use in describing abstract operators, its simple binary comparison cannot capture the effect of operations such as distillation. It should be clear that while the above operators are relatively simple, it is that simplicity that severely limits their applicability. It is the final operator, the functional operator, which will provide us with the necessary level of complexity to perform the planning of batch chemical processes.

4.4.3 Functional Operator

In order to extend the area of planning to solve a practical problem, Lakshmanan and Stephanopoulos [12] found it necessary to develop a new form of operator. They called this operator the *functional* operator. The functional operator removes the third restriction imposed upon the form of the propositions of the STRIPS operator model by allowing the postconditions of the functional operator to be functionally dependent on the preconditions. This operator is formulated by constructing a set of conjunctions of preconditions where each element in the set describes some possible situation that might exist before the operator is applied. For each element of the set of preconditions, there is a corresponding set of postconditions which specifies what facts are to be asserted in the world if the action is carried out with the corresponding preconditions being true before operator application [12}. This does not imply that a set of all possible preconditions is maintained for the functional operator. A set of preconditions that determine the feasibility of the operations is maintained and any state that satisfies this set of preconditions is passed on to the operator model which then generates the postconditions based on the initial state and the operating conditions.

> *Illustration 4.5*
>
> One of the preconditions for a standard distillation operation is that the feed be a single liquid phase. If the feed stream is a single liquid phase then it is passed on to the model of the distillation task and the output streams are generated. It is these output streams that define the postconditions of the distillation operation.

A schematic of the functional operator is given in Figure 4.13.

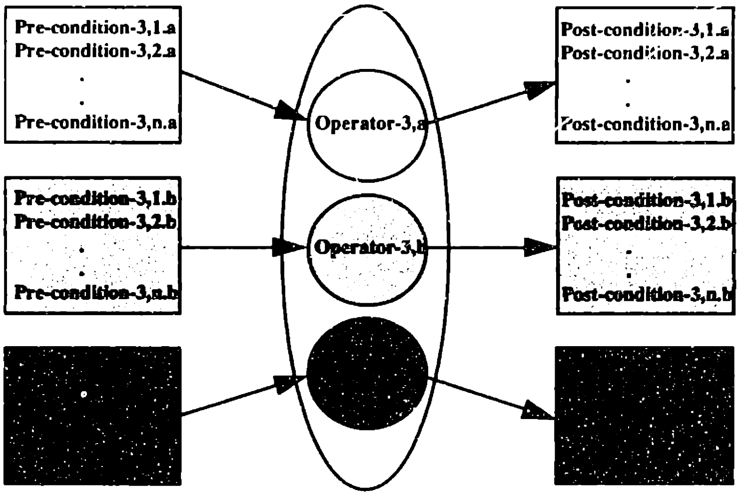


Figure 4.13 Schematic of the functional operator. [12]

Alternatively, the preconditions and postconditions of the functional operator can be expressed in terms of intervals rather than as single-valued conditions. For example, one of the preconditions of the membrane operation states that the diameter of the solute molecule must be greater than the diameter of a membrane pore. Thus, the membrane operation is applicable over a range of values. The advantage of specifying ranges of value for the preconditions and postconditions is that we can then develop algorithms to quickly screen alternative operations by matching the range values for the postconditions of an operation to the range of values for the preconditions of the subsequent operation.

If the ranges overlap then the subsequent operation is applicable. Additionally, the amount of overlap also gives us the range of applicability for the subsequent operation. If there is no overlap the subsequent operation is not applicable.

The relaxation of the non-functional restriction implies the applicability of the Second Intractability Theorem.

> *Second Intractability Theorem* - The problem of determining whether a proposition is necessarily true in a nonlinear plan whose action representation employs functional operators is NP-hard. The proof is given by Lakshmanan [12].

How does the Second Intractability Theorem effect the concept of planning for process synthesis? In the worst case, a planning algorithm using functional operators may never terminate. It may continue to apply operations in order to achieve various subgoals but since the postconditions are generated dynamically based on the input state, there is no guarantee that the postconditional state does not have attributes that actually make it more difficult to achieve the desired goal state. This implies that we must be aware of this situation and impose termination constraints on the problem. Chapman [2] has also provides us with some hope that the use of functional operators will be fruitful: "I have examined a number of specific domains, and found that for each of them it was easy to find an efficient truth criterion, but that these criteria were quite different." Since the planning of a batch process is a domain specific application of planning, we may be able to formulate the problem as Chapman suggests in such a way that we avoid making the problem NP-hard, although we have not proven this to be the case.

Now that we have discussed the key elements behind the development of the MEA-NMP methodology, let us proceed to show how these elements have been brought together to form a batch process synthesis methodology.

4.5 The MEA-NMP Approach

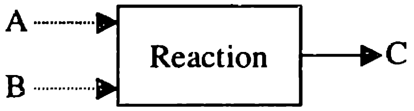
The approach that has been developed for the construction of process flowsheets is called the Means-Ends Analysis with Nonmonotonic Planning Approach. This approach to planning batch processes allows for the systematic and simultaneous selection and ordering of processing operations. The approach consists of three main concepts extracted from the field of planning. First, is the means-ends analysis used for difference and task identification. Second, is the nonmonotonic planning used to increase the efficiency of the planning by allowing the construction of partial plans. Third, is constraint posting without backtracking which allows for the identification of tasks to remove constraint violations and determines the relative ordering of these tasks. The previous sections have explained each one of these concepts independently. This section shows how they have been combined to form the planning methodology used for the construction of batch processes. Before the detailed description of the MEA-NMP approach is given let us go over a simple example showing how MEA and NMP can be integrated to achieve the desired result.

Illustration 4.6

Let's assume that we have the following reaction:

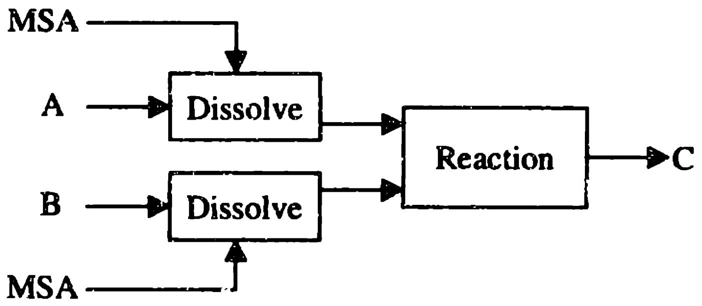
$$A + B \rightarrow C$$

where A and B are the raw materials and are available as solids. Our goal is to produce compound C. We first apply MEA to identify differences. In our example the difference is quite obvious, the molecular structure of A and B differ from the molecular structure of C. The next step is to identify an operation that can resolve this difference. In this case the task is chemical reaction:



From the figure it would appear that we are finished but before we can apply the operation we must first check the preconditions associated with the operation. Here, in order to have a reaction the reactants must come into close contact and since our reactants are both solids this is not possible. At this point a direct application of MEA would suggest marking the reaction as infeasible. MEA would find no solution. This is where the NMP helps us. Through the application of the NMP, we try to identify operations (white knights) to eliminate the precondition violations before we attempt to apply the react operation. The procedure for identifying the white knights is given in detail below. For this example, the white knight which removes the close contact violation is the dissolve operation. Since we are eliminating a precondition violation, the temporal ordering generated, places the white knight before the reaction operation.

> Thus we place the dissolve operation before the reaction. Assuming that the preconditions of the dissolve operations have been met, we can go ahead and apply the white knights, generate the new material states and then apply the reaction operation.



We see that NMP has made the elimination of the precondition violation possible and made the reaction, which MEA suggested was infeasible, feasible.

Now that we have demonstrated the behavior of the MEA-NMP methodology, let us revisit the modeling of operators to show how they have been incorporated into and impact the MEA-NMP methodology.

4.5.1 MEA-NMP Operators

The operators used in the MEA-NMP planning methodology are functional operators that were constructed using batch process operation models contained in *Batch Design-Kit*. Batch Design-Kit (BDK) is a batch process construction tool which uses simple models to provide a general simulation of the process. An overview of BDK is provided in Chapter 7. The MEA-NMP operators correspond to the various batch processing operations such as batch-distill, crystallize and extract. Since BDK does not perform any planning tasks, the operations contained within BDK do not have preconditions associated with them. These operations have been extended by coupling them with preconditions to make them usable within the MEA-NMP planning methodology. The operations have the same general construct as the functional operator shown above. Within the batch process operator, the sets of preconditions for the functional operator are replaced with a single set of preconditions that determine the applicability of the operator with respect to the current input state. The current operation models used by the MEA-NMP methodology are the simple batch process operation models contained within BDK. The postconditions are generated dynamically as the output of the model acting upon the input state. Thus, the functional operator is the most appropriate model for the operations used in the MEA-NMP methodology.

> *Illustration 4.7*
>
> Let us look at the construction of the batch-distill operator with the MEA-NMP framework. The model contained within BDK is the Rayleigh Distillation [14]. A set of preconditions for a simple batch distillation operation are:
>
> 1. No gas phase in the feed.
> 2. Single liquid phase feed.
> 3. No solid phase in the feed.
> 4. The keys must not form an azeotrope with any component.
> 5. The concentration of any dissolved solids must be less than the saturation concentration of the major liquid component of the bottoms product.
> 6. The decomposition temperature of all components must be greater than the highest boiling point of all components.
>
> The actual postconditions are generated by applying the model to the input state but we can capture a few generic postconditions which can be used to determine which task categories should contain this operation. An example of a generic postcondition for batch distillation is: multicomponent liquid separation. Compare this illustration to the operator model given in Figure 4.13 and see that the batch-distill operator is a functional operator.

An important point here is that the quality of the resulting plan is directly dependant upon the level of detail contained within the operator models. The more detailed the model, the more accurate the planning methodology will be at including the required operations. For example, if we use models that generate perfect splits for the separation operations, then the plan will only include the main separation operations. Using more detailed models will force the planning algorithm to generate plans with more complex sequences of operations. A nice feature of the planning methodology is that different sets of operator models can be used depending on the type of design being generated. In the early stages of process development, we may want to quickly screen many alternatives to see if any are potentially profitable. Using a set of operators that have simple models may be sufficient at this point. Later, when it has been established that the process could be profitable, it is possible to go back and perform the planning using a more detailed set of models. The more detailed models will give us a more accurate description of the process.

Illustration 4.8

Assume we have a four component mixture and we would like to isolate component B. If we apply operations that perform perfect splits, the plan generated will have the following format:

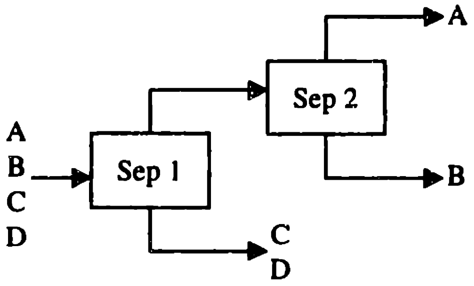


Figure 4.14 Isolation of B using perfect split operators.

If we then perform the same process using more detailed operators that give a more accurate description of the postconditions, the planning methodology generates a plan with the following form:

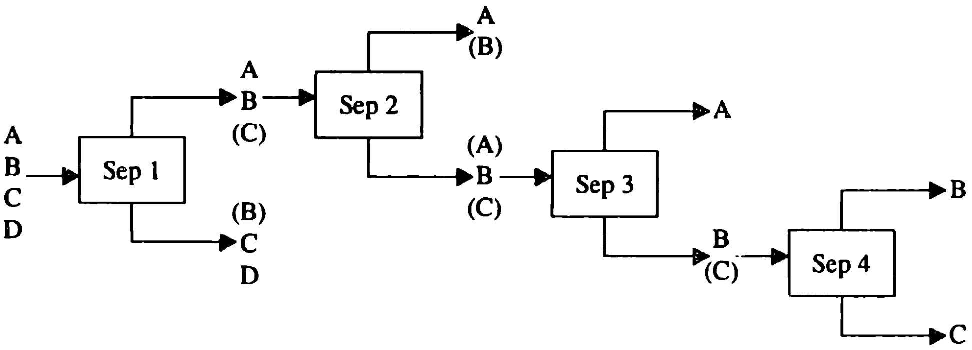


Figure 4.15 Isolation of B using more detailed operator models.

Clearly, the use of more detailed models provides a more accurate plan.

It is clear from the above illustration that using more detailed operators has the potential for increasing the complexity of the plan as well as gives a more accurate plan.

It should be noted that since we used general separation operators in the illustration we do not capture the added complexity of additional operators that would be added to remove precondition violations. Thus, an actual plan would be even more complex than the one contained in Illustration 4.8.

Once we have an understanding of how the operation models influence the process synthesis, we are finally ready to explain the details of the MEA-NMP methodology for the synthesis of batch processing schemes.

4.5.2 Details of the MEA-NMP Approach

The MEA-NMP methodology functions using six main steps. These steps are applied iteratively and recursively in order to develop the processing scheme. In this section we will describe each of these steps in detail and show how, together, they form the MEA-NMP planning methodology. The six main steps are:

1. Identify the differences between the current state and the desired goal state.
2. Screen the current state and the differences identified in Step 1 for critical features.
3. Apply MEA for task category and primary operation identification.
4. Evaluate the preconditions of the proposed operation.
    a) Identify white knights to remove precondition violations.
    b) Promote operator if white knight cannot be found.
5. Apply the operators and generate the new current state.
6. Check the plan for abstract operators.

A schematic of the MEA-NMP methodology is given in Figure 4.16. Notice that the flow diagram contains the six main steps listed above. At the top, left of the diagram we see that the *Current State* and the *Goal State* are both inputs to the *Identification of Differences* block. From there we *Screen for Critical Features.* If any are found we identify and *Insert Planning Islands* into the plan. After we have screened for critical features we begin the application of the MEA to identify the *State Category*, the *Task*

Category and the *Primary Alternative* operation. Once the Primary Alternative has been selected we *Evaluate its Preconditions* to ensure feasibility. If any precondition violations are found we the attempt to use the *Identify White Knight* operations in order to eliminate the precondition violations. If no white knight can be found to eliminate the precondition violation, we then *Promote* the operation and re-evaluate its preconditions. If the preconditions are not violated, we apply the operation and evaluate the preconditions of the next operation. If promoting the operation does not remove the precondition violations, the Primary Alternative is removed from consideration and a new Primary Alternative is selected. If a white knight operation is identified, the white knight operation is placed before the Primary Alternative in the plan. We must then check the preconditions of the white knight against the current state to ensure it is feasible. If the white knight operation has any precondition violations, we can go ahead and attempt to identify a white knight operation to remove this precondition violation. If another white knight is identified it is placed before the previous white knight. This continues until there are no more precondition violations. Once we have found an operation that has no precondition violations we can apply that operation to the current state and generate a new current state. This new current state is compared to the preconditions of the next operation in the plan. If no violations are found then next operation can be applied. This continues until there are no more operations to be applied. At this point we loop back to the top of the methodology and compare the new current state to the goal state to see if there are any remaining differences. If so, we continue as described above, if not then we must check the plan for abstract operators. If there are no abstract operators then we have complete the plan, if there are abstract operators we must further refine these operators to determine the detailed operations to replace the abstract operators. This provides a overview of the MEA-NMP methodology depicted in Figure 4.16. Now we will go back and describe each of the six main steps in detail.

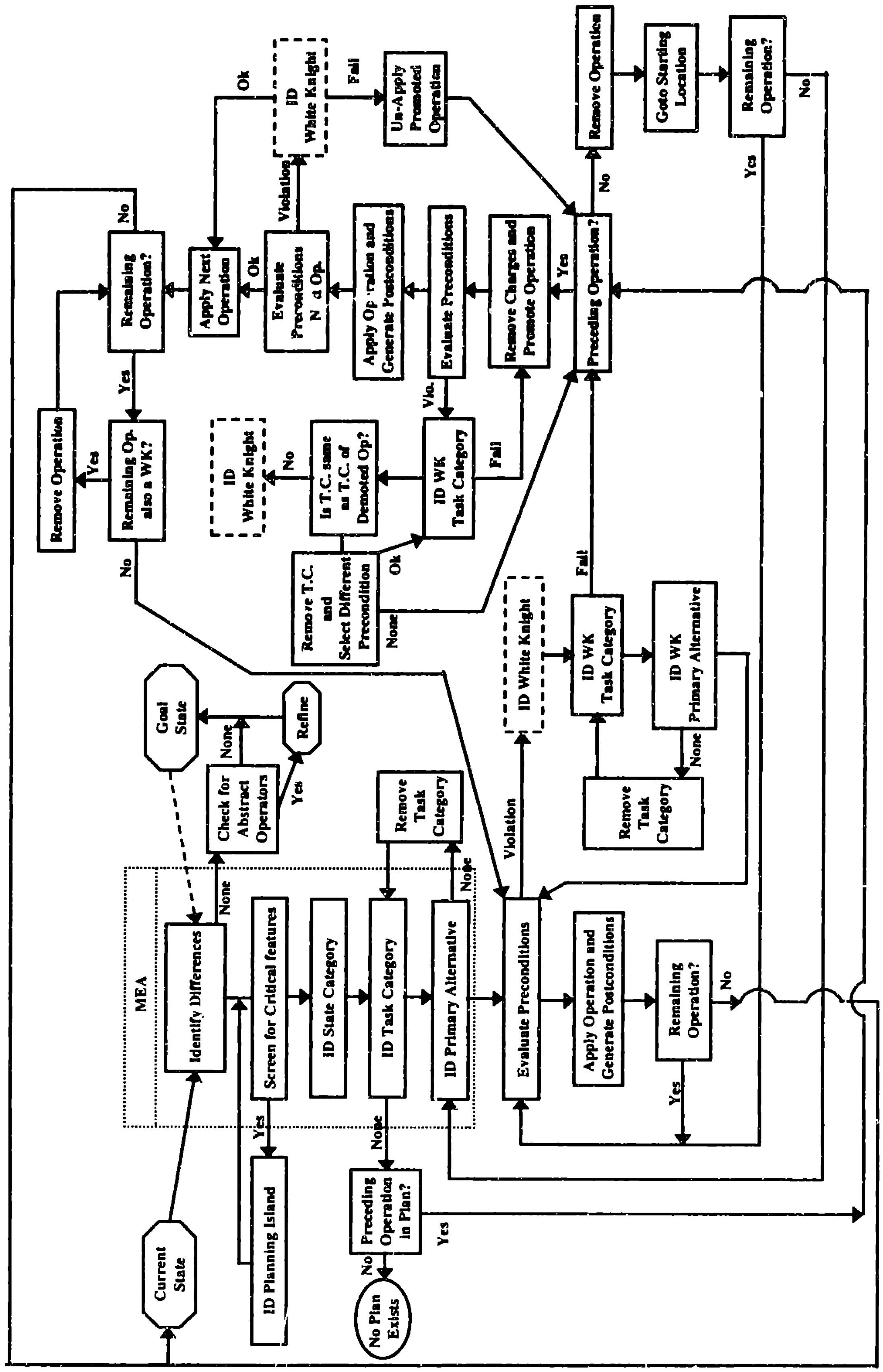


Figure 4.16 Schematic of the MEA-NMP Methodology.

4.5.2.1 Step 1 - Identification of the Differences Between the Current State and the Desired Goal State.

The identification of differences between the current state and the goal state is based upon a comparison of the properties of the current state and the properties of the desired goal state. This corresponds to the Identification of Differences block in Figure 4.16. The identification of differences takes as input the current state and the goal state as depicted in Figure 4.16. Siirola [5] proposed a hierarchical structure for six types of property differences. The order of the hierarchy is *identity*, *amount*, *concentration*, *phase*, *temperature* and *pressure* and *form*. In the words of Siirola [5]:

> *"The hierarchy arises because values of properties higher in the sequence set the behavior of lower properties, and methods which change properties lower in the sequence are less likely to also alter properties higher in the sequence. Properties lower in the hierarchy are usually more readily manipulated in order to satisfy the preconditions for the application of difference elimination methods for properties higher in the hierarchy."*

For example, from the chemical reaction specifications the raw materials, desired product and the by-products are identified. The first difference to be identified from this information is an identity difference; none of the raw material compounds is the main product compound. Step 1 of the MEA-NMP methodology concludes with all the property differences between the current state and the goal state being identified. Step 3 in the methodology will exploit the difference hierarchy.

4.5.2.2 Step 2 - Screen the Current State and the Differences Identified in Step 1 for Critical Features.

The next step is to identify any critical features that may be present in either the current state or within the differences detected in Step 1. Critical features are attributes of the system that can be identified a priori, which allow the identification of special operators called *planning islands* that act upon these features. The planning island concept was introduced to process synthesis by Siirola [5]. Siirola [5] describes critical features as "looking ahead for potential difficulties and develop contingencies early in the synthesis process to deal with them before running into dead ends." Planning islands are the methods for dealing with the critical features. Siirola suggests that the system be examined for critical features and the planning islands become the "strategic methods for crossing, breaking, by-passing, reaching or exploiting the critical feature."[5] A planning island is an operation that must be included in the plan and can be identified without specifying a complete plan. At the time of identification it may not be known how the island will be connected to the rest of the plan. The connections between the island and the rest of the plan will evolve as the rest of the plan is constructed. Identifying the planning islands and constraining the plan to contain the islands increases the efficiency of the planning by reducing the number of possible plans. By identifying these planning islands before applying the general planning algorithm, we can begin the process with a restricted search space. Normally, constraints are not desirable but, in this case, these constraints have the potential for dramatically decreasing the planning search space by identifying sections of the plan that must be included in the final plan. An example of a planning island would be the a priori identification of solvents that form azeotropes. If these solvents are to come into contact with each other within the process then, in order to recover and recycle these solvents, an azeotropic separation must be included in the plan. The azeotropic separation is the planning island and in order to achieve the goal the final plan must contain this azeotropic planning island. Thus we have eliminated from investigation any plans which do not contain this island.

Screening of critical features corresponds to the *Screen for Critical Features* block in Figure 4.16. Notice that if a critical feature is identified, the methodology moves to the *Insert Planning Island* block. Within this block, the appropriate planning island is identified which corresponds to the attributes of the critical feature. Once a planning island has been identified the methodology loops back to check for any additional critical features. This process is repeated until no additional critical features can be identified.

4.5.2.3 Step 3 - Apply Means-Ends Analysis for Task Category and Primary Operation Identification.

Once the differences and the planning islands have been identified, the next step is to apply MEA to identify an operation that will transform the current state to the goal state. If it is not possible to achieve this transformation using a single operator, an operator is selected that creates a new current state that is "closer" to the goal state.

Moving "closer" to the goal state is defined as achieving a state with attributes that more closely match that of the goal state.

How are operations selected in order to move the system closer to the goal state? The MEA has been subdivided into three steps as shown in Figure 4.16. First, the type of difference to be removed is identified. This step is easily performed since the type of difference to be removed is simply the type of difference that is highest in the above hierarchy. This selection of the type of difference to be resolved is called selection of the *state category.* Each state category contains a number of different *task categories.* Once the state category has been selected the next step is to identify the *task category.* A task category is a list of operations, all of which have a common postcondition. Figure 4.17 shows the hierarchy of state categories and the task categories that are contained within each state category.

State Categories →

Task Categories ↓

Identity	Amount	Concentration	Phase	Temperature	Pressure	Form
Molecular Structure Transformation Material Transfer	Mix Split	Phase Split Vapor/Liquid Liquid/Liquid Liquid/Solid Vapor/Solid Solid/Solid Concentration Gas Sep Multicomp Liquid Sep Solubility Reduction Solvent Replacement Solubility Increase Azeotrope Split	Vapor/Liquid phase change Solid/Liquid phase change Liquid/Solid phase change Liquid/Vapor phase change	Temperature Change	Pressure Change	Crystal Precipitate Liquid Vapor

Figure 4.17 State Categories and Task Categories.

The task categories are constructed by identifying an attribute common among various tasks or operations. The tasks contained within one category are not exclusive to that category. This allows the construction of a wide variety of categories depending on the goal that is to be achieved. Table 4.1 contains the current set of task categories with their member operations.

Phase Split SolidLiquidPhaseSplit LiquidLiquidPhaseSplit GasLiquidPhaseSplit GasSolidPhaseSplit	GasLiquidPhaseSplit Flash	LiquidLiquidPhaseSplit Decant
SolidLiquidPhaseSplit Filter Centrifuge Settle Evaporate	GasSolidPhaseSplit Purge Evacuate	VaportoLiquidPhaseChange Cool Pressurize Absorption
VaportoSolidPhaseChange Pressurize Cool	SolidSolid Sep Leach Recrystallize-(dissolve, crystallize) Dissolve	LiquidtoSolidPhaseChange Crystallize Precipitation Membrane Evaporation Cool
LiquidtoVaporPhaseChange BatchDistill DistillCont Flash Evaporation Pervaporation Stripping	SolidtoLiquidPhaseChange Dissolve Heat Melt Leach	SolidtoVaporPhaseChange Heat Sublime
VaporVaporSep Membrane Absorption Adsorption Condense Cool Pressurize	Multicomponent Liquid Sep Batch distill Continuous distill Azeotrope Split Extractive distill Extract Strip Ion exchange Membrane Adsorption Chromatography	Solubility Reduction Solvent replacement Concentrate Cool Charge MSA

Solubility Increase Heat Solvent Replacement Dilute	Material Transfer Charge Transfer Mix	Solvent Replacement Concentrate and Mix Extract Precipitate and Mix
MolecularStructure Transformation React	Pressure Change Pressurize Evacuate	Temperature Change Heat Cool
Azeotrope Split Azeotropic Distillation Extraction Membrane	Liquid from Solid Dry Wash-solid	Crystallize Cool Condense Membrane Add MSA
Dilute Mix	Concentrate Concentrate Evaporate Batch-Distill	Control Exotherm Cool Dilute
DissolvedMaterialRemoval Extract Crystallize Membrane Adsorption Ion Exchange	Manipulate SLE Pressure Change Temperature Change Material Transfer Solvent Replacement	

Table 4.1 Task Categories.

Selection of the Task Category

The task category is selected based on the postcondition that is to be achieved. For example, if the difference to be removed is a form difference and the product is to be isolated as a crystal, then the task category that would be selected is the crystal category. A rule-based approach has been developed in order to select the task category to identify an operation to remove the current difference. The rules are categorized based on type of difference to be resolved, but any rules that are applicable in a given situation can be applied. Table 4.2 contains the current list of rules that have been developed.

Task Category Selection Rules

Identity Difference Rules

- If the material ≠ a reactant or solvent then → MolecularStructureTransformation
- If the material = a reactant or solvent then → MaterialTransfer

Concentration Difference Rules

- If the product spec. material is solid and the current state of the product material is solid and the current state of other materials is liquid and $V_{Liquid} > 0.1*V_{Solid}$ then → SolidLiquidPhaseSplit
- If the product spec. material is solid and the current state of the product material is solid and the current state of other materials is liquid and $V_{Liquid} \leq 0.1*V_{Solid}$ then → LiquidfromSolid
- If the product spec. material is liquid and the current state of the product material is liquid and the current state of other materials is solid and $V_{Liquid} > 0.1*V_{Solid}$ then → SolidLiquidPhaseSplit
- If the product spec. material is liquid and the current state of the product material is liquid and the current state of other materials is solid and $V_{Liquid} \leq 0.1*V_{Solid}$ then → LiquidfromSolid
- If the product spec. is liquid and the current state is liquid then → MultiComponentLiquidSep
- If the product spec. contains material that is not a reaction product and the concentration of the material that is not a reaction product < the concentration of the material that is not a reaction product in the current state then → MaterialTransfer
- If the product spec. contains material dissolved in a solvent and the solvent is not present in the current state and a solvent is present in the current state then → SolventReplacement
- If the current state contains a product mixed with a solvent and the current state contains the solvent and the concentration of the product in the current state < the concentration of the product given by the product spec. then → Concentrate
- If the current state contains a product dissolved in a solvent and the concentration of other dissolved impurities > the concentration of the impurities in the product specification then → DissolvedMaterialRemoval
- If the product spec. is a solid product and the current state is solid and the concentration of impurities in the current state > the concentration of impurities in the product spec. then → SolidSolidSep.
- If the current state is vapor and the product spec is vapor and the concentration of nonproduct material in the current state is > the concentration of nonproduct material in the product spec. then → VaporVaporSep
- If the current state contains two liquid phases and the product spec. is a single liquid phase and the product materials are contained in a single phase in the current state then → LiquidLiquidPhaseSplit
- If the current state consists of a solid phase and a vapor phase and the product is concentrated in one of the phases then → GasSolidPhaseSplit
- If the current state consists of a vapor phase and a liquid phase and the product is concentrated in one of the phases then → VaporLiquidPhaseSplit

Phase Difference Rules

- If the product spec. material is solid and the current state of product material is liquid then → LiquidtoSolidPhaseChange
- If the product spec. material is vapor and the current state of product material is liquid then → LiquidtoVaporPhaseChange
- If the product spec. material is liquid and the current state of product material is vapor then → VaportoLiquidPhaseChange
- If the product spec. material is solid and the current state of product material is vapor then → VaportoSolidPhaseChange
- If the product spec. material is liquid and the current state of product material is solid then → SolidtoLiquidPhaseChange
- If the product spec. material is vapor and the current state of product material is solid then → SolidtoVaporPhaseChange

Table 4.2 Task Category Selection Rules

Selection of the Primary Alternative Operation

Once a task category has been identified, then the next step is to select an operation. This operation is called the selection of the *Primary Alternative*. The preconditions of the operations act as constraints for whether or not an operation can be applied in a particular situation. Thus, along with developing the operation models, the set of preconditions for each model must also be specified. In order to do this, a thorough understanding of the applicability of the model must be developed. The operation preconditions are stored along with the model as part of the operation. When determining which operation to choose as the primary alternative, it was stated that an operation that moves us "closer" to the goal state should be chosen. How is it to be determined if an operation moves the system closer to the goal state without applying the operation and generating the postconditions? Through the use of knowledge about the models associated with each operation, general postconditions about each operation can be developed. It is the attributes of the postconditions that are screened and matched to the goal state in order to chose the primary alternative. The postconditions are also stored along with the model and the preconditions. Table 4.3 contains the operations, their preconditions and the postcondition attributes. If no "best" operation can be determined, any operation from the task category can be selected and the planning is continued.

Task/Operation	Preconditions	Postcondition attributes
Charge	• None	• material transfer
Mix	• Number of inputs ≥ 2	• single output
Dissolve	• solid input • liquid input • $S_{Solute} \geq \frac{\text{mass of Solute}}{\text{Vol of Solution}}$	• single liquid output with dissolved solid
Cool	• none	• $T_2 < T_1$
Heat	• none	• $T_2 > T_1$
Decant	• single 2 phase liquid input • $V_1 \geq 0.1*V_2$	• 2 liquid outputs
Extract	• liquid input • $S_{MSA\ in\ solvent} \cong 0$ • $S_{solute\ i\ in\ MSA} > 0$ • $S_{solute\ j\ in\ MSA} < S_{solute\ i\ in\ MSA}$ • MSA present	• single 2 phase output • $C_{solute\ i\ in\ MSA\ out} > C_{solute\ i\ in\ MSA\ in}$ • $C_{solute\ i\ in\ solvent\ out} < C_{solute\ i\ in\ solvent\ in}$
Flash	• $K_i \geq 10$ for at least 1 component or vapor phase present* • $K_i \leq 0.1$ for at least 1 component or liquid phase present* * Douglas[4]	• liquid output • vapor output
Crystallize	• $C_{solute} \geq S_{solute}$ * • $T_{MPsolvent} < T_{MPsolute}$ • $S_{prod} < S_{others}$ • single liquid phase input • Composition of the input must lie within a region of the SLE that allows the desired product to from pure crystals	• 2 phase solid/liquid output • solid product crystals
Concentrate	• single phase liquid input • $P_{vap\ solvent} > P_{vap\ solute}$	• vapor output • liquid output • $C_{prod\ in\ Lin} < C_{prod\ in\ Lout}$
Condense	• $T_{BPi} > -48C^*$ • vapor input *Douglas[4]	• liquid output • vapor output
Evaporate	• liquid input	• vapor and/or liquid output
Batch-Distill	• $T_{BPi} \neq T_{BPj}$ • single phase liquid input • no azeotrope between keys	• multiple vapor or liquid outputs with $C_1 \neq C_2 \neq C_3$ ... • liquid output
Continuous-Distill	• $T_{BPi} \neq T_{BPj}$ • single phase liquid • no azeotrope between keys	• vapor stream • liquid stream
Filter	• single solid and liquid input	• liquid output • solid output
Absorption G-->L	• gas stream y_i • liquid stream x_i • $y_i > Kx_i$	• gas output • liquid output • $y_{i\ out} < y_{i\ in}$ • $x_{i\ out} > x_{i\ in}$
Stripping L-->G	• gas stream y_i	• gas output

	• liquid stream x_i • $y_i < Kx_i$	• liquid output • $y_{i\,out} > y_{i\,in}$ • $x_{i\,out} < x_{i\,in}$
Adsorption	• solid phase x_i • gas or liquid stream f_i • $f_i \leq$ capacity of adsorbent	• solid phase • gas or liquid stream • $x_{i\,out} > x_{i\,in}$ • $f_{i\,out} < f_{i\,in}$
Chromatography	• liquid input • $A_{solute\,i} \neq A_{solute\,i}$	• k liquid outputs • $x_{i,k} \neq x_{i,k+1}$
Ion exchange	• liquid input • $A_{solute\,i} \neq A_{solute\,i}$	• k liquid outputs • $x_{i,k} \neq x_{i,k+1}$
Membrane	• single phase liquid input or single phase gas input • $d_{solute\,molecule} > d_{membrane\,pore}$	• liquid or gas output • $x_{i\,out} > x_{i\,in}$ • liquid or gas output • $x_{i\,out} < x_{i\,in}$
Dry	• solid and liquid input	• vapor output • solid output
React	• reactants • catalysts • solvents • Dynamically Generate from Logic	• single output
Leach	• $S_{solute\,i\,in\,MSA} > 0$ • $S_{solute\,j\,in\,MSA} < S_{solute\,i\,in\,MSA}$ • solid phase input • MSA	• solid phase • liquid phase
Pressurize	• None	• $P_{final} > P_{initial}$
Evacuate	• None	• $P_{final} < P_{initial}$
Settle	• Single solid liquid phase input	• solid/liquid phase output • liquid phase output
Wash-solid	• $S_{solid\,in\,MSA} = \text{low}$ • MSA	• solid/liquid phase output • liquid phase output

Table 4.3 Operations, Precondtions and Postconditions

An important feature of the task category is that it provides a convenient method for maintaining a list of alternatives for each node in the plan and increases the efficiency of the search by reducing the search for the primary alternative to that of a search over key attributes, thus eliminating the need to search over all possible operations. This is the key to using MEA-NMP to construct a superstructure of feasible alternatives for optimization. The selection of the primary alternative allows us to generate a base-case design while simultaneously guiding the construction of the process superstructure. Each time a task category is selected in order to select the primary alternative, the task category is added to the superstructure plan. The superstructure that is generated is substantially smaller than the overall process superstructure and makes the optimization more efficient.

A more detailed description of the generation of process superstructures through the MEA-NMP methodology is given in Chapter 6.

4.5.2.4 Step 4 - Evaluate the Preconditions of the Proposed Operation.

Once the primary alternative has been selected, the preconditions for this alternative are screened to determine if any are violated. If no violations is detected we move on to Step 5. If a precondition is violated, then MEA is applied to try to identify a white knight that will eliminate the precondition violation.

Identify White Knights to Remove Precondition Violations

The white knight is identified by first selecting a white knight task category that has a postcondition corresponding to the removal of the precondition violation. Once the task category has been selected, a primary alternative is selected using the same procedure outlined in Step 3. If a white knight is identified, it is placed before the original operation and its preconditions are checked. This process is repeated until an operation is found that has no precondition violations. In Figure 4.16, this corresponds to the execution of *Evaluate Preconditions* block then moving to the *ID White Knight* block and then following the "yes" path to evaluate the preconditions of the white knight operation. An example of the removal of a precondition violation through the use of a white knight was given in Figure 4.5. From the Figure 4.5 it is clear that along with removing the precondition violation, the identification and placement of the white knight provides the temporal ordering of the plan. If no white knight is found then we promote the operation whose preconditions are violated.

Violated Precondition	**White Knight Task Category**
Liquid input → no solid	SolidLiquidPhaseSplit
Liquid input → no vapor	GasLiquidPhaseSplit
Vapor input → no solid	SolidVaporPhaseSplit
Vapor input → no liquid	VaporLiquidPhaseSplit
Solid input → no liquid	SolidLiquidPhaseSplit
Solid input → no vapor	SolidVaporPhaseSplit
Solid input	Material Transfer
Liquid input	Material Transfer

Vapor input	Material Transfer
$S_{MSA\ in\ solvent} \cong 0$	Solubility Reduction
$S_{solute\ i\ in\ MSA} > 0$	Solubility Increase
$S_{solute\ j\ in\ MSA} < S_{solute\ i\ in\ MSA}$	Dissolved Material Removal
MSA present	Material Transfer
$V_1 \geq 0.1*V_2$	none
Single 2 phase liquid input	Material Transfer
$K_i \geq 10$ for at least 1 component or vapor phase present	none
$K_i \leq 0.1$ for at least 1 component or liquid phase present	none
$C_{solute} \geq S_{solute}$	Solubility Reduction
$T_{MPsolvent} < T_{MPsolute}$	Solvent Replacement
$S_{prod} < S_{others}$	Solubility Reduction
Composition of the input must lie within a region of the SLE that allows the desired product to from pure crystals	Manipulate SLE
$S_{solute1} \neq S_{solute2}$	Solvent Replacement
$P^{vap}_{solvent} > P^{vap}_{solute}$	Solvent Replacement
$T_{Bpi} > -48C$	none
$T_{Bpi} \neq T_{Bpj}$	Pressure Change
key azeotropes = 0	Azeotrope Split
$y_i > Kx_i$	MultiComponentLiquidSep
$y_i < Kx_i$	VaporVaporSep
$f_i \leq$ capacity of adsorbent	Material Transfer
$A_{solutei} \neq A_{solutej}$	none
$d_{solute\ molecules} > d_{membrane\ pore}$	none
Reactant present	Material Transfer
Solvent present	Material Transfer
Catalyst present	Material Transfer
$S_{solid\ in\ MSA}$ = low	Solubility Reduction
$P^{vap}_{min} \leq P^{vap}_{solute}$	none
$T_1 = T_{spec}$	Temp change
$P_1 = P_{spec}$	Pressure change

Number of inputs ≥ 2	Material Transfer
$S_{solute} \geq \frac{\text{Mass of Solute}}{\text{Vol. of Solution}}$	Solubility Increase

Table 4.4 White Knights and Corresponding Task Categories

Promote operator if white knight cannot be found.

The promotion of the infeasible operation is an attempt to demote the clobbering condition and make the operation feasible. This is different than backtracking since we are not undoing any decisions that have been made. As a last resort, if promotion of the operation does not remove the precondition violations the operation is removed from the plan and an alternative operation is selected.

The procedure for the promotion of the infeasible operation begins with the failure to identify a white knight task category. In Figure 4.16 this is represented with the *fail* arrow leaving the *ID WK Task Category* block. Before we promote an operation we must first check to determine if there is a preceding operation in the plan, represented by the *Preceding Operation?* block. If there is no preceding operation, promotion is not possible thus, we remove the operation and either evaluate any remaining operations or attempt to identify a new alternative operation. This is represented by the *Remove Operation, Goto Starting Location* and *Remaining Operation?* blocks located in the lower right corner of Figure 4.16. If there is a preceding operation, we remove any charges associated with the operation and promote the operation represented by the *Remove Charges and Promote Operation* block in Figure 4.16. We then evaluate the preconditions of the promoted operation using the postconditions of the preceding operation. If there are any violations we attempt to identify a white knight using the white knight identification procedure described earlier. Once we have identified any necessary white knights and removed the precondition violations we go ahead and apply the promoted operation and generate its postconditions. We then evaluate the preconditions of the demoted operation. If there is a violation we attempt to identify a white knight. If a white knight cannot be found, we un-apply the promoted operation and repeat the promotion procedure. This series of steps is represented by the Apply *Operation and Generate Postconditions, Evaluate Preconditions of Next Op., ID White*

Knight and *Un-Apply Promoted Operation* blocks in the upper right corner of Figure 4.16. Once we have evaluated the preconditions of the demoted operation and any violations are removed we continue the planning by applying the operation and generating a new current state. From here we continue the planning as shown in Figure 4.16.

4.5.2.5 Step 5 - Apply the Operations and Generate the New Current State.

Once an operation is identified that has no precondition violations, the operation is applied and the new current state is generated. Application of the operation is carried out through execution of the model of that operation, using the current state as input. After the execution of the model, the resulting state becomes the new current state. If there are additional operations in the plan, the preconditions of the next operation in the plan are screened against the new current state. If violations exist then the white knight identification routine is applied and any white knights are placed before the operation whose preconditions are violated, but after the operation that has been executed. If no violations exist, then the operation is applied and another new current state is generated. This process is repeated until there are no remaining operations to be applied. The final new current state is compared against the desired goal state. If the new current state matches the desired goal state then the plan is complete. If there are any differences between the goal state and the new current state then the entire routine is repeated in order to remove these differences and achieve the goal state. Figure 4.16 depicts the application of all operations that have been identified and contains the iterative structure for screening for additional differences before signifying the plan as complete. The iteration is represented as the arrow that enters the *Current State* block.

4.5.2.6 Step 6 - Check the Plan for Abstract Operators.

Once the desired goal state has been achieved, the plan must be checked for *abstract operators*. Abstract operators may have entered the plan if the plan has been constructed hierarchically using operations of different levels of abstraction or as representations for planning islands. These abstract operators must be refined in order to achieve the detailed design. For example, the operator *Solvent Replacement* requires further refinement to specify the operations that actually perform the replacement of the solvent, such as *Extract* or *Concentrate and Mix*. If there are no abstract operators in the plan then, as shown in Figure 4.16, we have achieved the desired goal state and the planning is complete.

From the above description we can see that the methodology for planning is both iterative and recursive. It is iterative in the sense that the methodology loops over itself to continue identifying differences until the goal state has been achieved. It is recursive through its search for white knight operations and the evaluation and application of the operations that have been identified as necessary for the plan. It is the iterative and recursive nature of the MEA-NMP methodology that exploits the attributes of the nonmonotonic planning and has provided the means for a powerful and efficient methodology.

We have shown how the MEA-NMP methodology functions. It is an iterative method that relies heavily on the descriptions of its functional operators, their preconditions and postconditions. While the MEA-NMP methodology appears generic, it must be noted that its functionality relies on the *domain specificity*. The MEA-NMP methodology is applicable to other planning domains but key elements, such as the operator models and the task selection routine must be made specific to the problem domain of interest as has been done here for batch chemical process synthesis. We can also see from the above description of the MEA-NMP methodology and the example, that the MEA-NMP methodology relies on rules and logic in order to develop batch processes. It is the logic basis behind the MEA-NMP methodology which we will exploit in order perform process optimization. In Chapter 6, we will show how the MEA-NMP development of the process superstructure readily lends itself to the application of the logic-based superstructure optimization formulation proposed by Raman and Grossmann [15] (See Chapter 3). In the next section we provide a detailed example at the MEA-NMP methodology in order to develop a base case design for a pharmaceutical process. Chapter 5 will provide a detailed case study showing the planning of a batch process of the production of a pharmaceutical using the MEA-NMP methodology.

4.6 Application of MEA-NMP to a Batch Process

The most effective way to understand how the MEA-NMP methodology works is through its application to a practical example. The following example is an actual pharmaceutical process provided by a large pharmaceutical corporation. Here we apply the MEA-NMP methodology to develop the first stage, called the Carbinol stage, of an eleven stage process for producing a pharmaceutical. The complete case study can be found in Chapter 5.

4.6.1 Examine the Available Information

As Douglas [4] states: "You never have enough information." With this "rule" in mind we begin the development of a process knowing that we will not have all the information that we would like to have such as incomplete physical properties, unknown mixing rules and possibly undefined by-products. It then becomes the job of the engineer to develop a process under the constraint of a lack of information as well as to determine which information is absolutely necessary so that additional experiments can be designed and performed to collect this information. Fortunately, we are provided with a substantial amount of information about the process, which was collected by the chemist. It is this information and information which is available through various databanks, which we use to perform our initial designs and, as we shall see in most cases, the available information is at least sufficient enough to construct a base-case design.

We begin our process synthesis with the process description provided by the process chemist. The chemist provides us with a lab scale description of the process including the chemical reactions, the lab scale operating conditions (see Chapter 5), the solvents and catalysts used, the yield for the stage and the desired product specifications. In our case the chemist does provide us with the core information necessary in order to construct a base-case process plan. Figure 4.18 is a summary of the carbinol stage reaction chemistry taken from the chemist's process description.

From Figure 4.18 we see that the process consists of two main reactions that take the raw material, *trienone* (I) and through a Grignard reaction produce the desired product, *carbinol* (II). As a result of the Grignard reaction a quench reaction is required to deactivate the excess Grignard reagent. The Grignard reaction also generates a by-product in the form of a gel. This causes difficulty in the further processing of the mixture so an additional gel-break reaction is added. Finally, the product reacts with any acid present and generates an undesired by-product (III) through a side reaction.

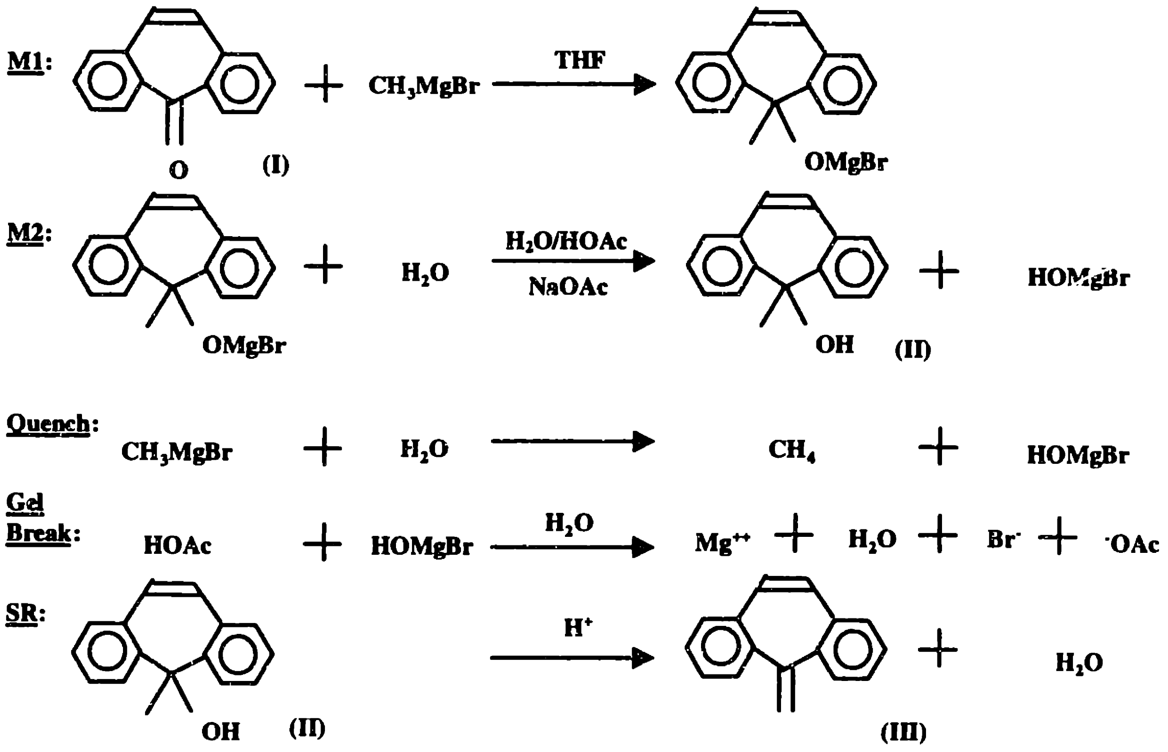


Figure 4.18 Carbinol Stage Reaction Chemistry.

Along with the reaction chemistry we are also provided with the desired product specifications such as purity and form and the yield. For the carbinol stage the carbinol is recovered as a white crystalline solid with a purity of 100 percent. From the product information we gain some insight into the process. In this case we see that the two main reactions are performed sequentially, with Reaction M1 being performed first and then the remaining reactions are performed simultaneously at the same operating conditions.

While there may be a number of external constraints imposed on the resulting process, such as minimum cost, recycle all non-product/by-products and minimum environmental impact, the main goal is to product the desired product at the required product specifications. In this example we will target the product production and show

how that allows us to gain some additional insight into the process. We will temporarily ignore any by-product streams and show that by targeting the main product production, we can get a picture of the type of waste streams the process will produce.

4.6.2 Generation of the Input/Output Structure

At this point we can simply generate an input/output structure for the stage by collecting the raw materials and the products and by-products. In the construction of the I/O structure, we do not include any materials that are not consumed by the reactions such as solvents and catalysts. This, essentially, imposes the constraint of recovering and recycling all non-product/by-product materials. Later we shall discuss the ramifications of this constraint on the process. Figure 4.19 depicts the input/output structure for the carbinol stage. While the construction of the I/O structure is not necessary for the MEA-NMP methodology, it provides us with a convenient method for summarizing the required raw materials and the by-products produced so we can get an early estimate of the types of materials we are going to have to deal with in our process.

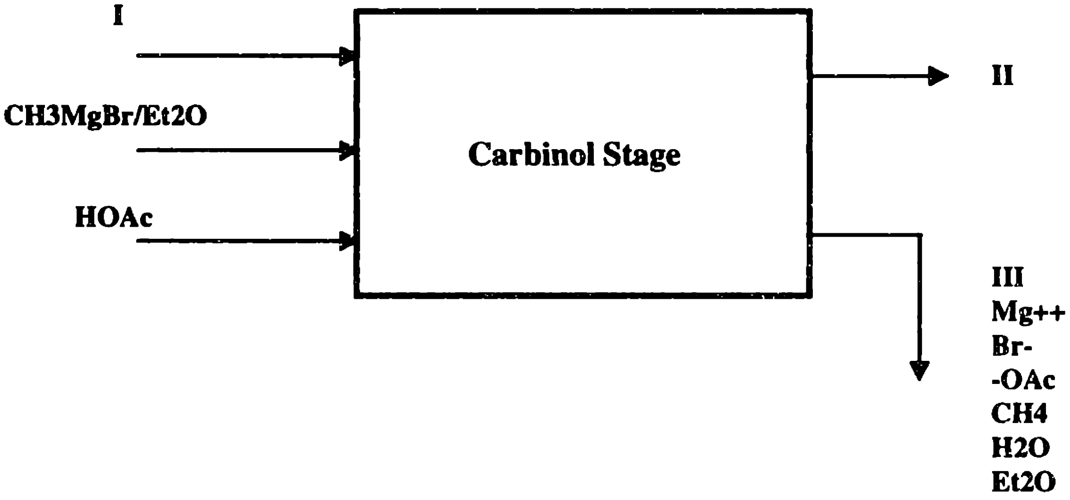


Figure 4.19 Carbinol Stage Input/Output Structure.

4.6.3 Difference Identification

Using the given information we can begin to apply the MEA-NMP methodology. Initially we define the current state as the raw materials for the first reaction, M1 and the goal state as the state of the product material, carbinol as a crystal with 100 percent purity. Now we identify differences between the current state and the goal state. This corresponds to the Identify Differences block in Figure 4.16, as well as Step 1 in the previous section.

We begin the difference identification with the first type of difference in Siirola's difference hierarchy [5], which is the identity difference. For the carbinol stage, it is clear that the molecular structure of the raw materials differs from the molecular structure of the product material. Thus, we have identified our first difference, the identity difference. While this may seem trivial, it is demonstrates the systematic nature of the methodology as well as demonstrating the power of the methodology to be applied from the beginning to the end of the process synthesis.

4.6.4 Critical Feature Screening

The next step is to identify any critical features such as azeotropes and solid eutectics. At this point there are no critical features so we move on to Step 3, application of means-ends analysis.

4.6.5 Application of MEA

Since we have already identified the state category, we next must identify the appropriate task category. Figure 4.17 shows that for the State Category identity difference there is only one task category, molecular structure transformation. At this point we need to identify an operation as our primary alternative. Again, this is made very simple for us since the only operation that can perform a molecular structure transformation is a reaction. The only question that remains is, which *reaction set* do we chose? Here we define a reaction set as those reactions that take place using the same raw materials and at the same operating conditions. Through this definition, in this stage we have two reaction sets, the first containing Reaction M1 and the second containing all remaining reactions, see Figure 4.18. When choosing the reaction set, we could choose either the first reaction set and proceed in a synthetic manner through a direct application of MEA or we could choose the second reaction set and exploit the NMP to "discover" the first reaction set. Since the reaction scheme limits the number of alternatives, it doesn't matter which one we choose to perform first, the methodology will identify the same reaction scheme. For this example, let us choose to perform the reaction set M1 first. We insert reaction M1 into the plan but before we can apply the reaction we must check its preconditions. Figure 4.20 depicts the insertion of reaction step M1 into the plan. M1 is connected using dashed lines since its relation to the rest of the plan is still undetermined, in essence it is a planning island. Once the actual relation among operations is established, the dashed lines will be replaced with solid lines.

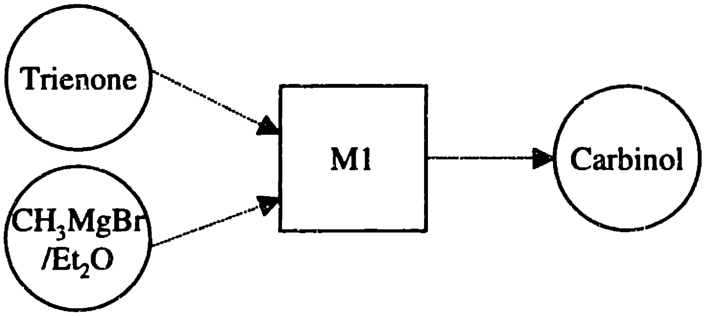


Figure 4.20 Insertion of Reaction Step M1 into the Plan.

4.6.5.1 Dynamic Determination of Reaction Preconditions.

Due to the uniqueness of the reaction chemistry to each process, the reaction preconditions for each reaction set must be determined dynamically. This is done through an interactive system that identifies certain features in the reaction set and then prompts the developer to enter information based on some predefined questions. For example, the trienone is a solid raw material. Whenever a solid raw material is present, the system prompts the developer by asking if the solid should be dissolved before the reaction. If the developer selects "yes," the developer is prompted to select a solvent. This amounts to the posting of a precondition that states the raw material must be dissolved before the reaction. Since we have directly identified an operation through the interaction with the developer, we can by-pass the normal white knight identification procedure and insert a dissolve operation directly into the plan as a white knight before the reaction and the reactant, trienone, is dissolved in the solvent, tetrahydrofuran. Another dynamic precondition is generated by checking the heat of reaction. If a reaction is highly exothermic, the system prompts the developer asking if additional cooling should be used. This is the case for the Grignard reaction. It is highly exothermic, so the system prompts the developer asking if additional cooling should be used. In this case the developer selects "yes" to cooling and a cool operation is added which takes place simultaneously with the reaction. We see that for this stage the developer has chosen to mix the methylmagnesium bromide with a solvent before reaction. The methylmagnesium bromide is provided in a solution with ethyl ether so the solution is mixed with the solvent. Then a mix operation is placed before the reaction step in parallel with the dissolve operation for the dissolution of the trienone. Table 4.5 contains the logic used to define the reaction preconditions.

Reaction Precondition Logic

1) Use Reaction Set Information to Define (Automatically or Manual):
   a) Reactants
   b) Products
   c) Catalysts
   d) Buffers
   e) Diluents
   f) Solvents

Note: The reaction logic functions by creating a set of partial plans which are then connected by the developer.

2) Screen the reaction set for side reactions → For each side reaction attempt to minimize by:
   a) Premix the offending reactant with a solvent?
      - Yes → Select Solvent → generate partial plan → Charge solvent, reactant and a Mix operation.
   b) Add buffer to solution?
      - Yes → Mix with solvent? → Yes → Select Solvent → Generate partial plan → Charge solvent, buffer and Mix operation.
   c) Eliminate Charged reactants and buffers from consideration list.
3) If Catalyst is used then
   a) Premix catalyst with solvent?
      - Yes → Select Solvent → Generate partial plan → Charge solvent, catalyst and Mix operation.
      - No → Generate partial plan → Charge catalyst.
   b) Eliminate Charged catalyst from above.
4) Screen reactants for solid or liquid reactants – For each S or L reactant
   a) Dissolve or Mix with solvent?
      - Yes → Select Solvent → Generate partial plan → Charge solvent, reactant and Dissolve or Mix operation.
      - No → Generate partial plan → Charge reactant.

  b) Eliminate Charged reactants from above.

5) Screen reactants for gaseous reactants – For each gaseous reactant
   a) Dilute?
      - Yes → Select Diluant → Generate partial plan → Charge diluant, reactant and Mix operation.
      - No → Generate partial plan → Charge reactant.
   b) Eliminate Charged reactants from above.
6) Screen for any Materials that have not been Charged
   a) Generate partial plan → Charge material.
7) Specify ordering of addition of all the partial plans created above.
   a) If the same solvent is used in sequential partial plans then suggest combining Mix operations.
   b) Generate partial plan → Charge and Mix steps for React operation.
8) After specifying connections check preconditions on all operations before they are applied.

Check Heat of Reaction
a) Exothermic → Cool?
   - Yes → Add simultaneous cooling
b) Endothemic → Heat?
   - Yes → Add simultaneous heating

Table 4.5 Reaction Precondition Logic

In this example, the interaction with the developer allowed us to by-pass the white knight identification through task categories. This is not always the case and the interaction with the developer can generate precondition violations that are resolved through the task category-white knight identification. Figure 4.21 shows the insertion of the dissolve and cool white knights into the plan.

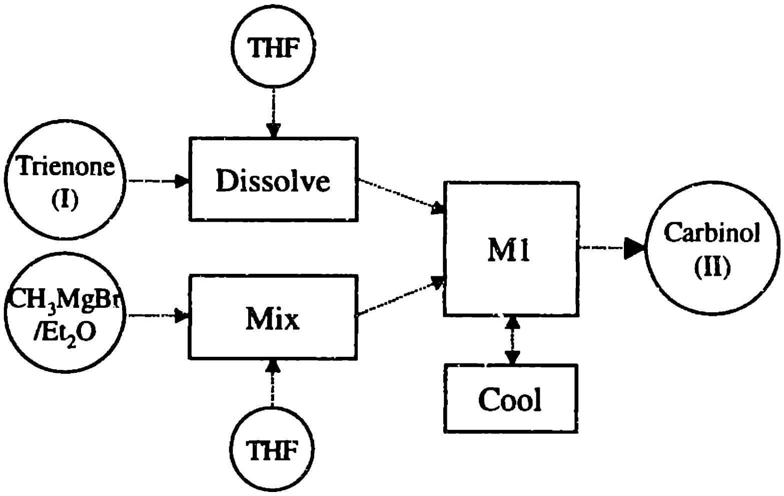


Figure 4.21 Insertion of White Knights into the Plan.

4.6.6 Evaluate Preconditions and Apply Operations

Before we can apply the white knights we must first check the preconditions of the white knight operations. Table 4.3 contains the list of operations and their corresponding preconditions. For the dissolve operation the preconditions are:

a) One input stream must be a liquid.
b) One input stream must be a solid.
c) The mass of solute divided by the volume of solution is less than the solubility for the solute in the solvent.

The dissolution of trienone in tetrahydrofuran (THF) does not violate any of these preconditions so we apply the dissolve operation and generate the postconditions. The postconditions are *trienone dissolved in THF*. In the case of the mixing of methylmagnesiumbromide with THF, we have a single precondition, the number of inputs must be greater than or equal to two. Since we have 2 input streams, this precondition is satisfied and we can apply the mix operation and generate the methylmagnesiumbromide-THF solution. When evaluating preconditions, if the information necessary to evaluate the preconditions is unavailable, we assume that there is no precondition violation. For the cool operation there are no preconditions since theoretically we can always cool a mixture, thus we can apply the cooling operation directly into the plan. Figure 4.22 shows the application of the dissolve, mix and cool operations and the generation of the corresponding postconditions. Note that the dashed arrows, up to the new postconditions, are now solid indicating the application of the operations.

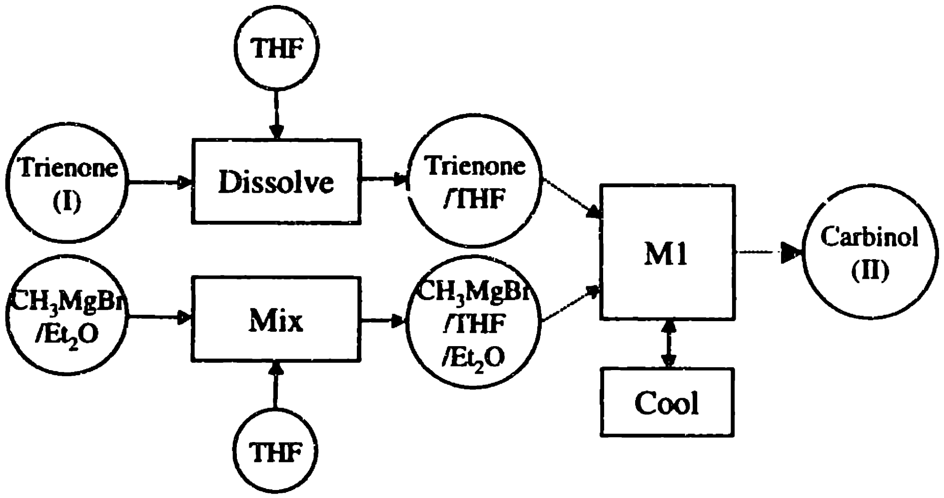


Figure 4.22 Application of the Dissolve, Mix and Cool and Generation of Postconditions.

By analysis of the above procedure, we see that we can simultaneously identify operations along with the order which these operations should be performed with respect to each other. One area in which the methodology cannot help us is in the ordering of addition of the reactants. While there may be some rules that could guide us, such as add acid to water not water to acid, there are too few of these rules and there are too many exceptions to these rules. So we are left with no other option but to ask for assistance from the process developer. For example, the dynamic precondition logic prompts the developer to determine if steps should be combined, such as the *mix* and *dissolve* operations shown in Figure 4.21. In this case, the developer has decided not to combine these steps (see Table 4.5, Step 7). This leads into another important point. The methodology always defaults to the developer to supply the necessary information when the data is unavailable from other sources. Also, especially in the reaction scheme synthesis, there are instances where the system is unable to determine all the necessary operations. To overcome this limitation the developer must have the freedom to intervene at any point in the synthesis to add necessary operations. For the carbinol stage, let us proceed with the development without any intervention by the developer and then go back and compare the resulting process with the process that was developed by the chemist to identify possible points of intervention.

4.6.6.1 Application of Remaining Operations in the Current Operation List

Now that the dissolve, mix and cool operations have been applied and the postconditions generated, these postconditions become the new current state of the system which we must use to evaluate the preconditions of the reaction operation. We now find that none of the preconditions is violated, so we can go ahead and apply the reaction operation and generate its postconditions. These postconditions become the new current state. Figure 4.23 shows the application of M1 and the generation of its postconditions.

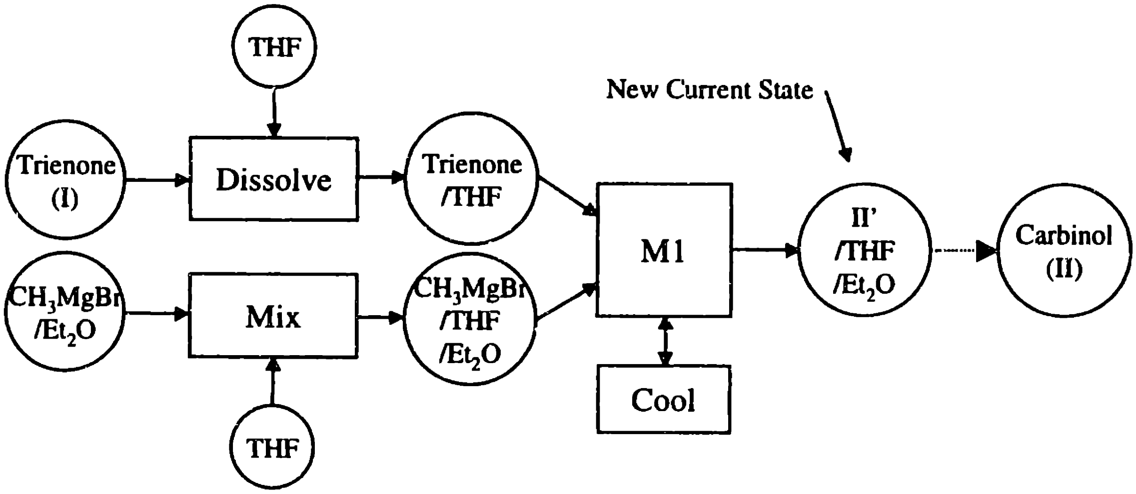


Figure 4.23 Application of the Reaction Operation M1 and the Generation of its Postconditions.

4.6.7 Identification of Differences and Screening for Critical Features.

From Figure 4.23 we see that we have generated a new current state consisting of a trienone-methylmagnesiumbromide complex (II'), THF solvent and the Et_2O solvent. The cool operation has maintained the temperature at the temperature specified by the reaction conditions. Now we again apply the MEA-NMP methodology by identifying differences between the new current state and the goal state. We do this exactly as described previously through the examination of the difference hierarchy. We also screen for any additional critical features that may have entered the process, but again we find there are none. We see that there is an identity difference and if we proceed as above, we identify the Molecular Transformation task category and determine that we must add a reaction operation corresponding to the second reaction step, which includes the second main reaction (M2), the quench reaction (Quench), the gel break reaction (Gel Break) and the side reaction (SR). Figure 4.24 show the addition of Reaction Set 2 and the accompanying raw materials to the plan.

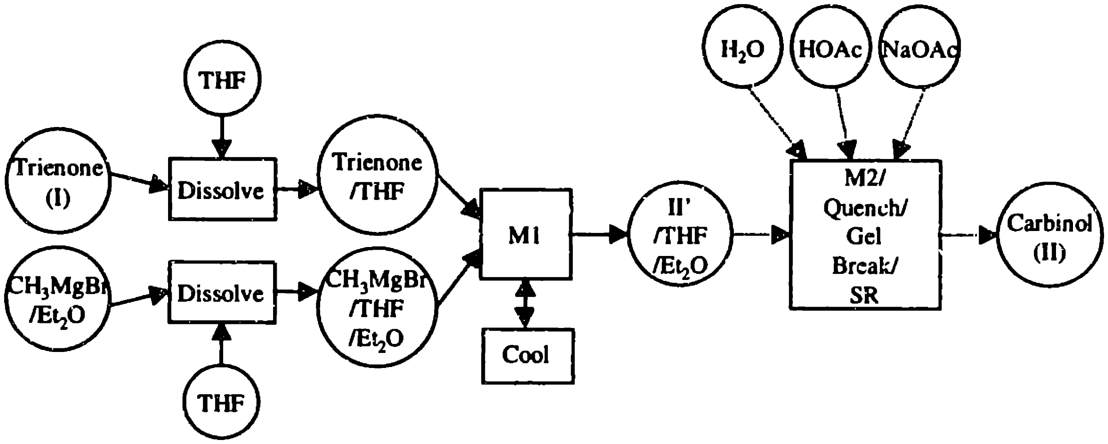


Figure 4.24 Addition of Reaction Set 2 to the Plan.

4.6.8 Screen for Precondition Violations

Examining Figure 4.24 we see that three new materials have been added to the plan along with the second reaction step. Before we apply the second reaction step, we must again determine if there are any precondition violations through the dynamic precondition generation as described above. As with the previous reaction step a solid material has been detected so the developer is prompted by asking if dissolution is required. In this case dissolution is required so a dissolve step will be added to the plan. Additionally, a side reaction has been detected so we apply a screening logic to help determine possible methods for reducing the side reaction. This logic is based on expertise of chemists. For instance, one method for reducing the side reactions is to keep the concentration of the offending reactant low. Here we prompt the developer and ask if the acetic acid should be mixed with a solvent. If the developer answers "yes" then the developer must select a solvent and a mix operation is added to the plan before the reaction.

The reaction logic generates the different reaction white knights as individual partial plans for which the developer must specify the ordering of execution. For example, the reaction output from reaction set one is considered a partial plan, as is the dissolution of the NaOAc in H_2O and the mixing of HOAc with H_2O. This is due to the inability of the reaction logic to identify the precise ordering for these operations and their addition to the second reaction set. In this case, the reaction logic has identified a case where the partial plans that have been developed through the interaction with the developer may be combined. Since the NaOAc and the HOAc use the same solvent and the dissolve and mix operations take place sequentially, we may be able to combine the sequential operations before the reaction. The system prompts the developer and in this case the developer selects "yes" so the steps are combined on a single line before the reaction as shown in Figure 4.25. As with the previous reaction task, these white knights are placed into the plan without resorting to the standard white knight identification procedure. Notice that the operations are connected with dashed lines indicating that the preconditions have not been evalua ed.

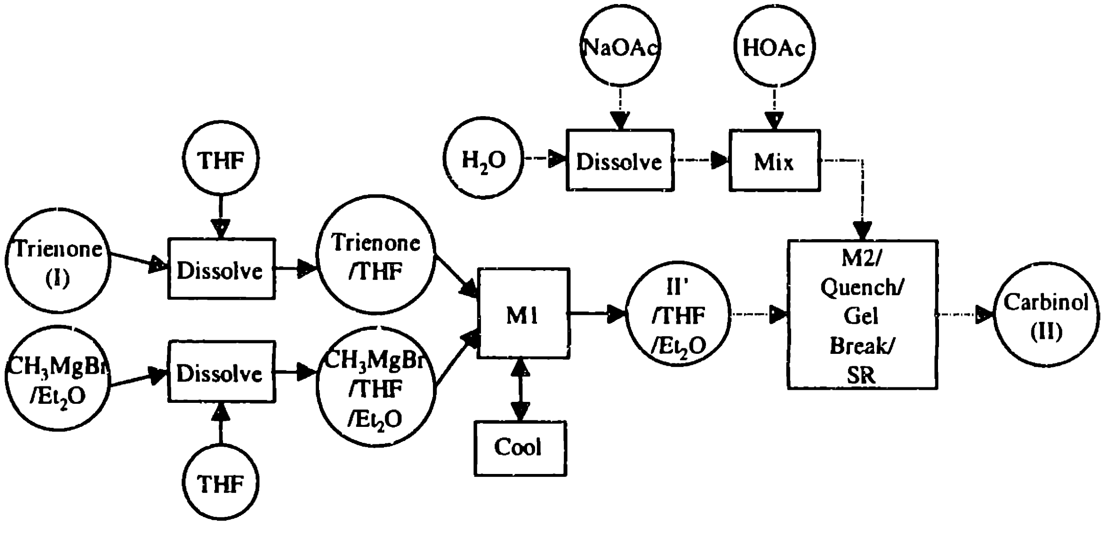


Figure 4.25 Addition of Dynamic White Knights to the Plan.

The next step is to check the preconditions on the new dissolve operation. None of the preconditions is violated so we can go ahead and apply the operation and generate its postconditions. We then move on and check the preconditions for the mix operation. The preconditions for the mix operation are similar to the dissolve operation. In this case the preconditions for the mix operation hold and we can apply the operation and generate the mix postconditions. We then evaluate the preconditions for the reaction set. We again detect an exotherm so the developer is prompted about the addition of an external cooling operation. For this process, the developer selects yes and a cooling operation is added to the plan to maintain the reaction mixture at the reaction operating conditions. Now there are no longer any violated preconditions for the second reaction set so we can apply the reaction set and generate the postconditions. Notice that the application of the previous three operations corresponds to "unwinding" the recursive white knight generation procedure. Figure 4.26 shows the complete reaction structure for the Carbinol stage.

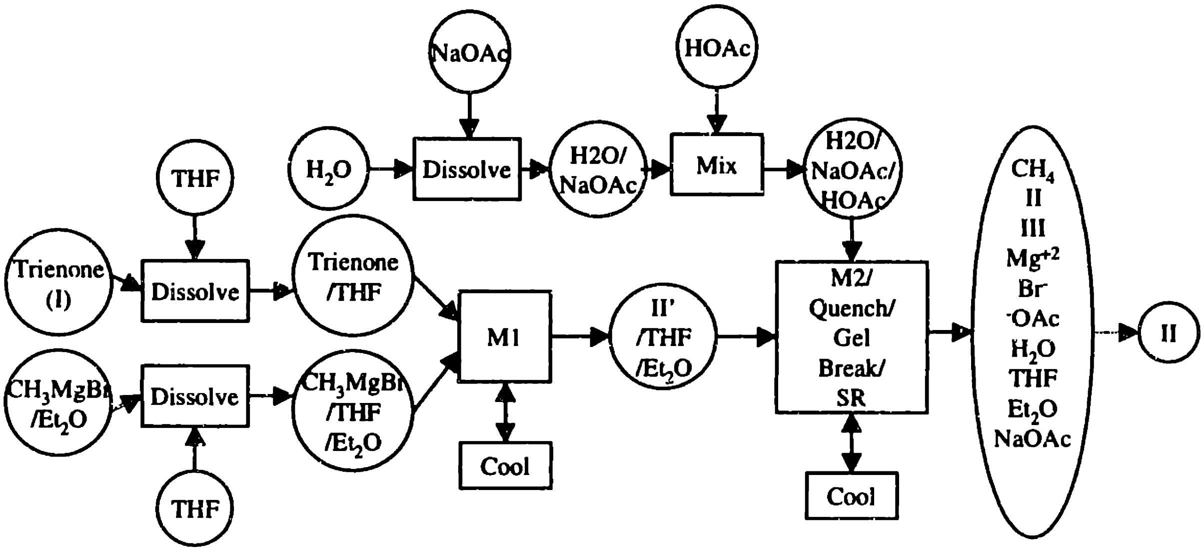


Figure 4.26 The Complete Carbinol Reaction System.

Once the reaction network has been completed, we examine the output from the final reaction step. In Figure 4.26 we see that we have a multiphase multicomponent mixture consisting of a vapor phase, an aqueous phase, an organic phase and dissolved solids. The observant reader may suggest that normally we would not have a stream leaving the reaction that contains these mixed phases. Here we must remind ourselves that we are dealing with an operations-based synthesis system so each operation must be explicitly identified, we cannot assume that a vapor liquid mixture will emerge as two distinct streams until this separation has been identified by the methodology. Once all the operations have been identified we can review the plan and look for *stream merging* and *task merging* opportunities. Stream merging in batch processes implies combining streams with similar compositions using intermediate storage in order to process the combined streams using a single task sequence. Stream merging in the carbinol stage will be discussed at the end of this section. Task merging, on the other hand, implies performing multiple operations in a single piece of equipment, for example reactive extraction combines a reaction task and an extraction task in the reaction vessel. Task merging impacts the amount of, and types of equipment required for the process. Task merging will be discussed in Chapter 6 and Chapter 7. For now let us continue on to the generation of the separation structure.

4.6.9 Separation System Generation

The generation of the separation system requires much less interaction with the developer than the reaction system development. This is because the preconditions of the operations used in the separation system can be standardized and are not generated dynamically. This does not imply that developer intervention is discouraged. At any point in the process, the developer is free to intervene in the process synthesis. Also, at points in the process synthesis, important information may not be available in the system databases and we would need to prompt the developer to provide this data. For now, let us proceed assuming the developer does not intervene in the synthesis.

4.6.9.1 Difference Identification.

We begin applying the MEA-NMP methodology by using the output from Reaction Set 2 as the new current state and comparing it to the goal state to identify differences. The mixture from Reaction Ret 2 contains the desired product so we do not have an identity difference. At this point we are producing a single batch so we do not have an amount difference. The goal state is pure carbinol, so it is obvious that there is a concentration difference between the current state and the goal state. Thus, we now attempt to remove the concentration difference.

4.6.9.2 Screen for Critical Features.

Following the MEA-NMP methodology, once the type of difference has been detected we screen for any critical features. We do not detect any critical features so we continue on to the task category selection.

4.6.9.3 Task Category and Operation Selection.

With such a multicomponent, multiphase mixture there are a number of tasks that could be performed in order to begin reducing the concentration difference. We could begin with phase splitting operation or we could try to eliminate component by component from the mixture. Since our ultimate goal is to produce the product material, we can use this to focus our synthesis efforts. This is done by targeting the product material for separation. Since our desired product is a crystalline solid and the product is dissolved in a liquid, we begin by selecting a task category that allows the isolation of a dissolved solid from a liquid. This task category is LiquidtoSolidPhaseChange. Once we have selected the task category, we then must select a primary alternative. In this case, since we want to produce a solid crystal we choose *crystallize*. Figure 4.27 shows the addition of the crystallize operation into the plan. In the figure we have included the actual phases that are present, we see that it is a three phase mixture consisting of a gas phase, an aqueous phase and an organic phase.

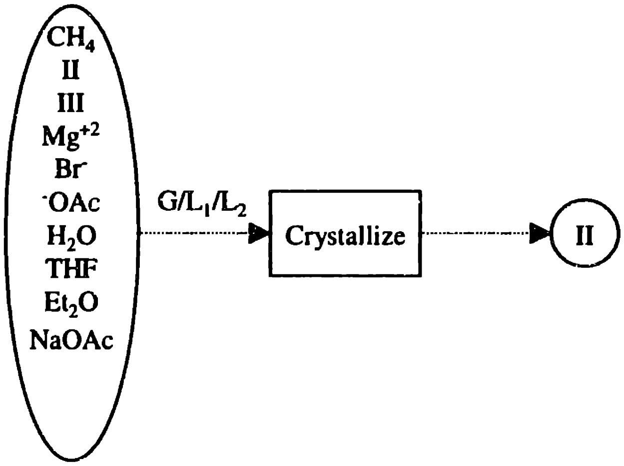


Figure 4.27 Identification of the Crystallize Operation.

4.6.9.4 Evaluate Preconditions, Identify White Knights and Apply Operations.

Before we can apply the crystallize operation, we must evaluate its preconditions. The preconditions for the crystallize operation are:

a) No vapor phase as input.
b) No multiple liquid phase as input.
c) No solid phase as input.
d) The composition of the feed mixture must lie within a region of the solid-liquid equilibrium that allows the desired component to form pure crystals.
e) The melting point of the solvent must be less than the melting point of the solute.
f) The concentration of the solute must be greater than the solubility of the solute.
g) Solubility of the product must be less than the solubility of all other components.

Comparing the preconditions to the current state, we see that precondition (c) is not violated. Preconditions (a) and (b) are clearly violated and preconditions (d), (e), (f) and (g) require additional information. Whenever additional information is require to evaluate preconditions, we continue the design attempting to eliminate the known precondition violations and then we return to those preconditions which require more information.

Within the MEA-NMP methodology we now proceed to Step 4 and identify a task category that contains operations which can eliminate the precondition violations. This is done by searching a list of precondition violations and identifying the corresponding task category that can resolve the violations. One may ask how do we know which precondition violation we should eliminate first? Within the MEA-NMP methodology the order in which the violations are eliminated does not matter, the resulting process will be the same. For example, Illustration 4.9 and Illustration 4.10 show how we can either remove violation a or b first and the resulting ordering of operations in the plan is not effected.

Illustration 4.9

Beginning with precondition (a), no vapor phase as input, we must identify a white knight. We first identify the white knight task category from Table 4.4. The corresponding task category is GasLiquidPhaseSplit. Next, we select the primary alternative which in this case is the Flash operation. The flash is inserted into the plan before the crystallize operation as shown in Figure 4.28.

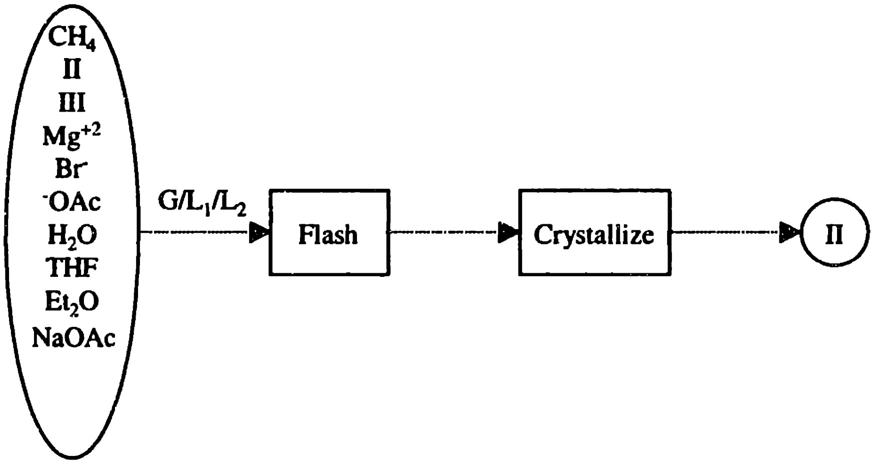


Figure 4.28 Addition of the Flash Operation into the Plan.

We then proceed to check the preconditions for the flash operation. The preconditions for the flash operation are

a) $K_i \geq 10$ for at least one component [4] or vapor phase exists.

b) $K_i \leq 0.1$ for at least one component [4] or liquid phase exists.

It is clear from the description of the exit stream from Reaction Set 2 that neither of these preconditions are violated, thus we can go ahead and apply the flash operation and generate the postconditions, see Figure 4.29.

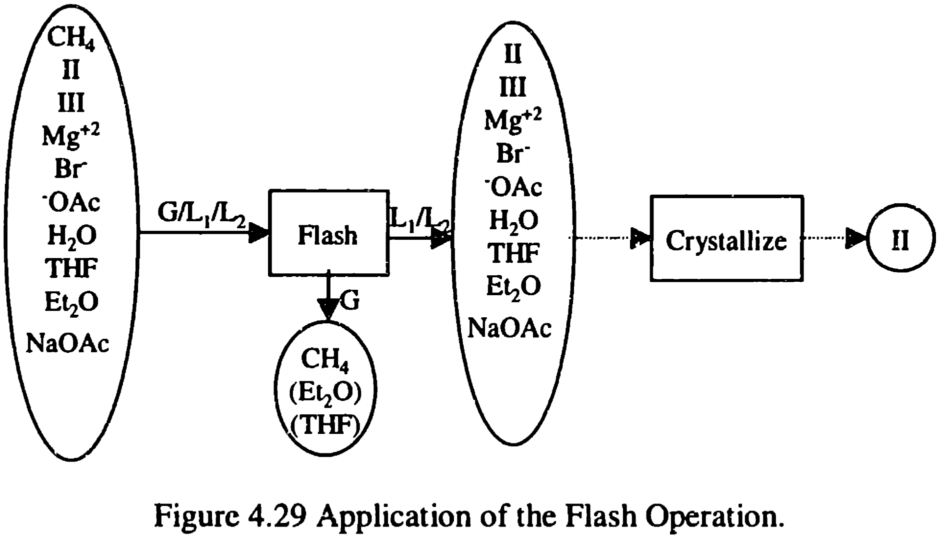


Figure 4.29 Application of the Flash Operation.

Now we again check the preconditions of the crystallize operation. We now find that precondition (a) is no longer violated but precondition (b) is still violated, so we no attempt to identify a task category that can resolve the violation. The white knight task category that is identified is the LiquidLiquidPhaseSplit task category. We now choose the primary alternative which in this case is the decant operation, see

Figure 4.30.

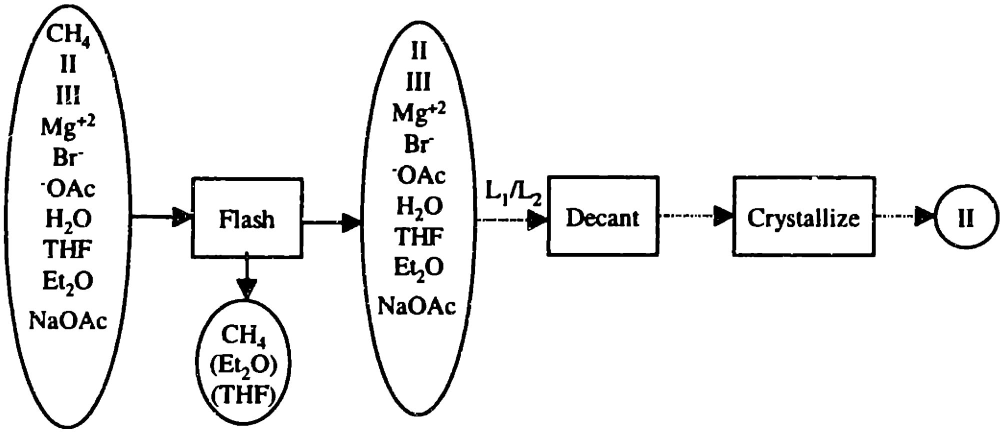


Figure 4.30 Addition of the Decant Operation to the Plan.

Before the decant operation can be applied its preconditions must be checked against the new current state generated by the flash operation. The preconditions for the decant operation are i) no gas phase input and ii) two phase liquid input and iii) $V_1 \geq 0.1 * V_2$. We see that these preconditions hold with respect to the input stream so we can apply the decant operation, see Figure 4.31.

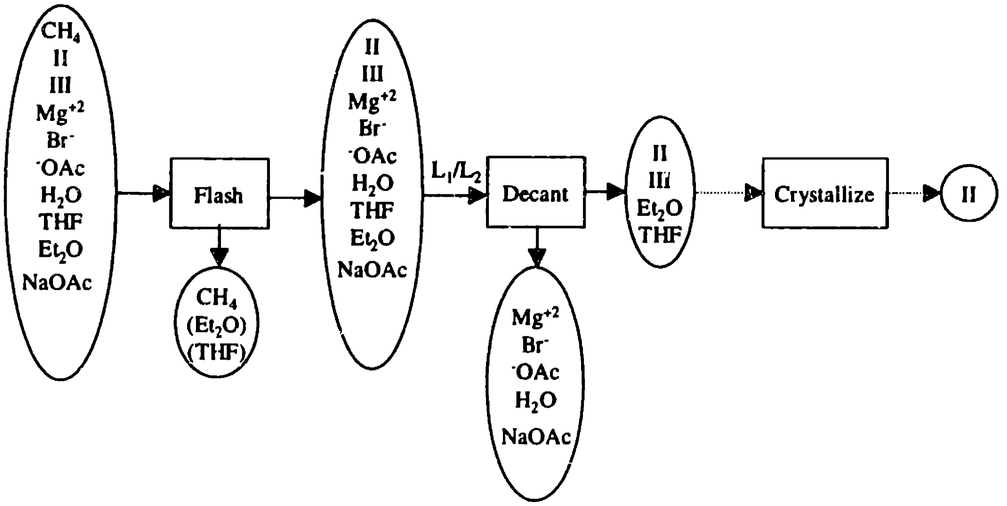


Figure 4.31 Application of the Decant Operation.

Now that we have eliminated the crystallize precondition violation (a) first and then (b), let us go back and show how the plan would evolve if we had decided to eliminate violation (b) first.

Illustration 4.10

We begin resolving b by first selecting a white knight task category and then the primary alternative. As above, for this precondition violation we select the decant operation, see Figure 4.32.

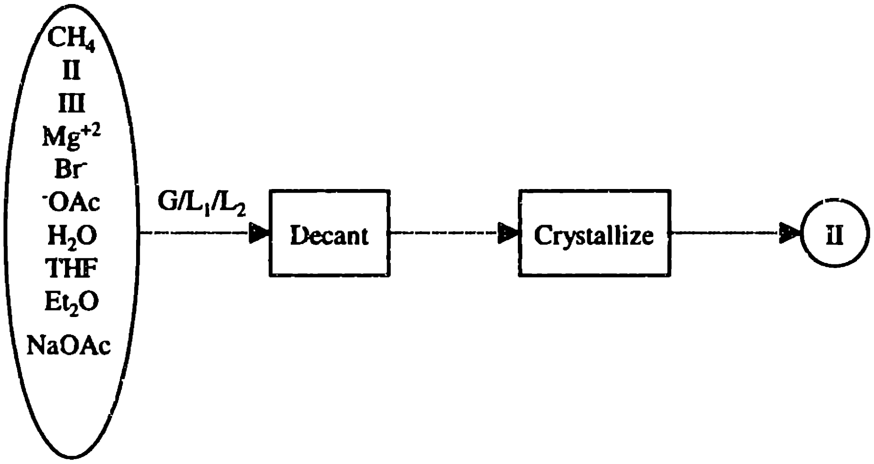


Figure 4.32 Addition of the Decant Operation to the Plan.

Now we check its preconditions against the output from reaction set two. Here we have a violation of the no vapor phase as an input precondition for the decant operation. We then leave the decant operation in the plan and search for a white knight that can remove the precondition violation of the decant operation. To remove the vapor phase, we select the *GasLiquidPhaseSplit* task category and select the flash operation as the primary alternative. We the insert the flash operation into the plan before the decant operation, see Figure 4.33.

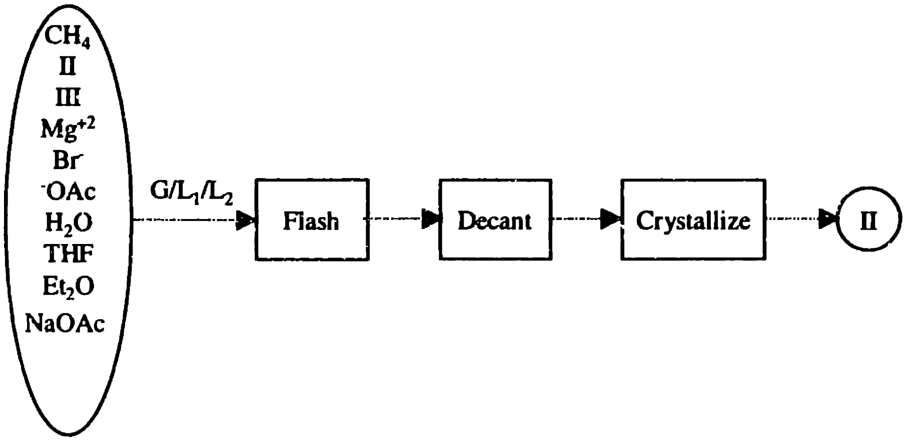


Figure 4.33 Addition of the Flash Operation to the Partial Plan.

We then check the flash operation preconditions against the current state generated by reaction step two. Here we find no precondition violations so we go ahead and apply the flash operation and generation the postconditions as shown in

Figure 4.34.

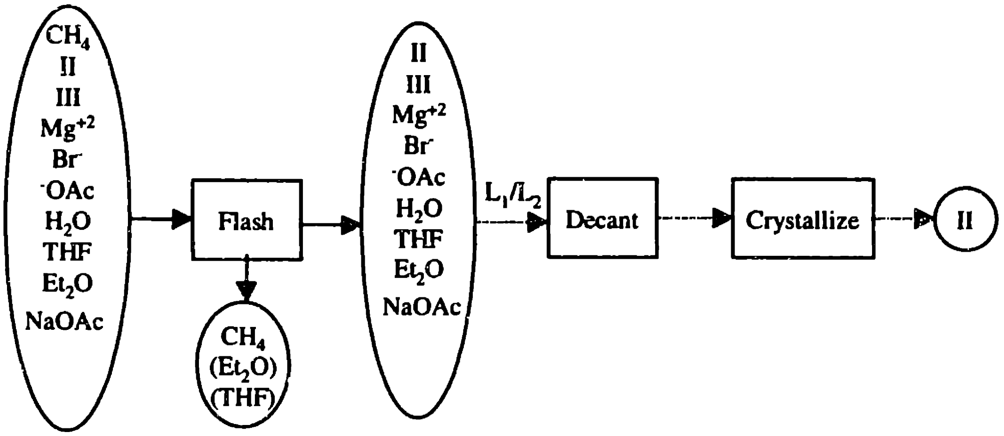


Figure 4.34 Application of the Flash Operation.

From here on we see that the development of the plan proceeds in exactly the same manner as above with the checking of the decant preconditions, application of the decant operation and generation of the postconditions. The resulting plan is the same as in Figure 4.31.

Now that we have shown that the ordering of the selection of precondition violations for resolution does not effect the resulting plan lets continue with the separation system synthesis beginning with Figure 4.31.

The new current state is the product containing layer from the decant operation. We now compare the new current state to the preconditions of the crystallize operation. We see that preconditions (a), (b) and (c) are now satisfied. We now need to collect information regarding the remaining preconditions: (d) the composition of the feed mixture must lie within a region of the solid-liquid equilibrium (SLE) that allows the desired component to form pure crystals, (e) the melting point of the solvent must be less than the melting point of the solute, (f) the concentration of the solute must be greater than the solubility of the solute at the minimum operating temperature and (g) the solubility of the product must be less than the solubility of all other components. This information is not always readily available, especially for new compounds such as carbinol. Generally, the chemist has obtained some of this information during the laboratory development of the reaction and separation scheme. For instance, while the chemist may not have the complete SLE diagram, they generally have a few points since they have performed a crystallization in the lab. Another option may be to estimate the required properties using compounds whose properties we already know and whose molecular structure is very similar to the compound of interest. While this is not very reliable it may be sufficient to complete a base-case design, and then, once it has been decided that this is a process of interest, we could go back and do the necessary experiments to collect the required information. A third option, which is becoming increasingly attractive and viable, is the use of property estimation methods. Advances in the area of property estimation allow for the estimation of many vapor and liquid systems. Eventually, the reliable estimation of solid properties will allow the use of these tools.

For now let us proceed with the information we have available. The melting point for THF is −108 C. Since carbinol is a crystal at room temperature, we can assume that the melting point of THF is lower than that of carbinol so precondition (e) is not violated. Since the chemist has produced the material as a pure crystal, we can assume that precondition (d) is not violated. Next, we must check the concentration of carbinol and its solubility in THF over the operating temperature range. We find that the concentration of the carbinol in the solution is lower than the solubility of carbinol over the temperature range of interest, thus precondition (f) is violated. We now identify a white knight task category and select a primary alternative. For this example the task category is *SolubilityReduction*.

Examining the elements of the SolubilityReduction task category we see that it contains within it another task category named *SolventReplacement*. This demonstrates the capability of this methodology to allow the definition of sub-categories. The sub-categories can then provide the developer with two options. First, when information is lacking, we can define the sub-category as an abstract operator that has behind it a generic model of the task. For example, SolventReplacement could have a model that simply exchanges one solvent for another. We would then come back and refine this abstract operator after we have obtained more information. The other option is to go ahead and select an operation from within the SolventReplacement sub-category that can perform the task. This allows the developer to continue to develop the base-case design including all the detailed operations.

In this example, let us go ahead and select an operation directly and continue with the development. The operation set we select is *concentrate and mix.* This points out another feature of the methodology, if certain operations need to be performed together to achieve a combined task, then we can add the group of operations to the plan. Thus, by selecting *concentrate and mix*, we insert them into the plan simultaneously. We should note however, that even though we insert these operations into the plan together, we still need to check the preconditions on operation independently. The addition of the concentrate and mix is shown in Figure 4.35.

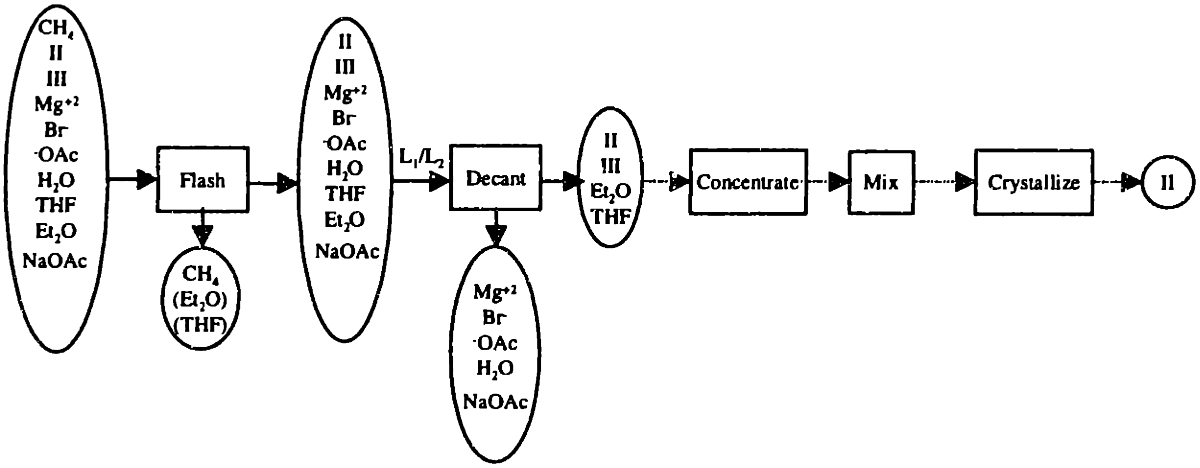


Figure 4.35 Addition of Concentrate and Mix to the Plan.

We now check the preconditions of the concentrate operation. No violations are detected so the concentrate operation is applied and the postcondition is generated, see Figure 4.36.

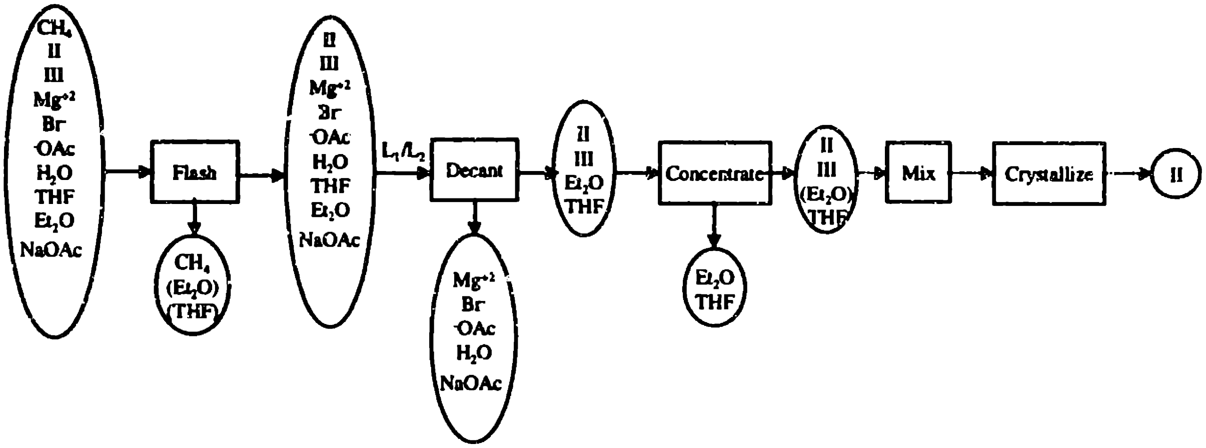


Figure 4.36 Application of the Concentrate Operation.

Again, before we can apply the mix operation we must check its preconditions. We find no violations so we can go ahead and apply the operation. For this operation we need to prompt the developer for a solvent to use to replace THF. Here we will use the solvent that has been suggested by the chemist, cyclohexane, see Figure 4.37.

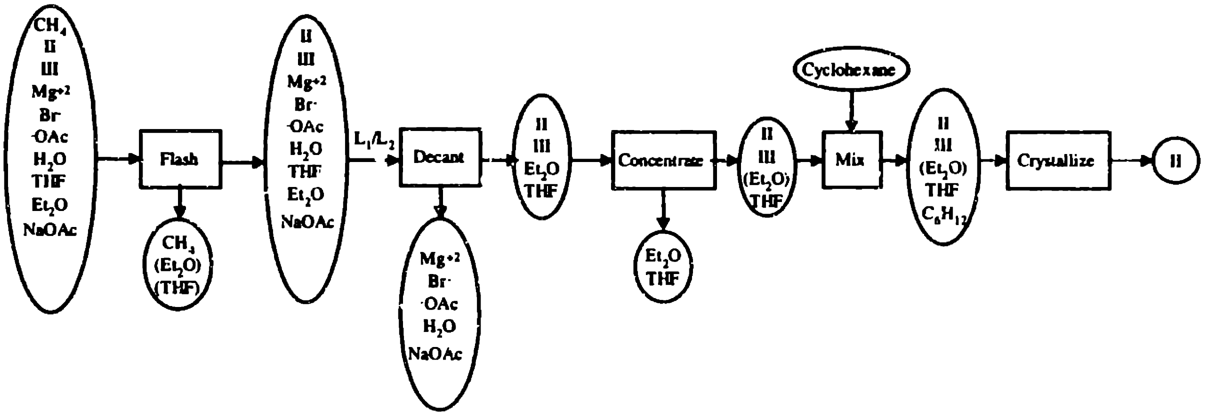


Figure 4.37 Application of the Mix Operation.

We now check the preconditions of the crystallize operation and find that precondition (f) is still violated. We then go ahead and look for a white knight task category and a white knight operation. Generally, when concentrate and mix are selected as the solvent replacement option, they are performed more than once in order to get the necessary dilution of the original solvent. Let us, again, select a concentrate and mix operation. We add the operations to the plan before the crystallize operation, check the preconditions on the concentrate operation, if none are found we generate the postconditions and then check the preconditions on the mix operation. Again, none are found so we can apply the mix operation and generate the postconditions, see Figure 4.38.

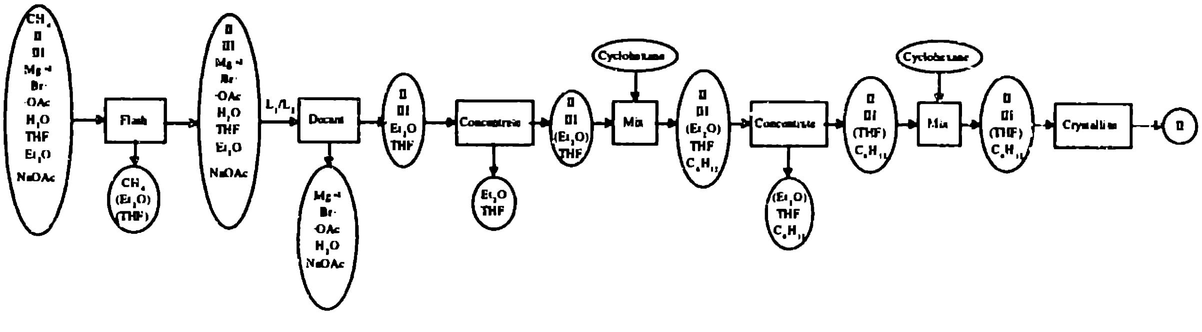


Figure 4.38 Application of Additional Concentrate and Mix Operations.

As above, we check the preconditions on the crystallize operation with respect to the new current state generated by the second mix operation. Now the concentration of THF is sufficiently low so that precondition (f) for the crystallization is no longer violated. We now check the final precondition (g). Since we are not supplied with solubility information for compound III, lets assume that (g) is not violated and make a note to check this assumption if the resulting design looks favorable. We can now apply the crystallize operation as shown in Figure 4.39.

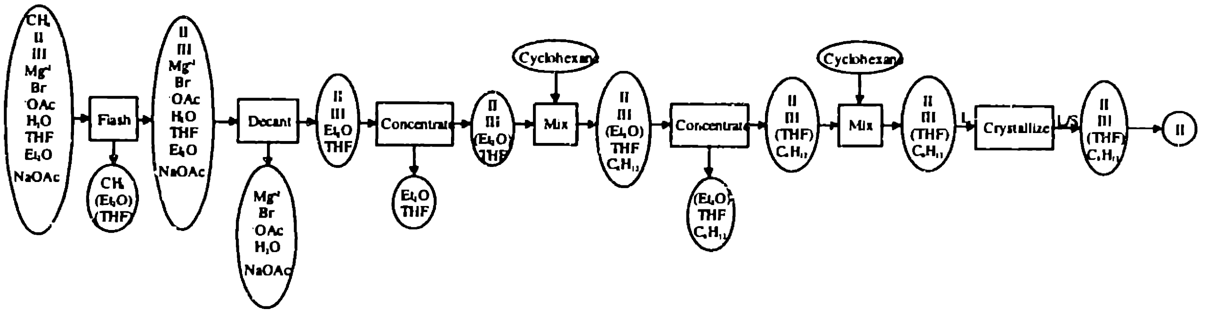


Figure 4.39 Application of the Crystallize Operation.

Now that the crystallize operation has been applied and a new current state generated, we start the MEA-NMP methodology from the beginning by comparing the new current state generated by the crystallize operation to the goal state and attempt to identify a difference. Again we have identified a concentration difference so we then we identify a task category that can resolve this difference. We note that the crystallization has given us a two phase mixture so we apply a phase separation. The task category selected is *SolidLiquidPhaseSeparation* and the primary alternative is the *filter* operation.

The filter operation is added to the plan and its preconditions are tested. No violation is found so the operation is applied and the new current state is generated. We then cycle to the beginning of the MEA-NMP methodology and perform the same procedure again; identify differences, identify a task category and select a primary alternative.

In this example the crystals are still wet with solvent after the crystallization so we identify the *LiquidfromSolid* task category and select the dry operation. The preconditions of the *dry* operation are tested and the operation is applied. We then cycle to the beginning of the MEA-NMP methodology and attempt to identify differences. We no longer have a concentration difference and we have simultaneously resolved all the differences lower in the difference hierarchy thus the synthesis of the plan with respect to the target product is complete. The complete separation system is shown in Figure 4.40

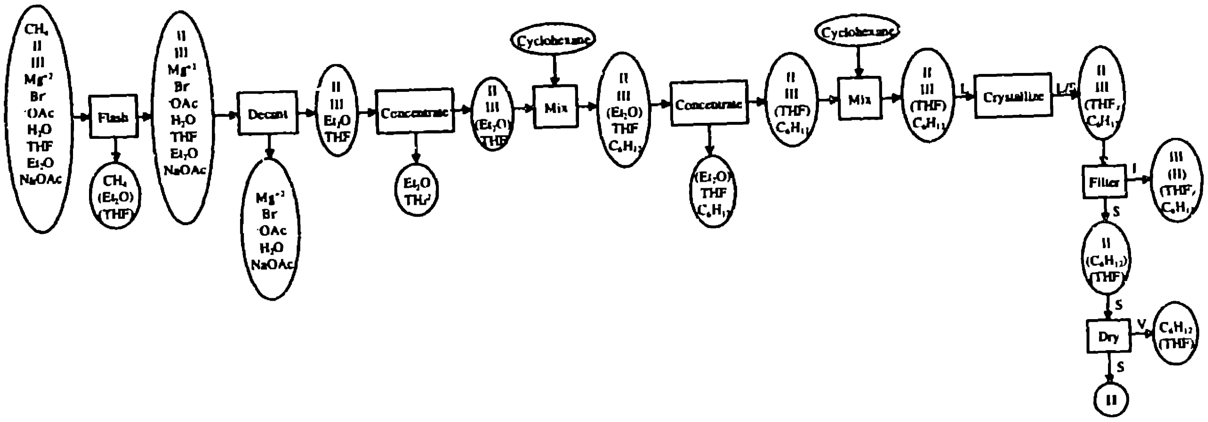


Figure 4.40 The Complete Separation Train for the Production of Carbinol.

4.6.10 The Complete Product Train

If we go back and bring together the reaction system with the separation system we then have the complete product train for the production of carbinol as shown in Figure 4.41.

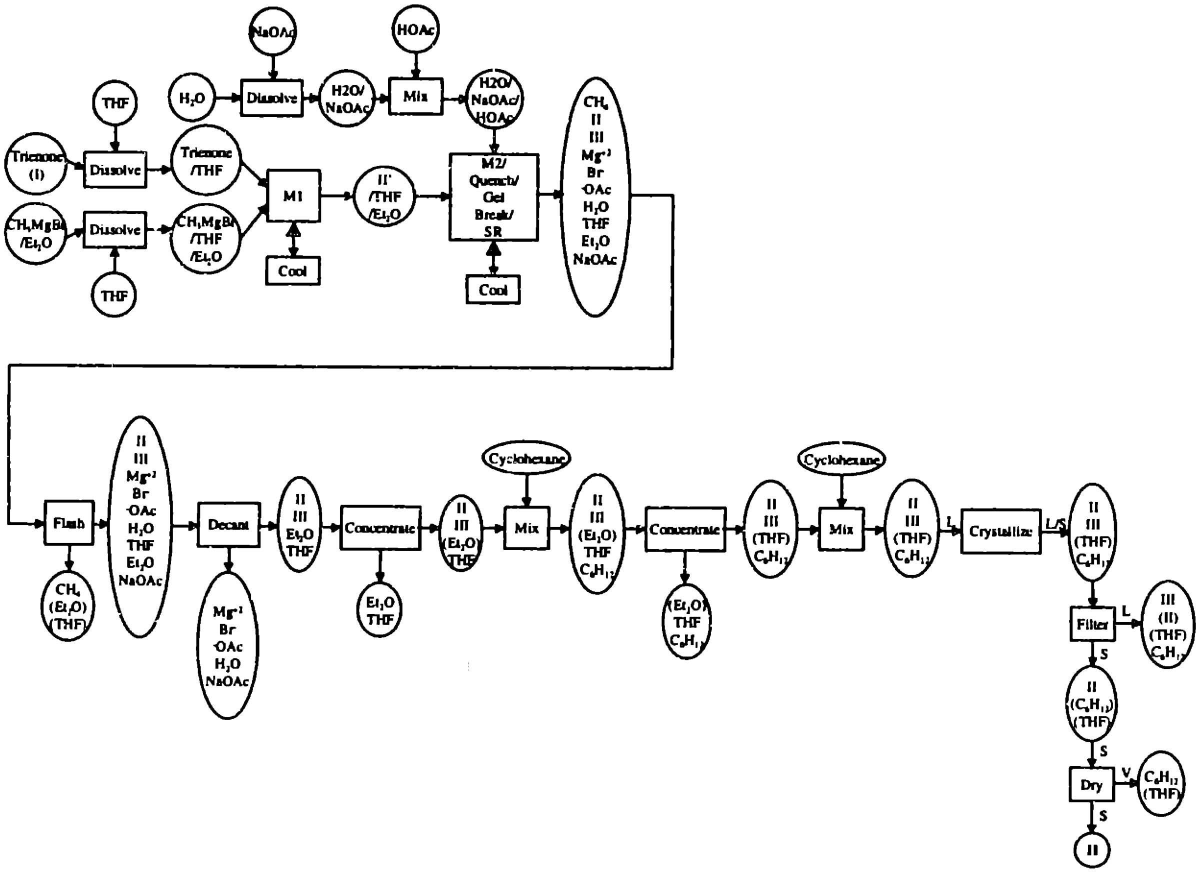


Figure 4.41 The Complete Product Train for the Production of Carbinol.

4.6.11 Waste Streams and the Complete Process Plan

Upon examination of the carbinol product train we see that the methodology has left a number of streams "dangling." These dangling streams are our process waste streams and provide us with a worst case analysis in terms of waste production for the specific process developed. Since many of the waste streams include solvents, we may want to examine the possibility of recovering these materials. To do this, we simply state a new goal state as the *production of pure solvent* and apply the MEA-NMP to each of the waste streams containing the solvent of interest. The waste material becomes the new current state and we apply the MEA-NMP as above. This can be performed for any compound of interest, not just solvent compounds. The complete process plan, including the isolation of every recoverable compound, is given in Figure 4.42. If we then assume that we recover and recycle all the non-by-product materials, this will give us the best case scenario for this process with respect to waste.

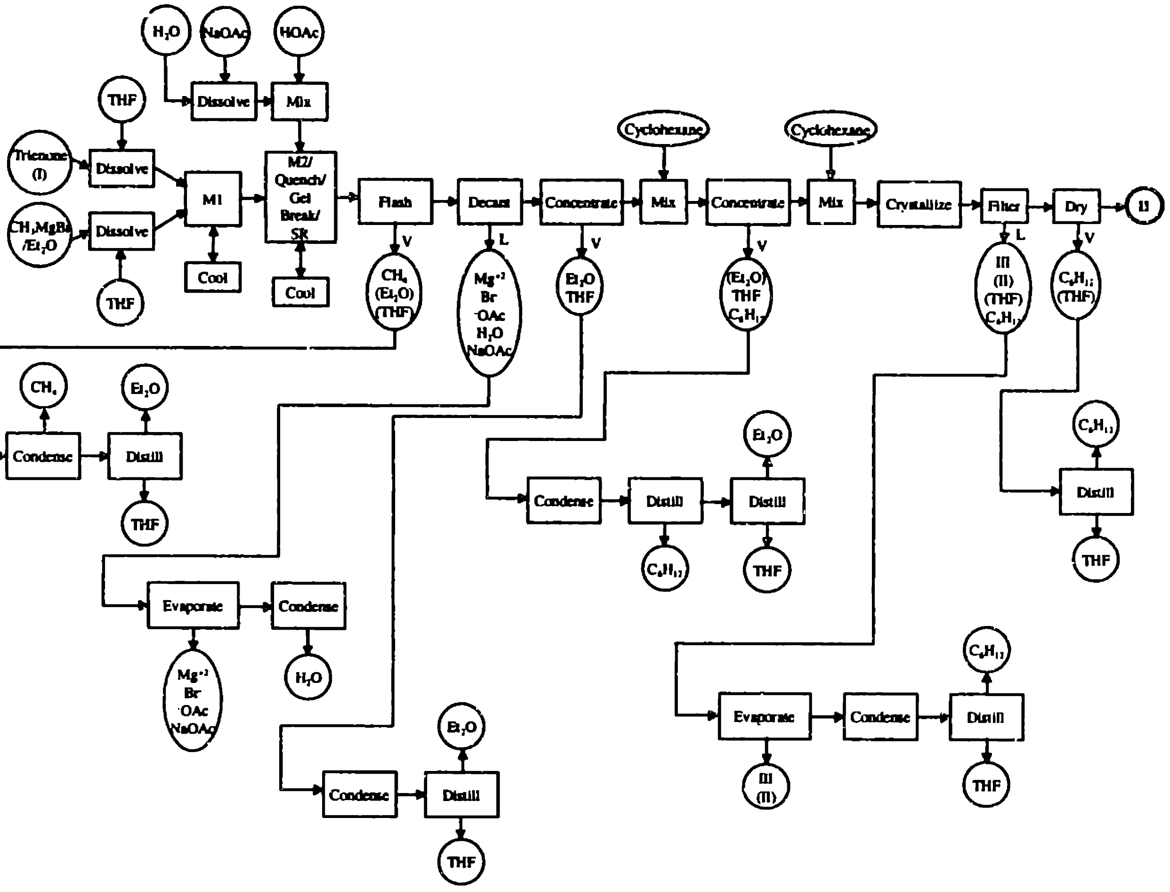


Figure 4.42 Application of MEA-NMP to All Waste Streams.

4.6.12 Stream Merging

We see from Figure 4.42 that there are a number of operations that appear to perform the same task during the material recovery. For example, the separation of Et_2O from THF so that the THF can be recycled. We can use this information to begin looking for opportunities for stream merging. In order to accomplish stream merging in a batch process, we must use intermediate storage since the streams are produced at different times during the process. In our case, we would send all the streams containing Et_2O and THF to a central intermediate storage tank and then run a single batch distillation to purify the THF. The merging of waste streams for the recovery of recyclable materials is commonly practiced within industry today, especially for processes which use large volumes of solvents.

4.6.13 MEA-NMP Process Versus the Chemists Process

Finally, lets compare the process developed using MEA-NMP to that produced by the chemist. Figure 4.43 contains the process produced by the process chemist. We see that Figure 4.43 is almost identical to Figure 4.41. There are a few differences that we should discuss.

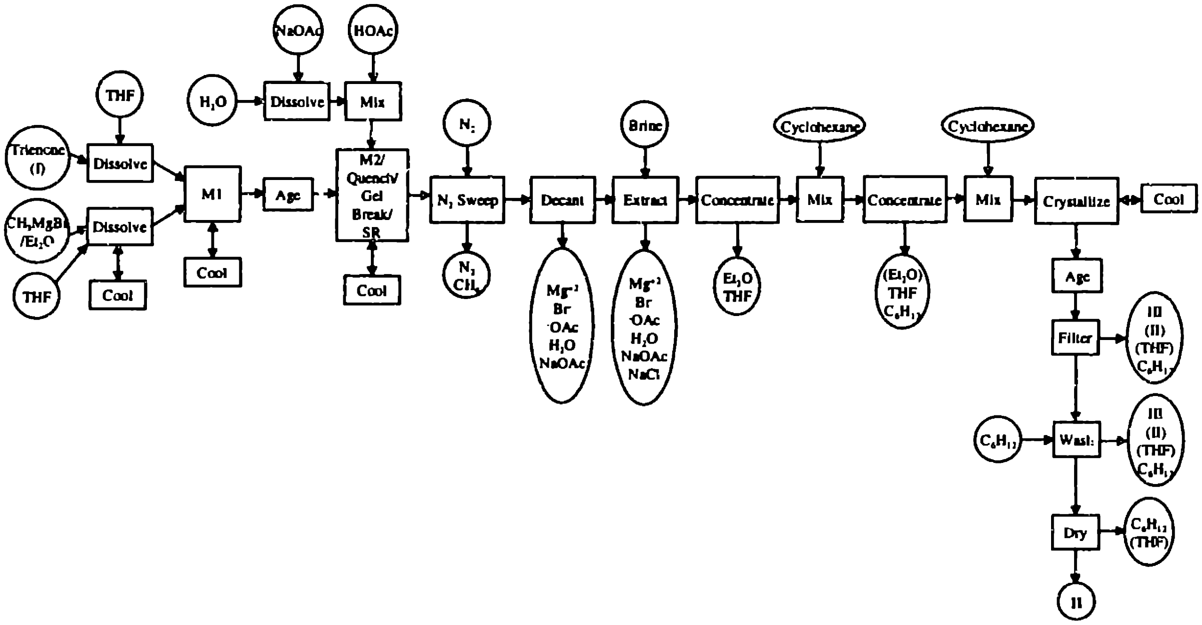


Figure 4.43 Carbinol Process Developed by the Process Chemist (Yield = 87%).

First, we see that the process developed by the chemist contains explicit *age* operations. Currently the MEA-NMP methodology has no means to place age operations explicitly into the process. The aging of certain operations is assumed to be captured by the operating conditions of specific operations. For example, reactions are assumed to require a finite amount of time and this must be specified by the developer or in the reaction model. Another example is the crystallization operation, which is again a function of the rate of crystal growth. The age must be captured by the crystallization model for it to appear in the plan. We also see that a number of cool operations are not included in the MEA-NMP plan. For example, the cool operation associated with the dissolve operation was not identified by the MEA-NMP methodology since we did not have the heat of mixing data. The cool operation associated with the crystallize operation was also not explicitly identified by the MEA-NMP methodology. This cool operation is imbedded within the crystallize model and is part of the operating procedure of the crystallize operation. We also see that the flash operation has been replace with a nitrogen sweep. This operation was not considered by the MEA-NMP because it is not included within the *GasLiquidSplit* task category. Furthermore, we see that there is an additional extraction using a brine solution. Since the decant model used provided a complete separation of the materials, the system did not attempt to identify any additional operations to complete the separation. If a more detailed model of the decant operation had been applied this would have triggered the MEA-NMP methodology to identify additional operations such as the extract operation. The same holds for the wash step associated with the filter operation. In the complete case study given in Chapter 5, we have used more detailed models in some of the stages and were able to identify operation such as the brine extraction and the wash step. So we see that the MEA-NMP is able to construct similar plans as that provided by the process chemist.

In this chapter we have described in detail the MEA-NMP methodology and its components. We have also provided an example of how the MEA-NMP methodology is applied to an industrial process. Later, in Chapter 6, we will discuss how the MEA-NMP approach can be extended to generate alternative plans through the superstructure optimization discussed in Chapter 3 and in Chapter 7 we will show how the allocation of the operations based processes into physical equipment provides opportunities for task merging.

4.7 References:

1. Siirola, J.J. and D.F. Rudd, *Computer-Aided Synthesis of Chemical Process Designs*. Industrial & Engineering Chemistry Fundamentals, 1971. **10**(3): p. 353-362.
2. Chapman, D., *Planning for Conjunctive Goals*. Artificial Intelligence, 1987. **32**: p. 333-377.
3. Douglas, J.M., *A Hierarchical Decision Procedure for Process Synthesis*. AIChE Journal, 1985. **31**: p. 353.
4. Douglas, J.M., *Conceptual Design of Chemical Processes*. McGraw-Hill Chemical Engineering Series. 1988, New York: McGraw-Hill Publishing Company. 601.
5. Siirola, J.J., *Strategic Process Synthesis: Advances in the Hierarchical Approach*. Computers & Chemical Engineering, 1996. **19 Supplement**(ESCAPE-6).
6. Russell, S.J. and P. Norvig, *Artificial Intelligence: A Modern Approach*. 1995, Englewood Cliffs, New Jersey: Prentice Hall. 932.
7. Newell, A., J.C. Shaw, and H.A. Simon, *A General Probelm-Solving Program for a Computer*. Computers and Automation, 1959(July): p. 10-17.
8. Simon, H.A., *The Sciences of the Artificial*. 1969, Cambridge, Massachusetts: The MIT Press. 231.
9. Winston, P.H., *Artificial Intelligence*. 3rd ed. 1992, Reading, Massachusetts: Addison-Wesley.
10. Lakshmanan, R., *Synthesis of Operating Procedures for Complete Chemical Plants*, in *Chemical Engineering*. 1989, Massachusetts Institute of Technology: Cambridge. p. 195.
11. Sacerdoti, E.D., *A Structure for Plans and Behavior*. 1977, New York: American Elsevier.
12. Lakshmanan, R. and G. Stephanopoulos, *Synthesis of Operating Procedures for Complete Chemical Plants - I. Hierarchical, Structured Modelling for Nonlinear Planning*. Computers & Chemical Engineering, 1988. **12**(9/10): p. 985-1002.
13. Fikes, R.E. and N.J. Nilsson, *STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving*. Artificial Intelligence, 1971. **2**: p. 189-208.
14. Linninger, A.A., S.A. Ali, and B. Harrison, *Batch Design-Kit Reference Manual*, . 1997, Laboratory for Intelligent Systems in Process Engineering: MIT, Cambridge, MA. p. 194.
15. Raman, R. and I.E. Grossmann, *Modelling and Computational Techniques for Logic Based Integer Programming*. Computers & Chemical Engineering, 1994. **18**(7): p. 563-578.

Chapter 5

Case Study

In this chapter we will present the complete Case Study III provided by a leading pharmaceutical company, which we have produced using the MEA-NMP methodology. This case study contains a process for the production of a pharmaceutical compound. The first stage of this case study was developed in detail and presented in Chapter 4. In this chapter, we have used an outline format for describing the steps in the methodology, adding explanations of the interesting elements directly within the stage. For each stage, we provide the reaction chemistry and then we go ahead and apply the MEA-NMP methodology. We state the current step of the MEA-NMP methodology that we are applying and, immediately following, we state the results of the application of the step. The details of the MEA-NMP methodology were presented in Chapter 4, including a flow diagram of the MEA-NMP methodology, the reaction structure logic and the task category selection logic. For each stage we focus only on the product train. The product train is sufficient to demonstrate the applicability of the methodology, thus we do not expand the waste streams as was done for the Carbinol stage in Chapter 4. After the completion of the case study, we provide some observations made during the development of the case study with respect to the MEA-NMP methodology.

Appendix A contains two additional pharmaceutical cases provided by a leading pharmaceutical company. We provide these cases as additional examples of typical pharmaceutical processes. These examples were developed from the actual process description provided by the process chemist. The two cases within Appendix A contain the reaction chemistry and the flowsheet of operations performed by the chemist. Case Study I contains nine stages similar to the Carbinol case developed in this Chapter. Case Study II consists of a single stage with multiple reactions.

5.1 Case Study III – Process Overview

In order to develop the Means-Ends Analysis with Nonmonotonic Planning methodology, we interacted with industrial representatives to gain insight into the current batch process development process employed by industry. As part of this interaction, we were supplied with a number of case studies representing actual industrial processes. One of the processes we examined was a case study for the development and manufacture of a pharmaceutical product. We have chosen to present this case study since it represents a typical batch process development case. The information provided to us by the company included three documents (i) the chemist's recipe (ii) the process description for use in pilot plant manufacturing and (iii) the final manufacturing batchsheets and process flowsheets. Examples of the information contained in each of these documents was given in Chapter 2.

The case study we present in this chapter, entitled Case Study III, is a multistep process for the production of a compound code named MK-VIII. This a nine stage process producing both solid and liquid phase intermediates. The final product, MK-VIII, is a solid crystalline material. Figure 5.1 contains the route diagram for the process. Each block in the diagram represents a stage in the process and the connections between the blocks represent the transfer of an intermediate from one stage to the next. The route level diagram only shows the raw materials, the intermediates and the reaction by-products, all solvents, catalysts and other materials are not shown and are assumed to be recovered and reused within the stage. The route diagram provides a convenient visual overview of the entire process. Each stage in the process can be thought of as an individual process which must be developed independently using the MEA-NMP methodology. In our development of this case study, we do not take into consideration the concept of stage integration i.e. we develop each stage as an independent chemical process and do not consider merging tasks between stages.

In this case study we do not include the operating conditions. The reader can assume the reactions take place at ambient conditions and that all cooling and heating operations are relative to ambient conditions. All reactions are liquid phase reactions and the yields given are those reported in the chemist's recipe.

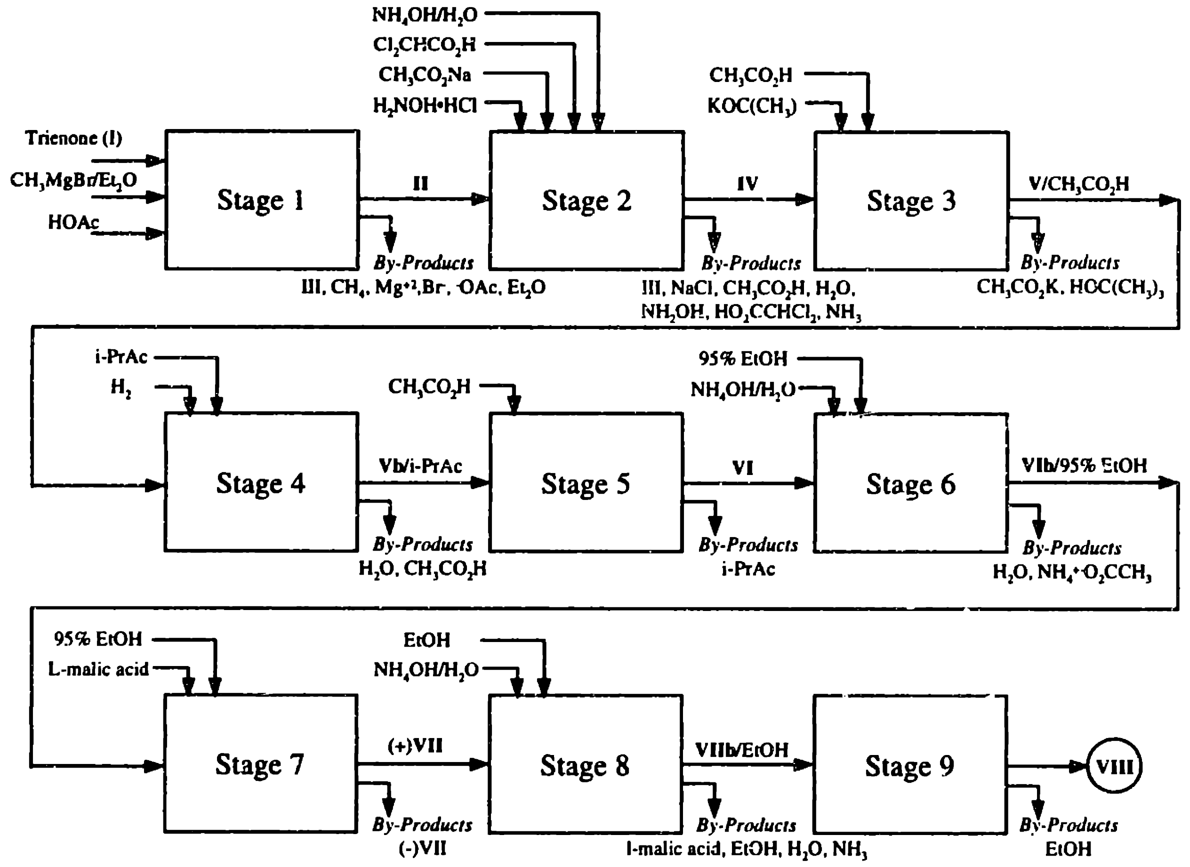


Figure 5.1 Route Diagram for the MK-VIII Process

5.2 Stage 1

As we have stated in Chapter 4, we begin the process synthesis using the information provided by the chemist in the form of a synthesis recipe. For now it will be sufficient to present the reaction information and if any additional information is required we will note it at the appropriate time. Figure 5.2 contains the reaction chemistry for the Carbinol Stage of the process we are developing. The chemist breaks the process into stages based on the isolation of a product or an intermediate. We have described the Carbinol Stage in detail in Chapter 4, so let us now go ahead and apply the MEA-NMP methodology to the Carbinol Stage in order to synthesize a process. We will use the Carbinol Stage to introduce the reduced description structure for the case study and we will also develop the superstructure of process alternatives for this example, which was not done in Chapter 4.

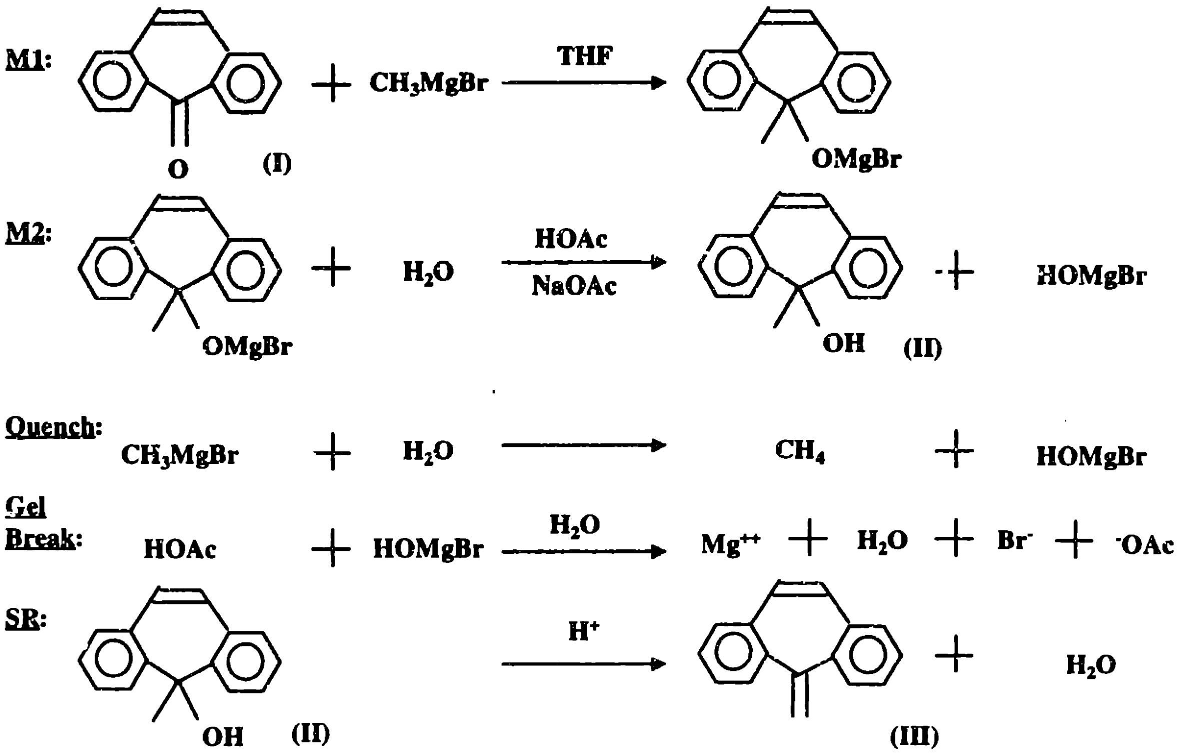


Figure 5.2 Carbinol Stage Reaction Chemistry

After the development of the input/output structure (not shown) and the grouping of the reactions into reaction sets, we can then go ahead and apply the MEA-NMP methodology.

Apply MEA-NMP

A) Creation of Reaction Set 1

➢ Identify Difference: *Difference in Chemical Identity*

– Screen for Critical Features: *None*

- Identify Task Category: *Molecular Structure Transformation*
- Identify Primary Alternative: *React → Select Reaction Set*: M1
  - Evaluate Preconditions: *React →Dynamic Precondition Identification*
  - Reaction Logic to Eliminate Missing Reactants Precondition Violation
    1) Use Reaction Set Information to Define:

- a) Reactants: I, CH_3MgBr/Et_2O
- b) Products: II'
- c) Catalysts: None
- d) Buffers: None
- e) Diluents: None
- f) Solvents: THF

2) Screen the reaction set for side reactions → *None*

3) Catalyst → *None*

4) Screen reactants for solid or liquid reactants – For each S or L reactant

   I → Solid →

   a) *Dissolve* or *Mix* with solvent?
      - *Dissolve* → *Select Solvent* → THF
      - Generate partial plan → PP1: *Charge* THF, I and *Dissolve*.

   b) Eliminate I from above.

   CH_3MgBr/Et_2O → Liquid →

   a) *Dissolve* or *Mix* with solvent?
      - *Mix* → *Select Solvent* → THF
      - Generate partial plan → PP2: *Charge* THF, I and *Mix*.

   b) Eliminate CH_3MgBr/Et_2O from above.

5) Screen reactants for gaseous reactants → *None*

6) Screen for any Materials that have not been Charged → *None*

7) Specify ordering of addition of all the partial plans created above.

   PP1: → *2*

   PP2: → *1*

   a) Combine? → *No*

   b) Generate partial plan →

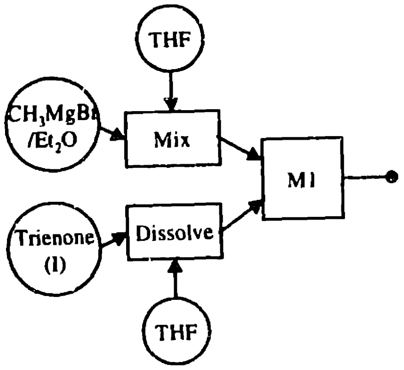


Figure 5.3 Stage1 Reaction Set 1 Partial Plan

8) Check preconditions on the pre-react operations → *OK* → *Apply*

- ☐ Evaluate Preconditions: *React* → Check heat of reaction : *Exothermic* →
  - ◆ *Cool Reaction?* → *Yes*
  - ◆ Evaluate Preconditions: *Cool* → *No Violation*
  - ◆ Apply Operation and Generate New Current State: *Cool* (M1)
- ☐ Evaluate Preconditions: *React* → *No Violation*
- Apply Operation and Generate New Current State: *React* (M1)

B) Creation of Reaction Set 2

➢ Identify Difference: *Difference in Chemical Identity*

– Screen for Critical Features: *None*

- Identify Task Category: *Molecular Structure Transformation*
- Identify Primary Alternative: *React* → *Select Reaction Set*: M2/Quench/Gel Break/SR (RS2)
  - ☐ Evaluate Preconditions: *React* →*Dynamic Precondition Identification*
  - ☐ Reaction Logic to Eliminate Missing Reactants Precondition Violation

1) Use Reaction Set Information to Define:
    a) Reactants: $RS1^{out}$, H_2O, HOAc
    b) Products: II,CH_4, Mg^{+2}, Br^-, ^-OAc, III
    c) Catalysts: HOAc
    d) Buffers: NaOAc
    e) Diluents: None
    f) Solvents: H_2O
2) Screen the reaction set for side reactions
    a) Premix HOAc with a solvent?
        i) *Yes* → *Select Solvent* → H_2O → Generate partial plan → PP1: *Charge* H_2O, HOAc and *Mix*.
    b) Add buffer to solution?

        i) *Yes* → *Dissolve*? → *Yes* → *Select Solvent* → H_2O → Generate partial plan → PP2: *Charge* H_2O, NaOAc and *Dissolve.*

    c) Eliminate HOAc and NaOAc from consideration list.

3) Catalyst → *None*

4) Screen reactants for solid or liquid reactants – For each S or L reactant

    $RS1^{out}$ → Liquid →

    a) *Mix* with solvent?

        i) *No* → Generate partial plan → PP3: *Charge* $RS1^{out}$.

    b) Eliminate $RS1^{out}$ from above.

5) Screen reactants for gaseous reactants → *None*

6) Screen for any Materials that have not been Charged → *None*

7) Specify ordering of addition of all the partial plans created above.

    PP1: → *2*

    PP2: → *1*

    PP3: → *3*

    a) Combine PP1, PP2? → *Yes*

    b) Generate partial plan →

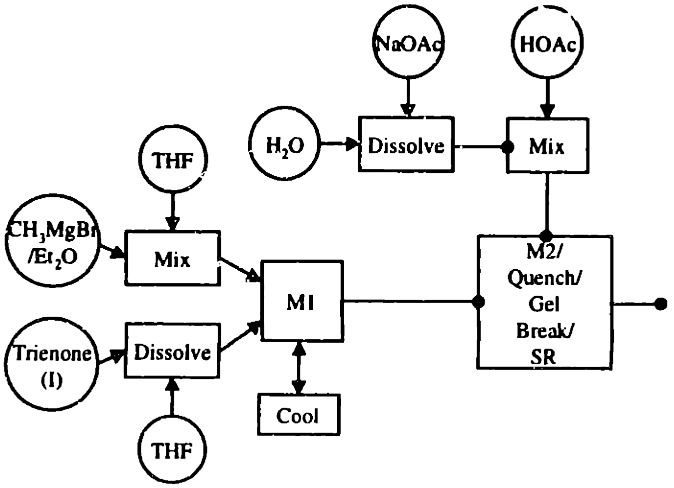


Figure 5.4 Stage1 Reaction Set 2 Partial Plan

8) Check preconditions on the pre-react operations → *OK* → *Apply*

    ☐ Evaluate Preconditions: *React*(RS2)→ Check heat of reaction : *Exothermic* →

◆ *Cool Reaction*? → *Yes*

- ◆ Evaluate Preconditions: *Cool* → *No Violation*
- ◆ Apply Operation and Generate New Current State: *Cool* (RS2)
    - □ Evaluate Preconditions: *React* (RS2) → *No Violation*
  - ● Apply Operation and Generate New Current State: *React* (RS2)

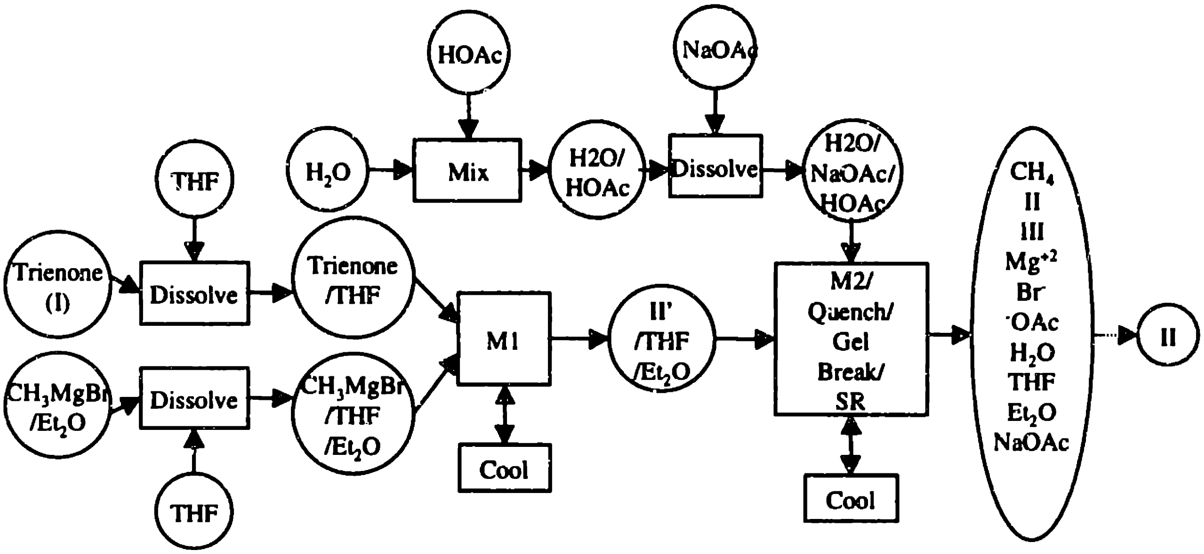


Figure 5.5 The Complete Carbinol Reaction System

C) <u>Creation of the Separation System</u>

- ➢ Identify Differences: *Difference in Concentration*
  - – Screen for Critical Features: *None*

Target Product Compound

- ● Identify Task Category: *LiquidtoSolidPhaseChange*
- ● Identify Primary Alternative: *Crystallize*
  - □ Evaluate Preconditions: *Crystallize*

a) No vapor phase as input → *Violation*

b) No multiple liquid phase as input → *Violation*

c) No solid phase as input → *OK*

d) The composition of the feed mixture must lie within a region of the solid-liquid equilibrium that allows the desired component to form pure crystals. → *OK*

e) The melting point of the solvent must be less than the melting point of the solute → *OK*

f) The concentration of the solute must be greater than the solubility of the solute → ?

g) Solubility of the product must be less than the solubility of all other components → ?

- Identify White Knight Task Category: Precond. (a) → *GasLiquidPhaseSplit*
- Identify Primary Alternative: *Flash*
  - Evaluate Preconditions: *Flash*
    - a) $K_i \geq 10$ for at least one component or vapor phase exists → *OK*
    - b) $K_i \leq 0.1$ for at least one component or liquid phase exists → *OK*

- Apply Operation and Generate New Current State: *Flash* (remove gas phase)
  - Evaluate Preconditions: *Crystallize* → *Violation* (b)
    - Identify White Knight Task Category: *LiquidLiquidPhaseSplit*
    - Identify Primary Alternative: *Decant*
      - Evaluate Preconditions: *Decant*
        - a) No gas phase input → *OK*
        - b) Two phase liquid input → *OK*
        - c) Volume of phase 1 ≥ 0.1*Volume of phase 2 (phase 1 being the lesser volume phase) → *OK*

- Apply and Generate New Current State: *Decant* (remove non-product containing phase)

- Evaluate Preconditions: *Crystallize* → (f) collect solubility info. → *Violation*
- Identify White Knight Task Category: *Solubilty Reduction* → *Solvent Replacement*
- Select Primary Alternative: *Concentrate and Mix*
    - Evaluate Preconditions: *Concentrate*
        - a) Single phase liquid → *OK*
        - b) $P^{vap}_{solvent} > P^{vap}_{solute}$ → *OK*
  - Apply and Generate New Current State: *Concentrate* (remove THF/Et_2O)
- Evaluate Preconditions: *Mix*
        - a) Number of inputs ≥ 2 → *Violation*
    - Identify White Knight Task Category: *Material Transfer*
    - Identify Primary Alternative: *Charge* → Select Material: Cyclohexane
        - Evaluate Preconditions: *Charge* → *None*
- Apply and Generate New Current State: *Charge* (supply cyclohexane for *Mix*)
- Evaluate Preconditions: *Mix* → *OK*
  - Apply and Generate New Current State: *Mix*

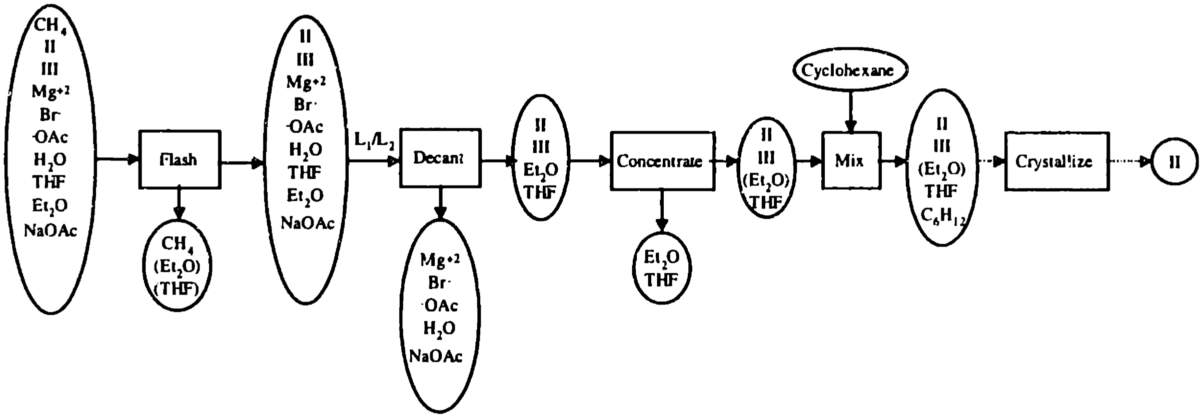


Figure 5.6 Application of the Mix Operation

- Evaluate Preconditions: *Crystallize* → (f) → *Violation*
  - Identify White Knight Task Category: *Solubilty Reduction* → *Solvent Replacement*
  - Select Primary Alternative: *Concentrate and Mix*
    - Evaluate Preconditions: *Concentrate*
      - a) Single phase liquid → *OK*
      - b) $P^{vap}_{solvent} > P^{vap}_{solute}$ → *OK*
  - Apply and Generate New Current State: *Concentrate* (remove Cyclohexane/THF/Et_2O)
    - Evaluate Preconditions: *Mix*
      - a) Number of inputs ≥ 2 → *Violation*
    - Identify White Knight Task Category: *Material Transfer*
    - Identify Primary Alternative: *Charge* → Select Material: Cyclohexane
      - Evaluate Preconditions: *Charge* → *None*
    - Apply and Generate New Current State: *Charge* (supply cyclohexane for *Mix*)
    - Evaluate Preconditions: *Mix* → *OK*
  - Apply and Generate New Current State: *Mix*
- Evaluate Preconditions: *Crystallize* → *OK*
- Apply and Generate New Current State: *Crystallize* (crystals in solution)
- Identify Difference: *Difference in Concentration*
  - Screen for Critical Features: *None*
    - Identify Task Category: PhaseSeparation → *SolidLiquidPhaseSeparation*

  - Identify Primary Alternative: *Filter*
    - ❑ Evaluate Preconditions: *Filter*
      - a) Input contains solid and liquid phases → *OK*
  - Apply and Generate New Current State: *Filter* (remove liquid phase)
- ➢ Identify Difference: *Concentration*
  - – Screen for Critical Features: *None*
  - Identify Task Category: *LiquidfromSolid*
  - Identify Primary Alternative: *Dry*
    - ❑ Evaluate Preconditions: *Dry*
      - a) Input contains solid and liquid phases → *OK*
  - Apply and Generate New Current State: *Dry*
- ➢ Identify Difference: *None*
- ➢ Check for Abstract Operators: *None*

Stage 1 Complete

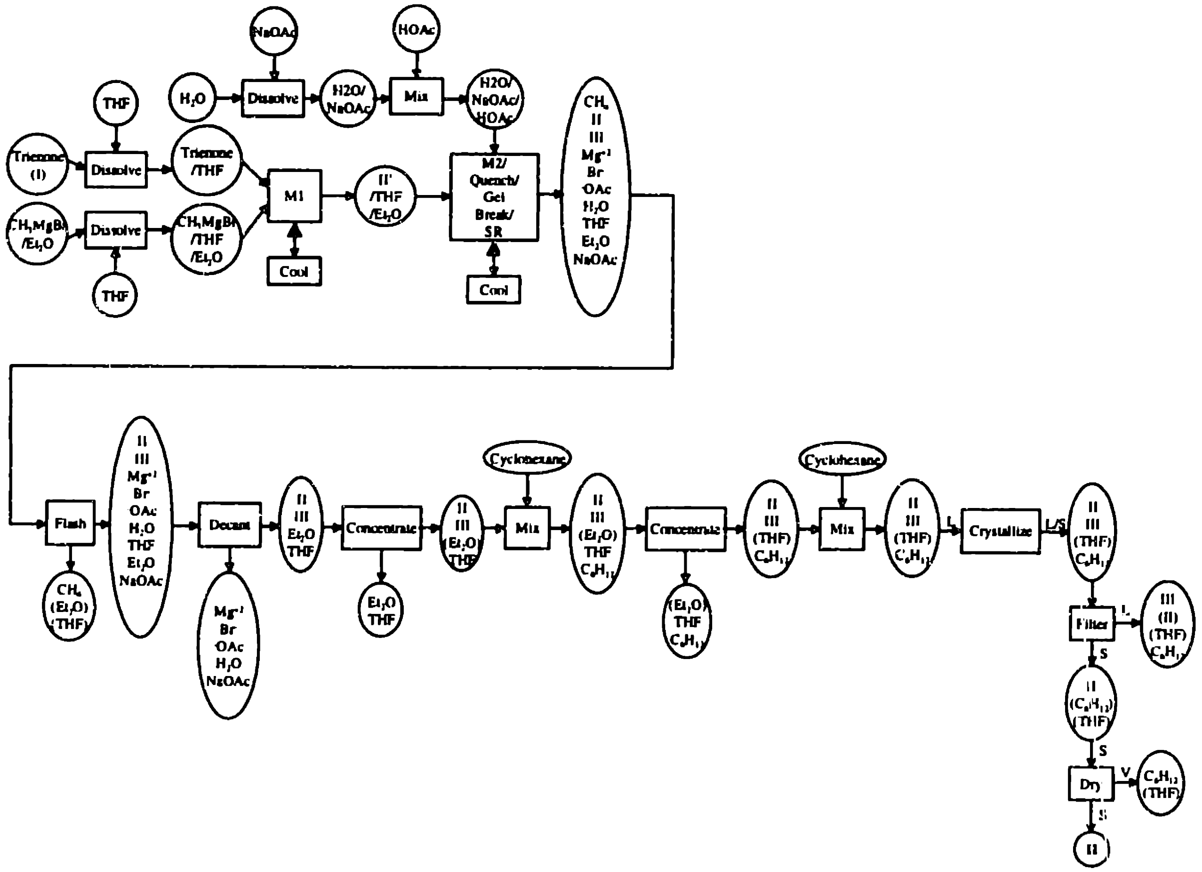


Figure 5.7 The Complete Carbinol Product Train (Yield = 87%)

From the base case developed in Figure 5.7 we can extract the superstructure of task categories. It should be clear from the superstructure that we can generate a number of alternative processes.

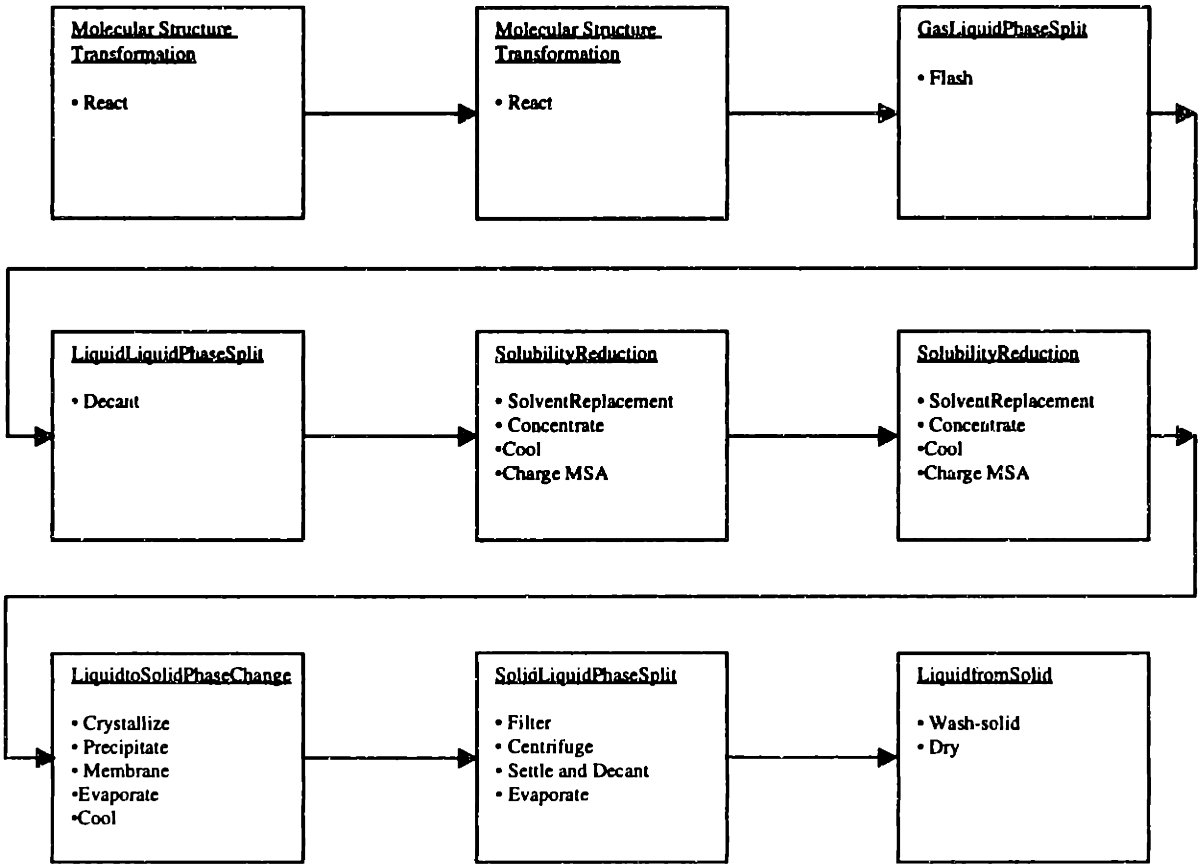


Figure 5.8 Stage 1 Task Category Superstructure

5.3 Stage 2

The key raw material for Stage 2 is the Carbinol produced in Stage 1. Again, we begin with the reaction provided by the chemist. The reactions have been grouped into reaction sets based on the operating conditions at which the reactions take place. In this Stage we will demonstrate how the user can override the methodology and insert operations directly into the plan. This is especially important when using simplified models. For example, if we use a simple model for the reaction operation which does not include reaction rate information we will be unable to identify the necessary aging operation in order to ensure the reaction is complete. The same could be true for operations that may require longer time scales such as crystallization or Decantation.

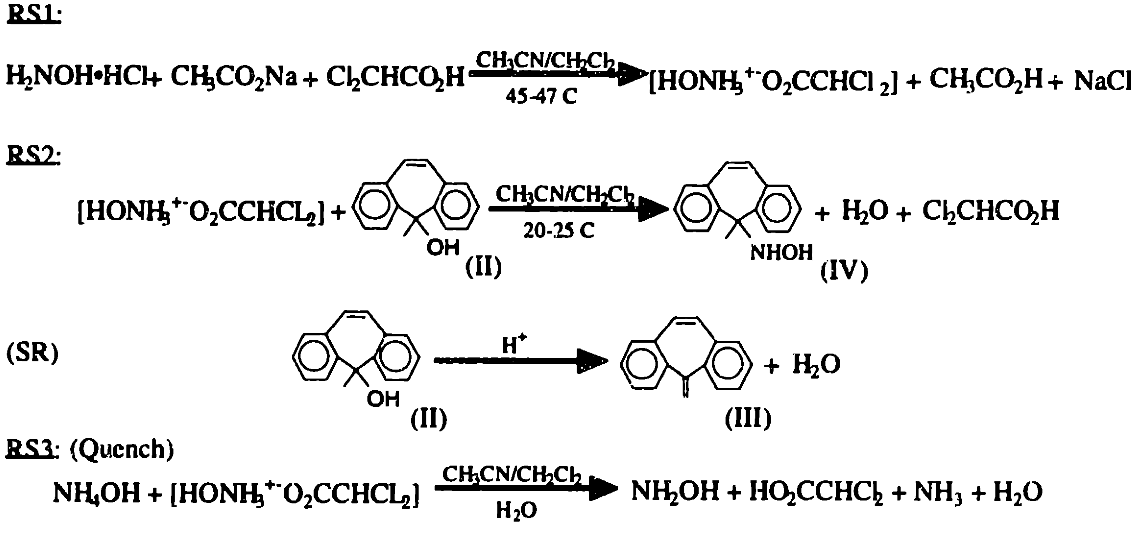


Figure 5.9 Stage 2 Reaction Chemisty

Apply MEA-NMP

A) Creation of Reaction Set 1

➤ Identify Difference: *Difference in Chemical Identity*

- Screen for Critical Features: *None*

  - Identify Task Category: *Molecular Structure Transformation*
  - Identify Primary Alternative: *React → Select Reaction Set*: RS1

☐ Evaluate Preconditions: *React →Dynamic Precondition Identification*

☐ Reaction Logic to Eliminate Missing Reactants Precondition *Violation*

1) Use Reaction Set Information to Define:
   a) Reactants: $H_2NOH{\bullet}HCl$, CH_3CO_2Na, Cl_2CHCO_2H
   b) Products: $[HONH_3^{+-}O_2CCHCl_2]$
   c) Catalysts: None
   d) Buffers: None
   e) Diluents: None
   f) Solvents: CH_3CN,CH_2Cl_2

2) Screen the reaction set for side reactions → *None*

3) Catalyst → *None*

4) Screen reactants for solid or liquid reactants – For each S or L reactant

   $H_2NOH \bullet HCl$ → Solid →

   a) *Dissolve* or *Mix* with solvent?

      - *Mix* → *Select Solvent* → CH_3CN,CH_2Cl_2
      - Generate partial plan → PP1: *Charge* CH_3CN and CH_2Cl_2, *Mix*, *Charge* $H_2NOH \bullet HCl$ and *Mix*.

   b) Eliminate $H_2NOH \bullet HCl$ from above.

   CH_3CO_2Na → Solid →

   a) *Dissolve* or *Mix* with solvent?

      - *Mix* → *Select Solvent* → CH_3CN,CH_2Cl_2
      - Generate partial plan → PP2: *Charge* CH_3CN and CH_2Cl_2, *Mix*, *Charge* CH_3CO_2Na and *Mix*.

   b) Eliminate CH_3CO_2Na from above.

   Cl_2CHCO_2H → Liquid →

   a) *Mix* with solvent?

      - No
      - Generate partial plan → PP3: *Charge* Cl_2CHCO_2H

   b) Eliminate Cl_2CHCO_2H from above.

5) Screen reactants for gaseous reactants → *None*

6) Screen for any Materials that have not been Charged → *None*

7) Specify ordering of addition of all the partial plans created above.

   PP1: → *1*

   PP2: → *2*

   PP3: → *3*

   a) Combine PP1 and PP2? → *Yes*

   b) Generate partial plan →

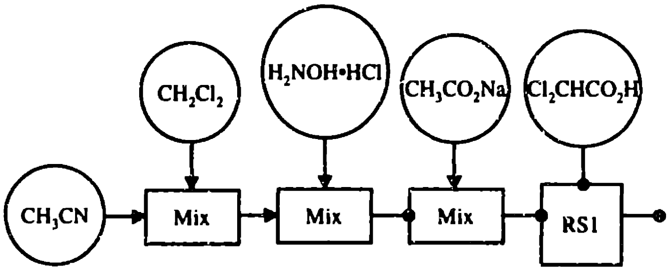


Figure 5.10 Stage2 Reaction Set 1 Partial Plan

8) Check preconditions on the pre-react operations → *OK* → *Apply*

□ Evaluate Preconditions: *React* → Check heat of reaction → *Exothermic* →

◆ *Cool Reaction*? → *YES*

⅄ Evaluate Preconditions: *Cool* → *OK*

◆ Apply Operation and Generate New Current State: *Cool*: RS1 (25 C)

□ Evaluate Preconditions: *React* → *OK*

- Apply Operation and Generate New Current State: *React*: RS1

B) Creation of Reaction Set 2

➢ Identify Difference: *Difference in Chemical Identity*

– Screen for Critical Features: *None*

- Identify Task Category: *Molecular Structure Transformation*
- Identify Primary Alternative: *React* → *Select Reaction Set*: RS2

□ Evaluate Preconditions: *React* →*Dynamic Precondition Identification*

□ Reaction Logic to Eliminate Missing Reactants Precondition Violation

1) Use Reaction Set Information to Define:
   a) Reactants: RS1^{out}, II
   b) Products: IV
   c) Catalysts: None
   d) Buffers: None
   e) Diluents: None
   f) Solvents: CH_3CN,CH_2Cl_2

2) Screen the reaction set for side reactions

    a) Premix II with a solvent?

        - *Yes* → *Select Solvent* → CH_3CN, CH_2Cl_2 → Generate partial plan → PP1: *Charge* CH_3CN and CH_2Cl_2, *Mix*, *Charge* $H_2NOH \bullet HCl$ and *Mix*.

    b) Add buffer to solution?

        - *No*

    c) Eliminate II from consideration list.

3) Catalyst → *None*

4) Screen reactants for solid or liquid reactants For each S or L reactant

    $RS1^{out}$ → Liquid →

    a) *Mix* with solvent?

        - *No*
        - Generate partial plan → PP2: *Charge* Cl_2CHCO_2H

    b) Eliminate $RS1^{out}$ from above.

5) Screen reactants for gaseous reactants → *None*

6) Screen for any Materials that have not been Charged → *None*

7) Specify ordering of addition of all the partial plans created above.

    PP1: → *2*

    PP2: → *1*

    a) Combine PP1 and PP2? → *No*

    b) Generate partial plan →

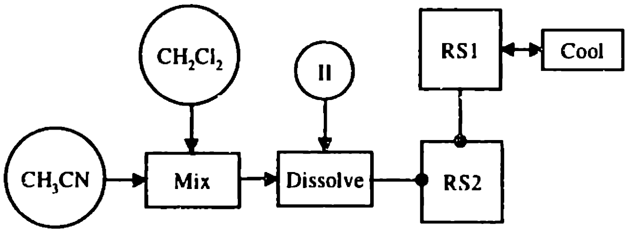


Figure 5.11 Stage2 Reaction Set 2 Partial Plan

8) Check preconditions on the pre-react operations → *OK* → *Apply*

☐ Evaluate Preconditions: *React* →Check heat of reaction → *Exothermic* →

- *Cool Reaction*? → *Yes*

- Evaluate Preconditions: *Cool → OK*
    - Apply Operation and Generate New Current State: *Cool*: RS2 (25 C)
  - Evaluate Preconditions: *React → OK*
- Apply Operation and Generate New Current State: React: RS2
- *User Override: Add Heat operation to complete reaction (45-47 C)*
- *User Override: Add Age operation to complete reaction (3-4hr)*
- *User Override: Add Cool operation to complete reaction (15-20 C)*

C) <u>Creation of Reaction Set 3</u>

- Identify Difference: *Difference in Chemical Identity*
  - Screen for Critical Features: *None*
- Identify Task Category: *Molecular Structure Transformation*
- Identify Primary Alternative: *React → Select Reaction Set*: RS3
  - Evaluate Preconditions: *React →Dynamic Precondition Identification*
  - Reaction Logic to Eliminate Missing Reactants Precondition Violation

1) Use Reaction Set Information to Define:
    a) Reactants: $RS2^{out}$, NH_4OH/H_2O
    b) Products: IV
    c) Catalysts: None
    d) Buffers: None
    e) Diluents: None
    f) Solvents: CH_3CN, CH_2Cl_2, H_2O
2) Screen the reaction set for side reactions → *None*
3) Catalyst → *None*
4) Screen reactants for solid or liquid reactants – For each S or L reactant $RS2^{out}$ → Liquid →
    a) *Mix* with solvent?

- No
- Generate partial plan → PP1: *Charge* RS2^{out}

b) Eliminate RS2^{out} from above.

NH_4OH/H_2O → Liquid →

a) *Mix* with solvent?

- No
- Generate partial plan → PP2: *Charge* NH_4OH/H_2O

b) Eliminate NH_4OH/H_2O from above.

5) Screen reactants for gaseous reactants → *None*
6) Screen for any Materials that have not been Charged → *None*
7) Specify ordering of addition of all the partial plans created above.

PP1: → *1*

PP2: → *2*

a) Generate partial plan →

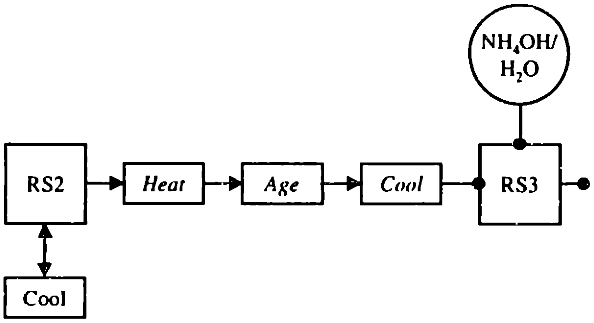


Figure 5.12 Stage2 Reaction Set 3 Partial Plan

8) Check preconditions on the pre-react operations → *OK* → *Apply*

☐ *Dynamic Precondition Identification*: Check heat of reaction → *Exothermic* →

- *Cool Reaction*? → *Yes*
  - Evaluate Preconditions: *Cool* → *OK*
- Apply Operation and Generate New Current State: *Cool*: RS3 (15-20C)

☐ Evaluate Preconditions: *React* → *OK*

- Apply Operation and Generate New Current State: *React*: RS3

- *User Override: Add Age operation to complete reaction (3-4hr)*

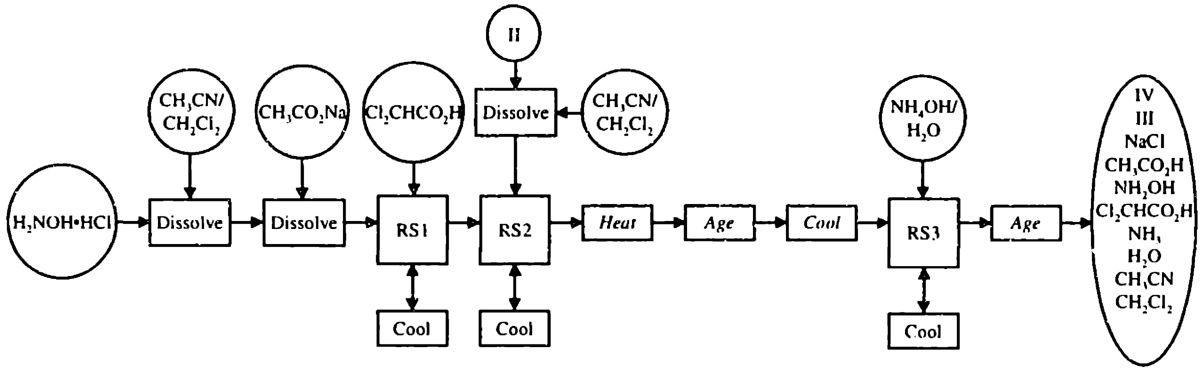


Figure 5.13 Stage 2 Reaction Structure

D) Creation of Separation System

- Identify Differences: *Difference in Concentration*
  - Screen for Critical Features: *None*

Target Product Compound

- Identify Task Category: *LiquidtoSolidPhaseChange*
- Identify Primary Alternative: *Crystallize*
  - Evaluate Preconditions: *Crystallize*

a) No vapor phase as input →OK

b) No multiple liquid phase as input → *Violation*

c) No solid phase as input → *OK*

d) The composition of the feed mixture must lie within a region of the solid-liquid equilibrium that allows the desired component to form pure crystals. → ?

e) The melting point of the solvent must be less than the melting point of the solute → *OK*

f) The concentration of the solute must be greater than the solubility of the solute → ?

g) Solubility of the product must be less than the solubility of all other components → ?

- Identify White Knight Task Category: Precond. (b) → *LiquidLiquidPhaseSplit*
- Identify Primary Alternative: *Decant*
  - Evaluate Preconditions: *Decant* →
    - a) No gas phase input → *OK*
    - b) Two phase liquid input → *OK*
    - c) Volume of phase 1 $\geq 0.1*$Volume of phase 2 (phase 1 being the lesser volume phase) → *OK*
- Apply and Generate New Current State: *Decant* (remove non-product containing phase)

☐ Evaluate Preconditions: *Crystallize* → (g) collect solubility info. → *Violation*(III)

- Identify White Knight Task Category: *Dissolved*MaterialRemoval
- Select Primary Alternative: *Extract* → Select Extracting Liquid → Brine
  - Evaluate Preconditions: Extract
    - a) liquid input → *OK*
    - b) MSA present? → *OK*
    - c) $S_{\text{solute i in MSA}} > 0$ → *OK*
    - d) $S_{\text{solute j in MSA}} << S_{\text{solute i in MSA}}$ → *OK*
    - e) $S_{\text{MSA in Solvent}} \cong 0$ → *OK*
- Apply and Generate New Current State: *Extract* (remove brine phase containing III)
  - Evaluate Preconditions: *Crystallize*

(g) → *OK*

(f) collect solubility info. → *Violation*

- ✱ Identify White Knight Task Category: *Solubilty Reduction → Solvent Replacement*
- ✱ Select Primary Alternative: *Concentrate and Mix*
  - ◊ Evaluate Preconditions: *Concentrate*
    - a) Single phase liquid → *OK*
    - b) $P^{vap}_{solvent} > P^{vap}_{solute}$ → *OK*
- ✱ Apply and Generate New Current State: *Concentrate* (remove CH_2Cl_2)
  - ◊ Evaluate Preconditions: *Mix*
    - a) Number of inputs ≥ 2 → *Violation*
    - ∇ Identify White Knight Task Category: *Material Transfer*
    - ∇ Identify Primary Alternative: *Charge* → Select Material: Cyclohexane
      - ~ Evaluate Preconditions: *Charge* → *None*
    - ∇ Apply and Generate New Current State: *Charge* (supply cyclohexane for *Mix*)
  - ◊ Evaluate Preconditions: *Mix* → *OK*
- ✱ Apply and Generate New Current State: *Mix*

☐ Evaluate Preconditions: *Crystallize*

(f) collect solubility info. → *Violation*

- ◆ Identify White Knight Task Category: *Solubilty Reduction → Solvent Replacement*

        - ◆ Select Primary Alternative: *Concentrate and Mix*
          - ⅄ Evaluate Preconditions: *Concentrate*
            - a) Single phase liquid → *OK*
            - b) $P^{vap}_{solvent} > P^{vap}_{solute}$ → *OK*
        - ◆ Apply and Generate New Current State: *Concentrate* (remove cyclohexane/CH_2Cl_2)
          - ⅄ Evaluate Preconditions: *Mix*
            - a) Number of inputs ≥ 2 → *Violation*
            - ✱ Identify White Knight Task Category: *Material Transfer*
            - ✱ Identify Primary Alternative: *Charge* → Select Material: Cyclohexane
              - ◊ Evaluate Preconditions: *Charge* → *None*
            - ✱ Apply and Generate New Current State: *Charge* (supply cyclohexane for *Mix*)
          - ⅄ Evaluate Preconditions: *Mix* → *OK*
        - ◆ Apply and Generate New Current State: *Mix*
      - ☐ Evaluate Preconditions: *Crystallize* → (d) → collect SLE info. → unavailable → assume *OK* and continue
    - • Apply and Generate New Current State: *Crystallize* (crystals in solution)
- ➢ Identify Difference: *Difference in Concentration*
  - – Screen for Critical Features: *None*
    - • Identify Task Category: PhaseSeparation → *SolidLiquidPhaseSeparation*
    - • Identify Primary Alternative: *Filter*
      - ☐ Evaluate Preconditions: *Filter*
        - a) Input contains solid and liquid phases → *OK*

- Apply and Generate New Current State: *Filter* (remove liquid phase)

- ➢ Identify Difference: *Difference in Concentration*
  - – Screen for Critical Features: *None*
  - Identify Task Category: *LiquidfromSolid*
  - Identify Primary Alternative: *Wash-Solid* → Select MSA → cyclohexane
    - ☐ Evaluate Preconditions: *Wash-Solid*
      - a) Solubility of solid in MSA = low → *OK*
  - Apply and Generate New Current State: *Wash-Solid*
- ➢ Identify Difference: *Difference in Concentration*
  - – Screen for Critical Features: *None*
  - Identify Task Category: *LiquidfromSolid*
  - Identify Primary Alternative: *Dry*
    - ☐ Evaluate Preconditions: *Dry*
      - a) Input contains solid and liquid phases → *OK*
  - Apply and Generate New Current State: *Dry*
- ➢ Identify Difference: *None*
- ➢ Check for Abstract Operators: *None*

Stage 2 Complete

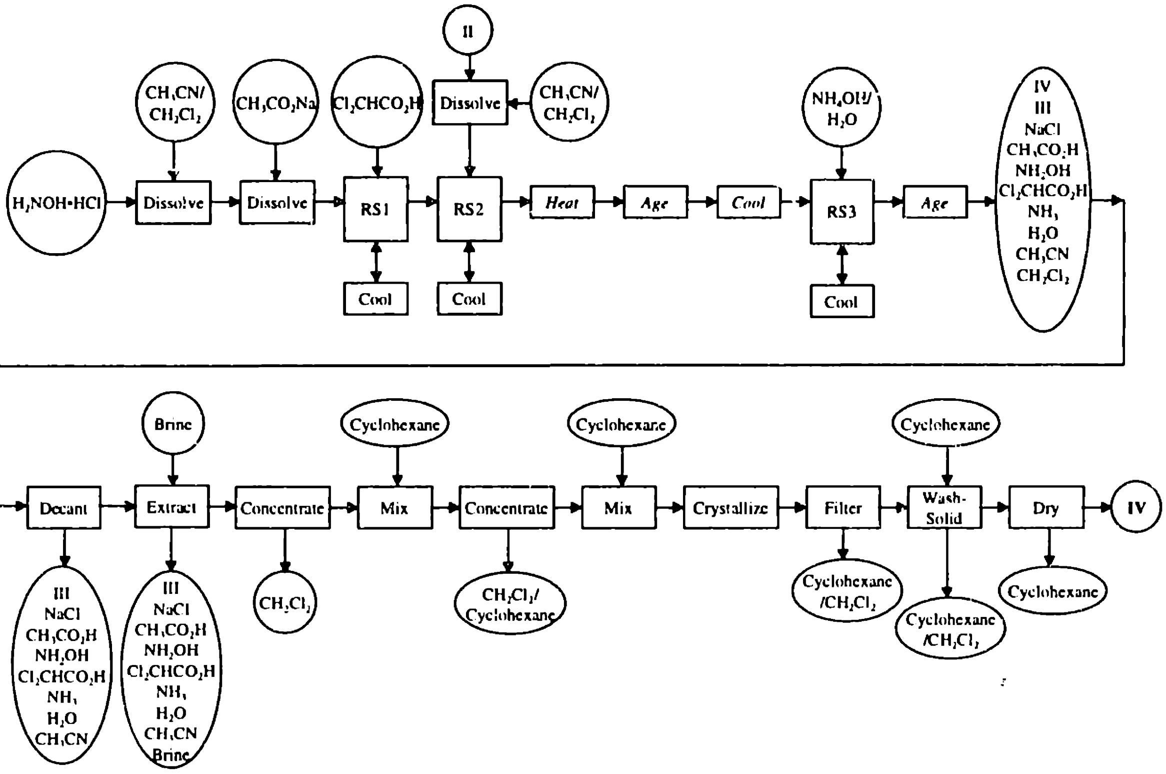


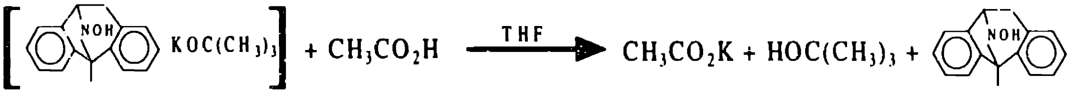
Figure 5.14 Stage 2 Complete Product Train (Yield = 93%)

5.4 Stage 3

RS1:

(IV)

RS2: (Quench)



(V)

Figure 5.15 Stage 3 Reaction Chemistry

Note: In this stage the product is a solution of V in CH_3CO_2H

Apply MEA-NMP

- Identify Difference: *Difference in Chemical Identity*

– Screen for Critical Features: *None*

- Identify Task Category: *Molecular Structure Transformation*
- Identify Primary Alternative: *React* → *Select Reaction Set*: RS1
  - ☐ Evaluate Preconditions: *React* →*Dynamic Precondition Identification*
  - ☐ Reaction Logic to Eliminate Missing Reactants Precondition Violation

1) Use Reaction Set Information to Define:
   a) Reactants: IV,$KOC(CH_3)_3$
   b) Products: IV'
   c) Catalysts: None
   d) Buffers· None
   e) Diluents: None
   f) Solvents: THF
2) Screen the reaction set for side reactions → *None*
3) Catalyst → *None*
4) Screen reactants for solid or liquid reactants – For each S or L reactant

   IV → Solid →
   a) *Dissolve* or *Mix* with solvent?
      - *No*
      - Generate partial plan → PP1: *Charge* IV
   b) Eliminate IV from above.

   $KOC(CH_3)_3$ → Solid →
   a) *Dissolve* or *Mix* with solvent?
      - *Mix* → *Select Solvent* → THF
      - Generate partial plan → PP2: *Charge* THF, $KOC(CH_3)_3$ and *Mix* operation.
   b) Eliminate $KOC(CH_3)_3$ from above.
5) Screen reactants for gaseous reactants → *None*
6) Screen for any Materials that have not been Charged → *None*
7) Specify ordering of addition of all the partial plans created above.

   PP1: → *2*

   PP2: → *1*

a) Generate partial plan → *Charge* THF, *Charge* $KOC(CH_3)_3$, *Mix*, *Charge* IV, React

8) Check preconditions on the pre-react operations → *OK* → *Apply*

☐ Evaluate Preconditions: *React* → *OK*

- Apply Operation and Generate New Current State:*React* → RS1

B) Creation of Reaction Set 2

➢ Identify Difference: *Difference in Chemical Identity*

– Screen for Critical Features: *None*

- Identify Task Category: *Molecular Structure Transformation*
- Identify Primary Alternative: *React* → *Select Reaction Set*: RS2

☐ Evaluate Preconditions: *React* →*Dynamic Precondition Identification*

☐ Reaction Logic to Eliminate Missing Reactants Precondition Violation

1) Use Reaction Set Information to Define:
    a) Reactants: $RS1^{out}$, CH_3CO_2H
    b) Products: V/CH_3CO_2H
    c) Catalysts: None
    d) Buffers: None
    e) Diluents: None
    f) Solvents: CH_3CO_2H
2) Screen the reaction set for side reactions → *None*
3) Catalyst → *None*
4) Screen reactants for solid or liquid reactants – For each S or L reactant

    $RS1^{out}$ → Liquid →
    - a) *Mix* with solvent?
        - No
        - Generate partial plan → PP1: *Charge* $RS1^{out}$
    - b) Eliminate $RS1^{out}$ from above.

    CH_3CO_2H → Liquid →
    - a) *Mix* with solvent?
        - No

            - Generate partial plan → PP2: *Charge* CH_3CO_2H
        b) Eliminate CH_3CO_2H from above.
    5) Screen reactants for gaseous reactants → *None*
    6) Screen for any Materials that have not been Charged → *None*
    7) Specify ordering of addition of all the partial plans created above.

       PP1: → *2*

       PP2: → *1*

        a) Generate partial plan → *Charge* CH_3CO_2H, *Charge* $RS1^{out}$, React
    8) Check preconditions on the pre-react operations → *OK* → *Apply*

☐ *Dynamic Precondition Identification*: Check heat of reaction → *Exothermic*→

- *Cool* Reaction? → *YES*
    - Evaluate Preconditions: *Cool* → *OK*
- Apply Operation and Generate New Current State: *Cool*: RS2 (20-30C)

☐ Evaluate Preconditions: *React* → *OK*

- Apply Operation and Generate New Current State:*React* → RS2
- *User Override: Add Age to complete reaction (30min)*

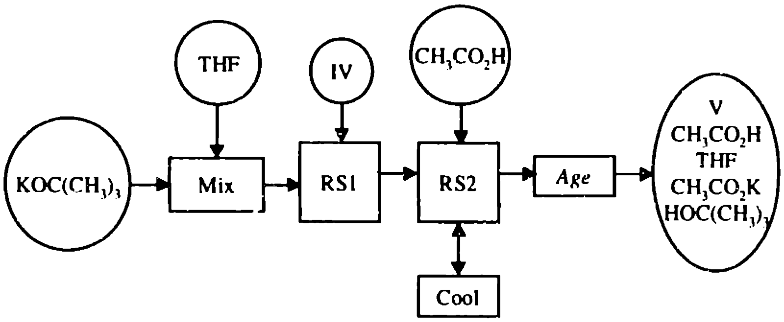


Figure 5.16 Stage 3 Reaction Route

C) <u>Creation of the Separation System</u>

➢ Identify Difference: *Difference in Concentration*

– Screen for Critical Features: *None*

- Identify Task Category: *SolventReplacement*
- Select Primary Alternative: *Concentrate and Mix*
  - ❑ Evaluate Preconditions: *Concentrate*
    - a) single phase liquid input → *Violation*
    - b) $P^{vap}_{solvent} > P^{vap}_{solute}$ → *OK*
      - ◆ Identify White Knight Task Category: *SolidLiquidPhaseSplit*
      - ◆ Select Primary Alternative: *Filter*
        - ⅄ Evaluate Preconditions: *Filter*
          - a) single solid and liquid input → *OK*
      - ◆ Apply Operation and Generate New Current State: *Filter* (remove CH_3CO_2K)
  - ❑ Evaluate Preconditions: *Concentrate* → *OK*
- Apply Operation and Generate New Current State: *Concentrate* (remove $HOC(CH_3)_3$/THF)
  - ❑ Evaluate Preconditions: *Mix*
    - a) Number of inputs ≥ 2 → *Violation*
    - ◆ Identify White Knight Task Category: *Material Transfer*
    - ◆ Identify Primary Alternative: *Charge* → Select Material: CH_3CO_2H
      - ⅄ Evaluate Preconditions: *Charge* → *None*
    - ◆ Apply and Generate New Current State: *Charge* (CH_3CO_2H)
  - ❑ Evaluate Preconditions: *Mix* → *OK*
- Apply and Generate New Current State: *Mix*

➢ Identify Difference: *Difference in Concentration*

– Screen for Critical Features: *None*

- Identify Task Category: SolventReplacement
- Select Primary Alternative: *Concentrate and Mix*
  - Evaluate Preconditions: *Concentrate and Mix*
    1. a) single phase liquid input → *OK*
    2. b) $P^{vap}_{solvent} > P^{vap}_{solute}$ → *OK*
- Apply Operation and Generate New Current State: *Concentrate* (remove $HOC(CH_3)_3$/THF)
  - Evaluate Preconditions: *Mix*
  - a) Number of inputs ≥ 2 → *Violation*
    - Identify White Knight Task Category: *Material Transfer*
    - Identify Primary Alternative: *Charge* → Select Material: CH_3CO_2H
      - Evaluate Preconditions: *Charge* → *None*
    - Apply and Generate New Current State: *Charge* (CH_3CO_2H)
  - Evaluate Preconditions: *Mix* → *OK*
- Apply Operation and Generate New Current State: *Mix* (V/CH_3CO_2H)

- Identify Difference: *None*
- Check for Abstract Operators: *None*

Stage 3 Complete

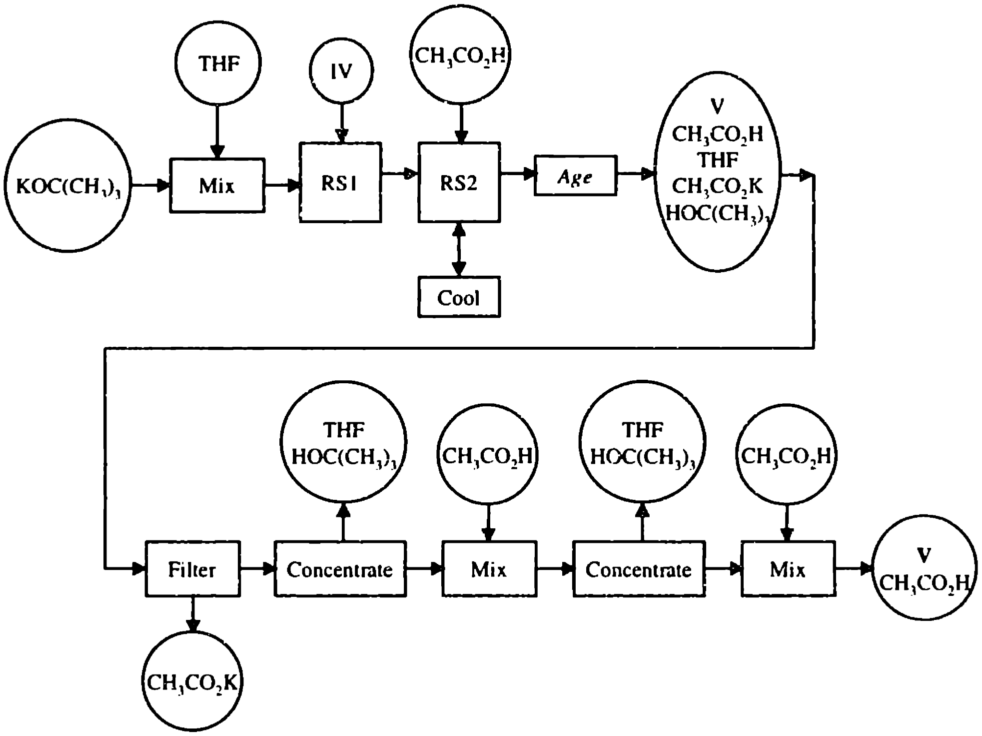


Figure 5.17 Stage 3 Complete Product Train (Yield = 100%)

5.5 Stage 4

<u>RS1:</u>

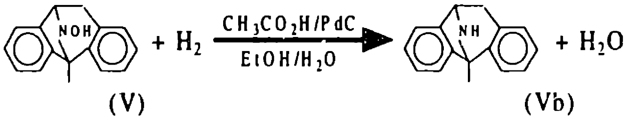


Figure 5.18 Stage 4 Reaction Chemistry

Note: In this stage the product is a solution of Vb in i-PrAc

Apply MEA-NMP

A) <u>Creation of Reaction Set 1</u>

- Identify Differences: *Difference in Chemical Identity*
  - Screen for Critical Features: *None*
    - Identify Task Category: *Molecular Structure Transformation*

- Identify Primary Alternative: *React* → *Select Reaction Set*: RS1
  - ☐ Evaluate Preconditions: *React* →*Dynamic Precondition Identification*
  - ☐ Reaction Logic to Eliminate Missing Reactants Precondition Violation
    1) Use Reaction Set Information to Define:
       a) Reactants: V/CH_3CO_2H, H_2
       b) Products: Vb/i-PrAc
       c) Catalysts: PdC
       d) Buffers: None
       e) Diluents: None
       f) Solvents: EtOH, H_2O, CH_3CO_2H
    2) Screen the reaction set for side reactions → *None*
    3) Catalyst → PdC
       a) *Mix* with solvent? → *Yes* → *Select Solvent* → H_2O
       b) Generate partial plan → PP1: *Charge* H_2O, *Charge* PdC, *Mix*
    4) Screen reactants for solid or liquid reactants

       V/CH_3CO_2H → Liquid →
       a) *Mix* with solvent?
          - No
          - Generate partial plan → PP2: *Charge* V/CH_3CO_2H
       b) Eliminate V/CH_3CO_2H from above.
    5) Screen reactants for gaseous reactants
       a) Dilute?
          - *No*
          - Generate partial plan → PP3: *Charge* H_2
    6) Screen for any Materials that have not been Charged
       a) Generate partial plan → PP4: *Charge* EtOH
    7) Specify ordering of addition of all the partial plans created above.

       PP1: → *1*

       PP2: → *3*

       PP3: → *4*

       PP4: → *2*

a) Generate partial plan → *Charge* H_2O, *Charge* PdC, *Mix*, *Charge* EtOH, *Mix*, *Charge* V/CH_3CO_2H, *Mix*, *Charge* H_2, React

8) Check preconditions on the pre-react operations → *OK* → *Apply*

☐ Evaluate Preconditions: *React* → Check heat of reaction → *OK*

☐ Evaluate Preconditions: *React* → *OK*

- Apply Operation and Generate New Current State: *React* → RS1

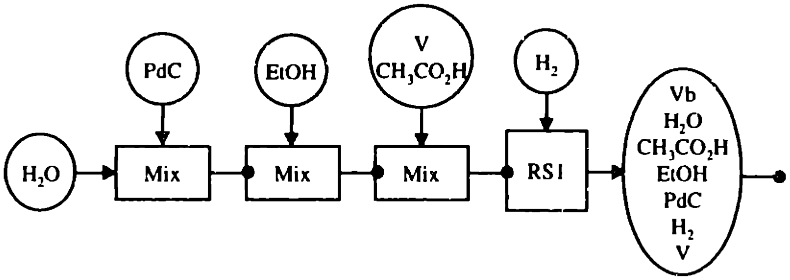


Figure 5.19 Stage 4 Complete Reaction System

B) Creation of the Separation System

➢ Identify Difference: *Difference in Concentration*

– Screen for Critical Features: *None*

- Identify Task Category: *SolventReplacement*
- Select Primary Alternative: *Extract* → Select Extracting Liquid → i-PrAc

☐ Evaluate Preconditions: *Extract*

a) liquid input → *Violation*

b) MSA present? → *OK*

c) $S_{\text{solute i in MSA}} > 0$ → ?

d) $S_{\text{solute j in MSA}} < S_{\text{solute i in MSA}}$ → ?

e) $S_{\text{MSA in Solvent}} \cong 0$ → ?

◆ Identify White Knight Task Category: *SolidLiquidPhaseSplit*

◆ Select Primary Alternative: *Filter*

⅄ Evaluate Preconditions: *Filter*

a) single solid and liquid input → *Violation*

✶ Identify White Knight Task Category: *GasLiquidPhaseSplit*

✶ Select Primary Alternative: *Flash*

◊ Evaluate Preconditions: *Flash*

a) Vapor phase present → *OK*

b) Liquid phase present → *OK*

✶ Apply Operation and Generate New Current State: *Flash* (remove H_2)

⅄ Evaluate Preconditions: *Filter* → *OK*

◆ Apply Operation and Generate New Current State: *Filter* (remove PdC)

☐ Evaluate Preconditions: *Extract* (d) → *Violation*

◆ Identify White Knight Task Category: *Dissolved*ComponentsRemoval

◆ Select Primary Alternative: *Extract* → Select Extracting Liquid → Toluene

⅄ Evaluate Preconditions: *Extract*

a) liquid input → *OK*

b) MSA present? → *OK*

c) $S_{\text{solute i in MSA}} > 0$ → *OK*

d) $S_{\text{solute j in MSA}} < S_{\text{solute i in MSA}}$ → *OK*

e) $S_{\text{MSA in Solvent}} \cong 0$ → *OK*

◆ Apply Operation and Generate New Current State: *Extract* (remove V)

☐ Evaluate Preconditions: *Extract* → (e) → *Violation*

◆ Identify White Knight Task Category: *Solubilty Reduction*

- Select Primary Alternative: *Charge* MSA → Select MSA → NH_4OH/H_2O
  - ⅄ Evaluate Preconditions: *Charge* → *OK*
- Apply Operation and Generate New Current State: *Charge*

☐ Evaluate Preconditions: *Extract* → *OK*

- Apply Operation and Generate New Current State: *Extract* (Vb into i-PrAc)

➢ Identify Difference: *Difference in Concentration*

– Screen for Critical Features: *None*

- Identify Task Category: SolventReplacement
- Select Primary Alternative: *Extract* → Select Extracting Liquid → Brine

☐ Evaluate Preconditions: *Extract*

a) liquid input → *OK*

b) MSA present? → *OK*

c) $S_{\text{solute i in MSA}} > 0$ → *OK*

d) $S_{\text{solute j in MSA}} < S_{\text{solute i in MSA}}$ → *OK*

e) $S_{\text{MSA in Solvent}} \cong 0$ → *OK*

- Apply Operation and Generate New Current State: *Extract* (remove Brine, etc.)

➢ Identify Difference: *Difference in Concentration*

– Screen for Critical Features: *None*

- Identify Task Category: *Concentrate*
- Select Primary Alternative: *Concentrate*

☐ Evaluate Preconditions: *Concentrate*

a) single phase liquid input → *OK*

b) $P^{vap}_{solvent} > P^{vap}_{solute}$ → *OK*

- Apply Operation and Generate New Current State: *Concentrate* (remove i-PrAc)

➢ Identify Difference: *None*

➢ Check for Abstract Operators: *None*

Stage 4 Complete

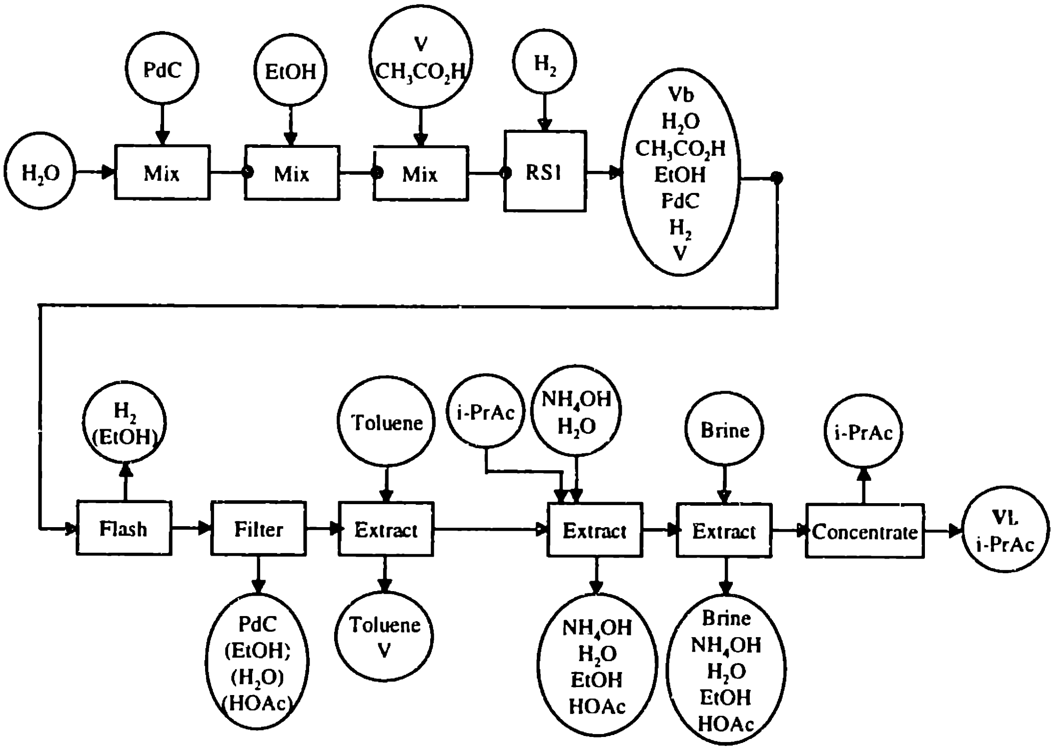
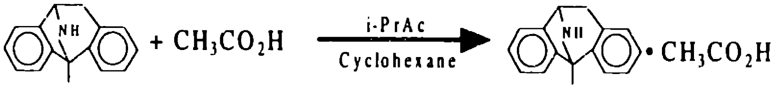


Figure 5.20 Stage 4 Complete Product Train (Yield = 95%)

5.6 Stage 5

RS1:



(Vb)

(VI)

Figure 5.21 Stage 5 Reaction Chemistry

Apply MEA-NMP

A) <u>Creation of Reaction Set 1</u>

- ➢ Identify Differences: *Difference in Chemical Identity*
  - – Screen for Critical Features: *None*
    - Identify Task Category: *Molecular Structure Transformation*
    - Identify Primary Alternative: *React* → *Select Reaction Set*: RS1
      - ☐ Evaluate Preconditions: *React* →*Dynamic Precondition Identification*
      - ☐ Reaction Logic to Eliminate Missing Reactants Precondition Violation

1) Use Reaction Set Information to Define:

a) Reactants: Vb/i-PrAc, CH_3CO_2H

  b) Products: VI

  c) Catalysts: None

  d) Buffers: None

  e) Diluents: None

  f) Solvents: i-PrAc, Cyclohexane

2) Screen the reaction set for side reactions → *None*

3) Catalyst → *None*

4) Screen reactants for solid or liquid reactants

Vb/i-PrAc → Liquid →

  a) *Mix* with solvent?
    - *Yes* → *Select Solvent* → Cyclohexane
    - Generate partial plan → PP1: *Charge* Vb/i-PrAc, *Charge* Cyclohexane, *Mix*

  b) Eliminate Vb/i-PrAc from above.

CH_3CO_2H → Liquid →

  a) *Mix* with solvent?
    - *Yes* → *Select Solvent* → Cyclohexane
    - Generate partial plan → PP2: *Charge* CH_3CO_2H, *Charge* Cyclohexane, *Mix*

  b) Eliminate CH_3CO_2H from above

5) Screen reactants for gaseous reactants → *None*

6) Screen for any Materials that have not been Charged → *None*

7) Specify ordering of addition of all the partial plans created above.

   PP1: → *1*

   PP2: → *2*

   a) Generate partial plan → *Charge* Vb/i-PrAc, *Charge* Cyclohexane, *Mix*, *Charge* CH_3CO_2H, *Charge* Cyclohexane, *Mix*, *Charge* PP1, *Charge* PP2, React

8) Check preconditions on the pre-react operations → *OK* → *Apply*

- ☐ Evaluate Preconditions: *React* → Check heat of reaction → *OK*

- ☐ Evaluate Preconditions: *React* → *OK*

- Apply Operation and Generate New Current State: *React* → RS1

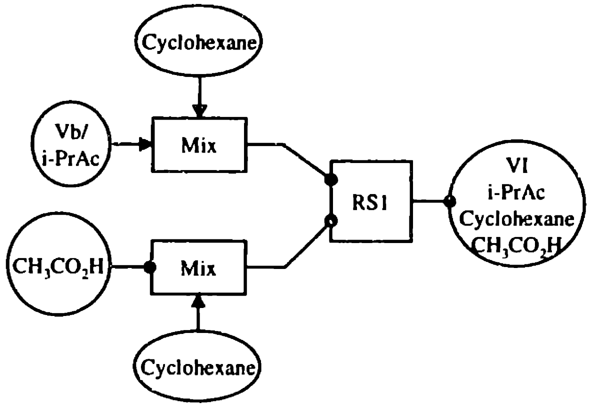


Figure 5.22 Stage 5 Complete Reaction System

B) Creation of the Separation System

➢ Identify Difference: *Difference in Concentration*

– Screen for Critical Features: *None*

- Identify Task Category: SolidtoLiquidPhaseChange

- Select Primary Alternative: *Crystallize*

- ☐ Evaluate Preconditions: *Crystallize*

  a) No vapor phase as input →OK

  b) No multiple liquid phase as input → *OK*

        - c) No solid phase as input → *OK*
        - d) The composition of the feed mixture must lie within a region of the solid-liquid equilibrium that allows the desired component to form pure crystals. → *OK*
        - e) The melting point of the solvent must be less than the melting point of the solute → *OK*
        - f) The concentration of the solute must be greater than the solubility of the solute → *OK*
        - g) Solubility of the product must be less than the solubility of all other components → *OK*
    - Apply Operation and Generate New Current State: *Crystallize* (VI)

- ➢ Identify Difference: *Difference in Concentration*
  - – Screen for Critical Features: *None*
    - Identify Task Category: *SolidLiquidPhaseSplit*
    - Select Primary Alternative: *Filter*
      - ❑ Evaluate Preconditions: *Filter*
        - a) single solid and liquid input → *OK*
    - Apply Operation and Generate New Current State: *Filter* (VI –wet)

- ➢ Identify Difference: *Difference in Concentration*
  - – Screen for Critical Features: *None*
    - Identify Task Category: *LiquidfromSolid*
    - Select Primary Alternative: *Wash-Solid* → Select MSA → i-PrAc/cyclohexane
      - ❑ Evaluate Preconditions: *Wash-Solid*
        - a) Solubility of solid in MSA = low → *OK*

- Apply and Generate New Current State: *Wash-Solid* (VI – wet)

➢ Identify Difference: *Difference in Concentration*

– Screen for Critical Features: *None*

- Identify Task Category: *LiquidfromSolid*
- Select Primary Alternative: *Dry*
  - ☐ Evaluate Preconditions: *Dry*
    a) solid and liquid input → *OK*
- Apply and Generate New Current State: *Dry* (VI)

➢ Identify Difference: *None*

➢ Check for Abstract Operators: *None*

Stage 5 Complete

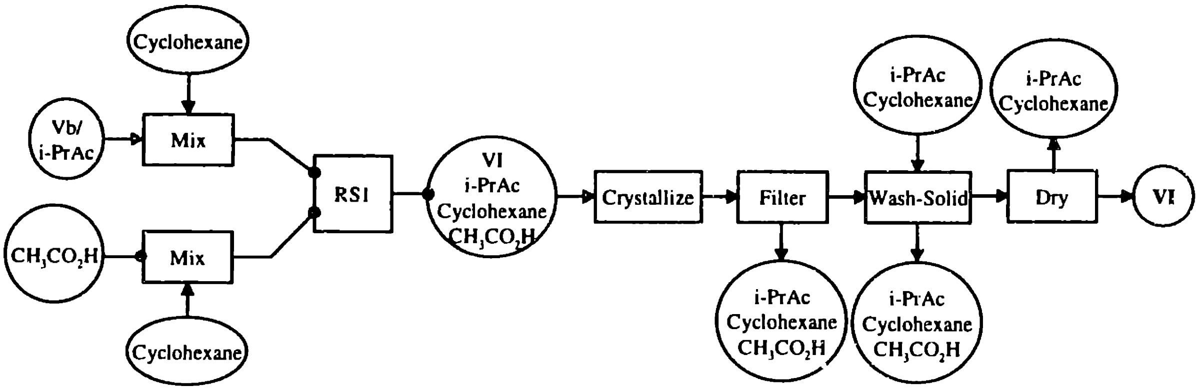


Figure 5.23 Stage 5 Complete Product Train (Yield = 87%)

5.7 Stage 6

In this stage we will show how the promotion of operations is used when white knights cannot be identified.

RS1:

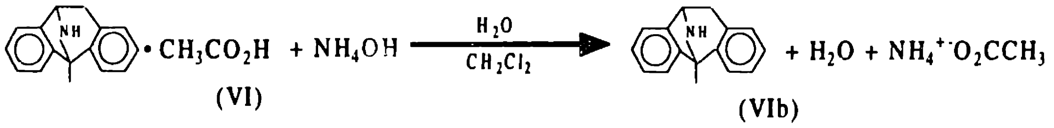


Figure 5.24 Stage 6 Reaction Chemistry

Note: The stage product is a mixture of VIb in 95% EtOH

Apply MEA-NMP

A) Creation of Reaction Set 1

- ➢ Identify Differences: *Difference in Chemical Identity*
  - – Screen for Critical Features: *None*
    - • Identify Task Category: *Molecular Structure Transformation*
    - • Identify Primary Alternative: *React → Select Reaction Set*: RS1
      - ☐ Evaluate Preconditions: *React →Dynamic Precondition Identification*
      - ☐ Reaction Logic to Eliminate Missing Reactants Precondition Violation
        1) Use Reaction Set Information to Define:
           a) Reactants: VI, NH_4OH/H_2O
           b) Products: VIb/EtOH/H_2O
           c) Catalysts: None
           d) Buffers: None
           e) Diluents: None
           f) Solvents: H_2O, CH_2Cl_2
        2) Screen the reaction set for side reactions → *None*
        3) Catalyst → *None*
        4) Screen reactants for solid or liquid reactants

VI → Solid →

a) *Dissolve* or *Mix* with solvent?

- *Dissolve* → *Select Solvent* → CH_2Cl_2, H_2O
- Generate partial plan → PP1: *Charge* CH_2Cl_2, *Charge* H_2O, *Mix*, *Charge* VI, *Dissolve*

b) Eliminate VI from above.

NH_4OH/H_2O → Liquid →

a) *Mix* with solvent?

- *No*
- Generate partial plan → PP2: *Charge* NH_4OH/H_2O

b) Eliminate NH_4OH/H_2O from above

5) Screen reactants for gaseous reactants → *None*

6) Screen for any Materials that have not been Charged → *None*

7) Specify ordering of addition of all the partial plans created above.

PP1: → *1*

PP2: → 2

a) Generate partial plan → *Charge* CH_2Cl_2, *Charge* H_2O, *Mix*, *Charge* VI, *Dissolve*, *Charge* NH_4OH/H_2O, React

8) Check preconditions on the pre-react operations → *OK* → *Apply*

☐ Evaluate Preconditions: *React* → Check heat of reaction → *OK*

☐ Evaluate Preconditions: *React* → *OK*

- Apply Operation and Generate New Current State: *React* → RS1

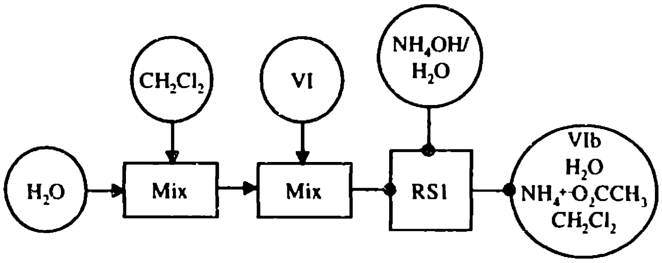


Figure 5.25 Stage 6 Complete Reaction System

B) Creation of the Separation System

- Identify Difference: *Difference in Concentration*
  - Screen for Critical Features: *None*
    - Identify Task Category: *Solvent Replacement*
    - Select Primary Alternative: *Concentrate and Mix*
      - Evaluate Preconditions: *Concentrate*
        - a) Single phase liquid → *Violation*
        - b) $P^{vap}_{solvent} > P^{vap}_{solute}$ → *OK*
        - Identify White Knight Task Category: *LiquidLiquidPhaseSplit*
        - Select Primary Alternative: *Decant*
          - Evaluate Preconditions: *Decant*
            - a) No gas phase input → *OK*
            - b) Two phase liquid input → *OK*
            - c) $V_1 \geq 0.1 * V_2$ → *OK*
        - Apply and Generate New Current State: *Decant* (remove H_2O and most $NH_4^{+-}O_2CCH_3$)
      - Evaluate Preconditions: *Concentrate* → *OK*
    - Apply and Generate New Current State: *Concentrate* (remove CH_2Cl_2)
      - Evaluate Preconditions: *Mix*
        - a) Number of inputs ≥ 2 → *Violation*
        - Identify White Knight Task Category: *Material Transfer*
        - Identify Primary Alternative: *Charge* → Select Material: 95% EtOH
          - Evaluate Preconditions: *Charge* → *None*
        - Apply and Generate New Current State: *Charge* (95% EtOH)

      - ☐ Evaluate Preconditions: *Mix* → *OK*
    - Apply and Generate New Current State: *Mix*

- ➢ Identify Difference: *Difference in Concentration*
  - – Screen for Critical Features: *None*
    - Identify Task Category: *DissolvedMaterialRemoval*
    - Select Primary Alternative: *Extract* → Select Extracting Liquid → Brine
      - ☐ Evaluate Preconditions: *Extract*
        - a) liquid input → *OK*
        - b) MSA present → *OK*
        - c) $S_{\text{solute i in MSA}} > 0$ → *OK*
        - d) $S_{\text{solute j in MSA}} < S_{\text{solute i in MSA}}$ → *Violation*
        - e) $S_{\text{MSA in Solvent}} \cong 0$ → *Violation*

Note: In this step lets assume we cannot find a solvent the will preferentially Extract the impurity.

- ◆ Identify White Knight Task Category: *None*

The methodology now moves on to check if there is an operation preceding the Extract operation. In this case Extract is preceded by Concentrate and Mix which we treat as a single operation. The methodology now indicates that we promote the Extract operation before the Concentrate operation.

- Promote Extract
  - ☐ Evaluate Preconditions: Extract
    - a) liquid input → *OK*
    - b) MSA present? → *OK*
    - c) $S_{\text{solute i in MSA}} > 0$ → *OK*
    - d) $S_{\text{solute j in MSA}} < S_{\text{solute i in MSA}}$ → *OK*

e) $S_{\text{MSA in Solvent}} \cong 0$ → *OK*

- Apply Operation and Generate New Current State: *Extract* (remove Aqueous Layer)

We now evaluate the preconditions of the Concentrate operation versus the postconditions generated by the Extract operation. There is no Violation so we go ahead and apply the Concentrate operation.

- ☐ Evaluate *Concentrate* Preconditions → *OK*
- Apply operation and Generate New Current State: *Concentrate* (remove CH_2Cl_2)
- ☐ Evaluate Preconditions: *Mix* → *OK*
- Apply operation and Generate New Current State: *Mix* (Concentrated mixture with 95% EtOH)

The preceding steps have shown how promotion can be used to remove preconditions Violations. We now continue with the MEA-NMP methodology by identification of the next difference.

➢ Identify Difference: *Difference in Concentration*

- Screen for Critical Features: *None*
  - Identify Task Category: *Concentrate*
  - Identify Primary Alternative: *Concentrate*
    - ☐ Evaluate Preconditions: *Concentrate*
      a) single liquid phase input → *OK*
      b) $P^{vap}_{solvent} > P^{vap}_{solute}$ → *OK*
  - Apply and Generate New Current State: *Concentrate* (remove EtOH, CH_2Cl_2)

➢ Identify Difference: *None*

➢ Check for Abstract Operators: *None*

Stage 6 Complete

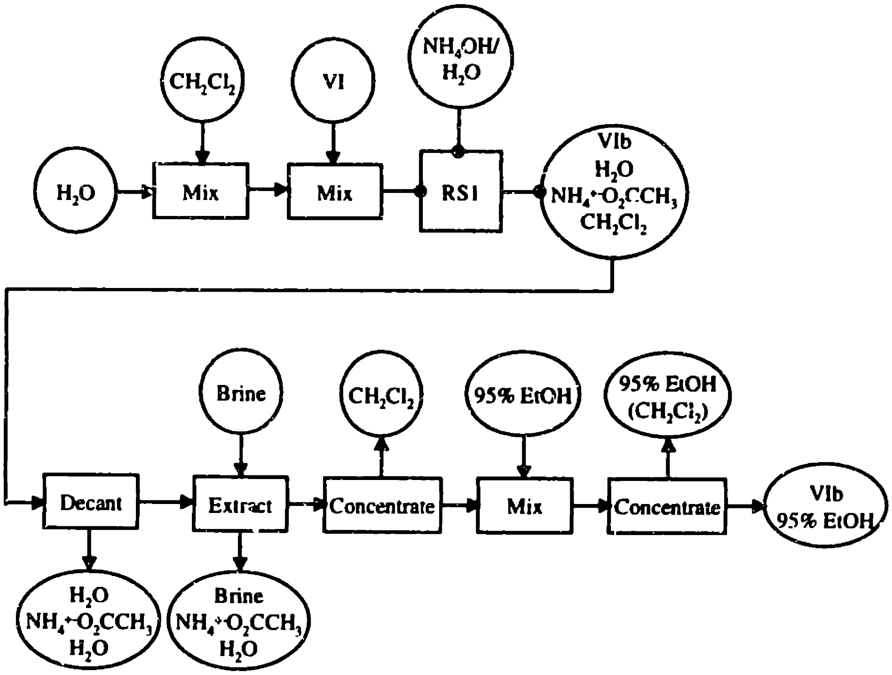


Figure 5.26 Stage 6 Complete Product Train (Yield = 99%)

5.8 Stage 7

RS1:

(VIb)

(VII)

Figure 5.27 Stage 7 Reaction Chemistry

Apply MEA-NMP

A) Creation of Reaction Set 1

- Identify Differences: *Difference in Chemical Identity*
  - Screen for Critical Features: *None*
    - Identify Task Category: *Molecular Structure Transformation*

- Identify Primary Alternative: *React* → *Select Reaction Set*: RS1
  - ☐ Evaluate Preconditions: *React* →*Dynamic Precondition Identification*
  - ☐ Reaction Logic to Eliminate Missing Reactants Precondition Violation

1) Use Reaction Set Information to Define:
   a) Reactants: VIb/EtOH/H_2O, EtOH, l-malic acid
   b) Products: VII
   c) Catalysts: None
   d) Buffers: None
   e) Diluents: None
   f) Solvents: 95% EtOH
2) Screen the reaction set for side reactions → *None*
3) Catalyst → *None*
4) Screen reactants for solid or liquid reactants

   VIb/EtOH/H_2O → Liquid →
   a) *Mix* with solvent?
      - *Mix* → *Select Solvent* → 95% EtOH
      - Generate partial plan → PP1: *Charge* VIb/EtOH/H_2O, *Charge* 95% EtOH, *Mix*
   b) Eliminate VIb/EtOH/H_2O from above.

   EtOH → Liquid →
   a) *Mix* with solvent?
      - No
      - Generate partial plan → PP2: *Charge* EtOH
   b) Eliminate EtOH from above

   l-malic acid → Solid →
   a) *Dissolve* and *Mix* with solvent?
      - *Dissolve* → *Select Solvent* → 95% EtOH
      - Generate partial plan → PP3: *Charge* l-malic acid, *Charge* 95% EtOH
   c) Eliminate l-malic acid from above
5) Screen reactants for gaseous reactants → *None*
6) Screen for any Materials that have not been Charged → *None*

7) Specify ordering of addition of all the partial plans created above.

   PP1: → *1*

   PP2: → 2

   PP3: → *3*

   a) Combine PP1 and PP2? → *Yes*
   b) Combine PP1/PP2 and PP3 → No
   c) Generate partial plan → *Charge* VIb/EtOH/H_2O, *Charge* 95% EtOH, *Mix*, *Charge* 95% EtOH, *Charge* l-malic acid, *Dissolve*, *Charge* PP1, *Charge* PP3, React

8) Check preconditions on the pre-react operations → *OK* → *Apply*

☐ Evaluate Preconditions: *React* → Check heat of reaction → *OK*

☐ Evaluate Preconditions: *React* → *OK*

- Apply Operation and Generate New Current State: *React* → RS1

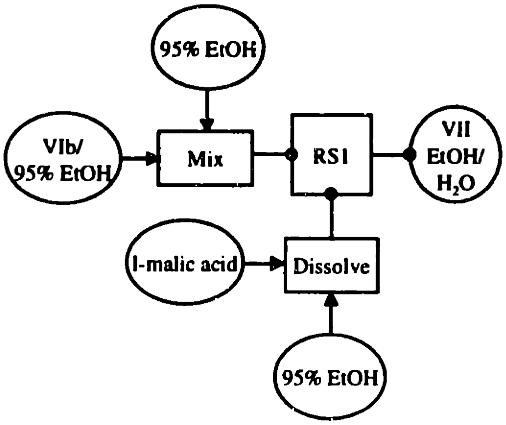


Figure 5.28 Stage 7 Complete Reaction Structure

B) Creation of the Separation System

➢ Identify Difference: *Difference in Concentration*

– Screen for Critical Features: *None*

- Identify Task Category: *SolidtoLiquidPhaseChange*

- Identify Primary Alternative: *Crystallize*

☐ Evaluate Preconditions: *Crystallize*

a) No vapor phase as input → *OK*

            b) No multiple liquid phase as input → *Violation*

            c) No solid phase as input → *OK*

            d) The composition of the feed mixture must lie within a region of the solid-liquid equilibrium that allows the desired component to form pure crystals. → *OK*

            e) The melting point of the solvent must be less than the melting point of the solute → *OK*

            f) The concentration of the solute must be greater than the solubility of the solute → *Violation*

            g) Solubility of the product must be less than the solubility of all other components → *OK*

- - - Identify White Knight Task Category: *Solubilty Reduction*
      - Identify Primary Alternative: *Cool*
        - Evaluate Preconditions: *Cool* → *OK*
      - Apply Operations and Generate New Current State: *Cool*
    - Evaluate Preconditions: *Crystallize* → *OK*
  - Apply Operation and Generate New Current State: *Crystallize* (VII solid)
- Identify Difference: *Difference in Concentration*
  - Screen for Critical Features: *None*
    - Identify Task Category: *SolidLiquidPhaseSplit*
    - Identify Primary Alternative: *Filter*
      - Evaluate Preconditions: *Filter*

        a) single solid and liquid input → *OK*
    - Apply Operation and Generate New Current State: *Filter* (remove liquid, solid contains enantiomers and the product is the (+)-enantiomer)

- Identify Difference: *Difference in Concentration*
  - Screen for Critical Features: *None*
    - Identify Task Category: *SolidSolidSep*
    - Identify Primary Alternative: *ReCrystallize* (*Dissolve* and *Crystallize*) → *Select Solvent* → 90% EtOH

Note: The reCrystallize operation is a compound operation made up of Dissolve and Crystallize.

- Evaluate Preconditions: *Dissolve*
  1. solid input → *OK*
  2. liquid input → *OK*
  3. $S_{solute} \geq \frac{\text{mass of Solute}}{\text{vol of Solution}}$ → *Violation*
  - Identify White Knight Task Category: *Solubility Increase*
  - Identify Primary Alternative: *Heat*
    - Evaluate Preconditions: *Heat* → *OK*
  - Apply Operation and Generate New Current State: *Heat*
- Evaluate Preconditions: *Dissolve* → *OK*

- Apply Operation and Generate New Current State: *Dissolve*
  - Evaluate Preconditions: *Crystallize* → (f) →*Violation*
    - Identify White Knight Task Category: *Solubilty Reduction*
    - Identify Primary Alternative: *Cool*
      - Evaluate Preconditions: *Cool* → *OK*
    - Apply Operation and Generate New Current State: *Cool*
  - Evaluate Preconditions: *Crystallize* → *OK*
- Apply Operation and Generate New Current State: *Crystallize*

- ➢ Identify Difference: *Difference in Concentration*
  - – Screen for Critical Features: *None*
    - Identify Task Category: *SolidLiquidPhaseSplit*
      - Identify Primary Alternative: *Filter*
        - ☐ Evaluate Preconditions: *Filter*
          - a) single solid and liquid input → *OK*
      - Apply operation and Generate New Current State: *Filter* (remove liquid)
- ➢ Identify Difference: *Difference in Concentration*
  - – Screen for Critical Features: *None*
    - Identify Task Category: *LiquidfromSolid*
      - Identify Primary Alternative: *Wash-Solid* → *Select Solvent* → 95% EtOH
        - ☐ Evaluate Preconditions: *Wash-Solid*
          - a) $S_{\text{solid in MSA}}$ = low → *OK*
          - b) Solid input → *OK*
          - c) Liquid input → *OK*
      - Apply Operation and Generate New Current State: *Wash-Solid*
- ➢ Identify Difference: *Difference in Concentration*
  - – Screen for Critical Features: *None*
    - Identify Task Category: *LiquidfromSolid*
      - Identify Primary Alternative: *Dry*
        - ☐ Evaluate Preconditions: *Dry*
          - a) solid and liquid input → *OK*
      - Apply Operation and Generate New Current State: *Dry* (VII solid)
- ➢ Identify Difference: *None*

- Abstract Operations: *None*

Stage 7 Complete

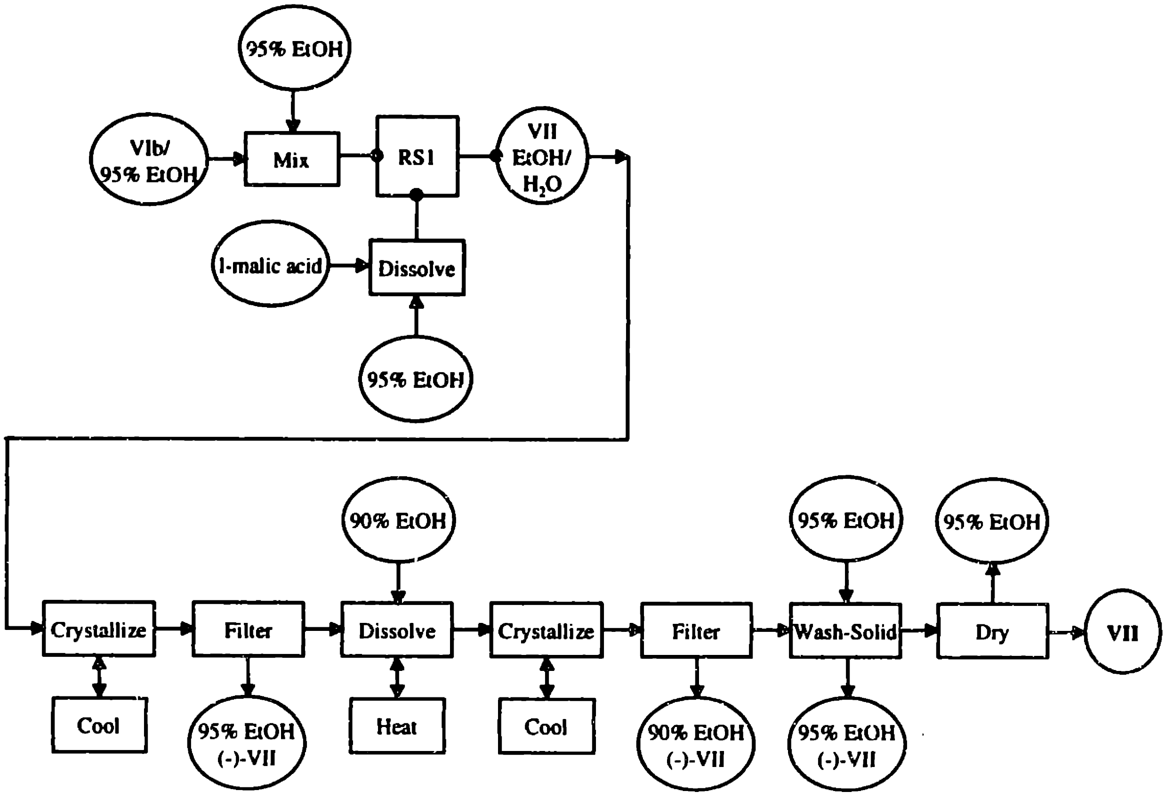


Figure 5.29 Stage 7 Complete Product Train (Yield = 87%)

5.9 Stage 8

RS1:

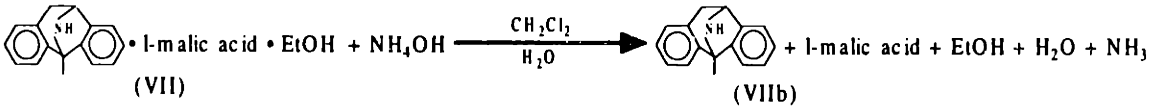


Figure 5.30 Stage 8 Reaction Chemistry

Apply MEA-NMP

A) Creation of Reaction Set 1

- Identify Differences: *Difference in Chemical Identity*

  - Screen for Critical Features: *None*

- Identify Task Category: *Molecular Structure Transformation*
- Identify Primary Alternative: *React* → *Select Reaction Set*: RS1
  - ☐ Evaluate Preconditions: *React* →*Dynamic Precondition Identification*
  - ☐ Reaction Logic to Eliminate Missing Reactants Precondition Violation

1) Use Reaction Set Information to Define:
   a) Reactants: VII, NH_4OH/H_2O
   b) Products: VIIb/EtOH
   c) Catalysts: None
   d) Buffers: None
   e) Diluents: None
   f) Solvents: CH_2Cl_2, H_2O
2) Screen the reaction set for side reactions → *None*
3) Catalyst → *None*
4) Screen reactants for solid or liquid reactants

   VII → Solid →
   a) *Dissolve* or *Mix* with solvent?
      - *Dissolve* → *Select Solvent* → CH_2Cl_2/H_2O
      - Generate partial plan → PP1: *Charge* CH_2Cl_2, *Charge* H_2O, *Mix*, *Charge* VII, *Dissolve*
   b) Eliminate VII from above.

   NH_4OH/H_2O → Liquid →
   a) *Mix* with solvent?
      - *No*
      - Generate partial plan → PP2: *Charge* NH_4OH/H_2O
   b) Eliminate NH_4OH/H_2O from above
5) Screen reactants for gaseous reactants → *None*
6) Screen for any Materials that have not been Charged → *None*
7) Specify ordering of addition of all the partial plans created above.

   PP1: → *1*

   PP2: → *2*
   a) Generate partial plan → *Charge* CH_2Cl_2, *Charge* H_2O, *Mix*, *Charge* VII, *Dissolve*, *Charge* NH_4OH/H_2O, React

8) Check preconditions on the pre-react operations → *OK* → *Apply*

- Evaluate Preconditions: *React* → Check heat of reaction → *Exothermic* →
  - *Cool*? → *Yes* (20-25C)
    - Evaluate Precondtions: *Cool* → *OK*
  - Apply Operation and Generate New Current State: *Cool* → RS1
- Evaluate Preconditions: *React* → *OK*

- Apply Operation and Generate New Current State: *React* → RS1

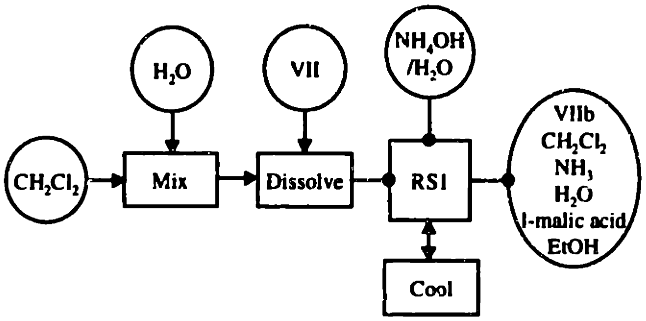


Figure 5.31 Stage 8 Complete Reaction Structure

B) Creation of the Separation System

- Identify Difference: *Difference in Concentration*
  - Screen for Critical Features: *None*
    - Identify Task Category: *Solvent Replacement*

- Select Primary Alternative: *Concentrate and Mix*
    - Evaluate Preconditions: *Concentrate*

        a) single phase liquid → *Violation*

        b) $P^{vap}_{solvent} > P^{vap}_{solute}$ →OK

        - Identify White Knight Task Category: *LiquidLiquidPhaseSplit*
        - Select Primary Alternative: *Decant*
            - Evaluate Preconditions: *Decant*

                a) No gas phase input → *OK*

                b) Two phase liquid input → *OK*

                c) Volume of phase 1 ≥ 0.1*Volume of phase 2 (phase 1 being the lesser volume phase) → *OK*

        - Apply operation and Generate New Current State: *Decant* (remove aqueous phase)
    - Evaluate Preconditions: *Concentrate* → *OK*
- Apply Operation and Generate New Current State: *Concentrate*
    - Evaluate Preconditions: *Mix*

        a) Number of inputs ≥ 2 → *Violation*

        - Identify White Knight Task Category: *Material Transfer*
        - Identify Primary Alternative: *Charge* → Select Material: EtOH
            - Evaluate Preconditions: *Charge* → *None*
        - Apply and Generate New Current State: *Charge* (supply EtOH for *Mix*)
    - Evaluate Preconditions: *Mix* → *OK*
- Apply Operation and Generate New Current State: *Mix*

- Identify Difference: *Difference in Concentration*
  - Screen for Critical Features: *None*
    - Identify Task Category: *Solvent Replacement*
    - Select Primary Alternative: *Concentrate and Mix*
      - Evaluate Preconditions: *Concentrate*
        - a) single phase liquid → *Violation*
        - b) $P^{vap}_{solvent} > P^{vap}_{solute}$ →*OK*
        - Identify White Knight Task Category: *LiquidLiquidPhaseSplit*
        - Select Primary Alternative: *Decant*
- Evaluate Preconditions: *Decant*
          - a) No gas phase input → *OK*
          - b) Two phase liquid input → *OK*
          - c) Volume of phase 1 ≥ 0.1*Volume of phase 2 (phase 1 being the lesser volume phase) → *OK*
        - Apply operation and Generate New Current State: *Decant* (remove aqueous phase)
      - Evaluate Preconditions: *Concentrate* → *OK*
    - Apply Operation and Generate New Current State: *Concentrate*
      - Evaluate Preconditions: *Mix*
        - a) Number of inputs ≥ 2 → *Violation*
        - Identify White Knight Task Category: *Material Transfer*
        - Identify Primary Alternative: *Charge* → Select Material: EtOH
          - Evaluate Preconditions: *Charge* → *None*
        - Apply and Generate New Current State: *Charge* (supply EtOH for *Mix*)

        - Evaluate Preconditions: *Mix* → *OK*
    - Apply Operation and Generate New Current State: *Mix*

- Identify Difference: *Difference in Concentration*
  - Screen for Critical Features: *None*
    - Identify Task Category: *DissolvedMaterialRemoval*
    - Select Primary Alternative: *Extract* → Select Extracting Liquid → Brine
      - Evaluate Preconditions: *Extract*
      a) liquid input → *OK*
      b) MSA present → *OK*
      c) $S_{\text{solute i in MSA}} > 0$ → *OK*
      d) $S_{\text{solute j in MSA}} < S_{\text{solute i in MSA}}$ → *Violation*
      e) $S_{\text{MSA in Solvent}} \cong 0$ → *OK*
        - Identify White Knight Task Category: *None*
    - Promote *Extract*
      - Evaluate Preconditions: *Extract*
      a) liquid input → *OK*
      b) MSA present? → *OK*
      c) $S_{\text{solute i in MSA}} > 0$ → *OK*
      d) $S_{\text{solute j in MSA}} < S_{\text{solute i in MSA}}$ → *Violation*
      e) $S_{\text{MSA in Solvent}} \cong 0$ → *OK*
        - Identify White Knight Task Category: *None*
    - Promote *Extract*
      - Evaluate Preconditions: *Extract*
        a) liquid input → *OK*

b) MSA present? → *OK*

c) $S_{\text{solute i in MSA}} > 0$ → *OK*

d) $S_{\text{solute j in MSA}} < S_{\text{solute i in MSA}}$ → *OK*

e) $S_{\text{MSA in Solvent}} \cong 0$ → *OK*

- Apply Operation and Generate New Current State: *Extract* (remove Aqueous Layer)
  - ☐ Evaluate *Concentrate* Preconditions → *OK*
- Apply operation and Generate New Current State: *Concentrate* (remove CH_2Cl_2)
  - ☐ Evaluate Preconditions: *Mix* → *OK*
- Apply operation and Generate New Current State: *Mix* (Concentrated mixture with EtOH)
- Remaining Operations? → *Yes*
  - ☐ Evaluate *Concentrate* Preconditions → *OK*
- Apply operation and Generate New Current State: *Concentrate* (remove CH_2Cl_2)
  - ☐ Evaluate Preconditions: *Mix* → *OK*
- Apply operation and Generate New Current State: *Mix* (Concentrated mixture with EtOH)

➢ Identify Difference: *Difference in Concentration*

- Screen for Critical Features: *None*
  - Identify Task Category: *Concentrate*
  - Identify Primary Alternative: *Concentrate*
    - ☐ Evaluate Preconditions: *Concentrate*

a) single liquid phase input → *OK*

b) $P^{vap}_{solvent} > P^{vap}_{solute}$ → *OK*

- Apply and Generate New Current State: *Concentrate* (remove EtOH, CH_2Cl_2)

➢ Identify Difference: *None*

➢ Abstract Operators? *None*

Stage 8 Complete

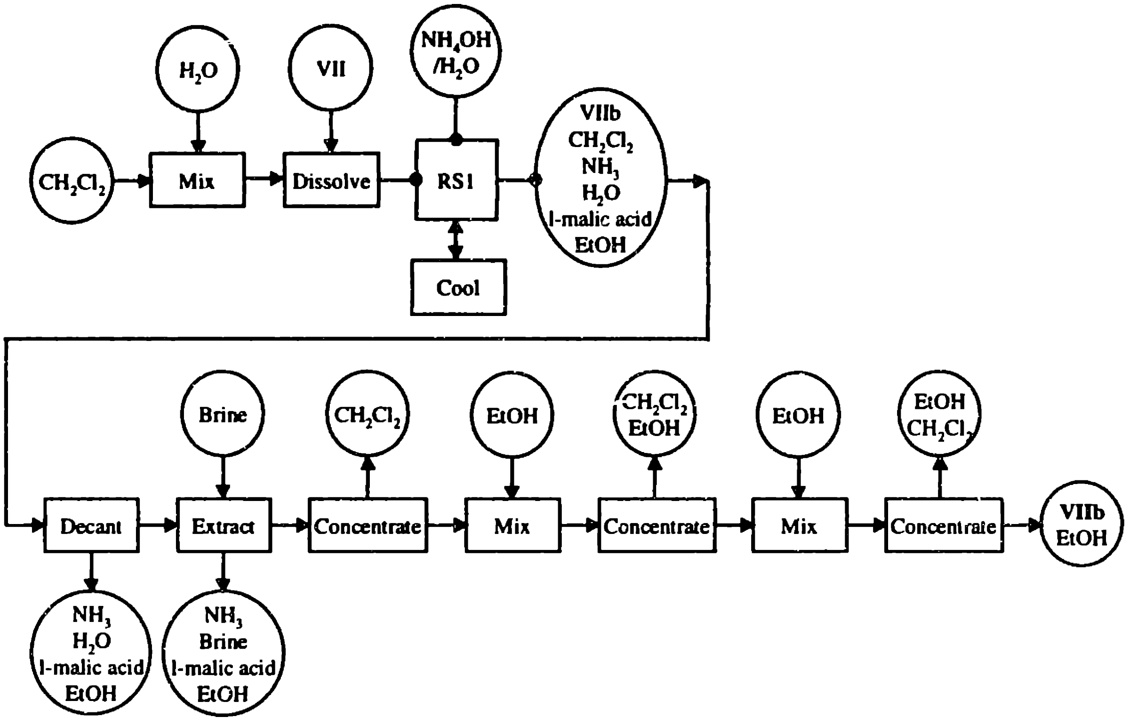


Figure 5.32 Stage 8 Complete Product Train (Yield = 100%)

5.10 Stage 9

<u>RS1:</u>

(VIIb) (VIII)

Figure 5.33 Stage 9 Reaction Chemistry

Apply MEA-NMP

A) <u>Creation of Reaction Set 1</u>

- ➢ Identify Differences: *Difference in Chemical Identity*
  - – Screen for Critical Features: *None*
    - Identify Task Category: *Molecular Structure Transformation*
    - Identify Primary Alternative: *React* → *Select Reaction Set*: RS1
      - ☐ Evaluate Preconditions: *React* →*Dynamic Precondition Identification*
      - ☐ Reaction Logic to Eliminate Missing Reactants Precondition Violation

1) Use Reaction Set Information to Define:
   a) Reactants: VIIb/EtOH, maleic acid
   b) Products: VIII
   c) Catalysts: None
   d) Buffers: None
   e) Diluents: None
   f) Solvents: EtOH
2) Screen the reaction set for side reactions → *None*
3) Catalyst → *None*
4) Screen reactants for solid or liquid reactants

   VIIb/EtOH → Liquid →

   a) *Mix* with solvent?
      - *No*
      - Generate partial plan → PP1: *Charge* VIIb/EtOH
   b) Eliminate VIIb/EtOH from above.

   maleic acid → Solid →

   a) *Dissolve* or *Mix* with solvent?
      - *Dissolve* → *Select Solvent* → EtOH
      - Generate partial plan → PP2: *Charge* EtOH, *Charge* maleic acid, *Dissolve*
   b) Eliminate maleic acid from above.
5) Screen reactants for gaseous reactants → *None*
6) Screen for any Materials that have not been Charged → *None*

7) Specify ordering of addition of all the partial plans created above.

PP1: → *1*

PP2: → 2

a) Generate partial plan → *Charge* VIIb/EtOH, *Charge* EtOH, *Charge* maleic acid, *Dissolve*, *Charge* PP2, React

8) Check preconditions on the pre-react operations → *OK* → *Apply*

☐ Evaluate Preconditions: *React* → Check heat of reaction → *Exothermic* →

- ◆ *Cool?* → *Yes* (25-30C)
  - ⅄ Evaluate Preconditions: *Cool* → *OK*
- ◆ Apply Operation and Generate New Current State: *Cool* → RS1

☐ Evaluate Preconditions: *React* → *OK*

- Apply Operation and Generate New Current State: *React* → RS1

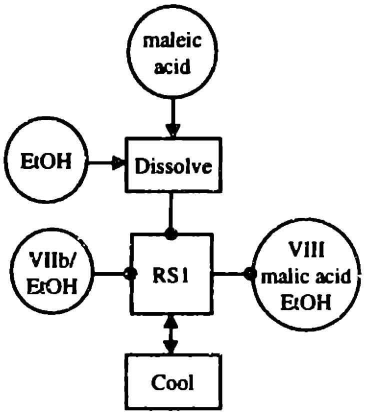


Figure 5.34 Stage 9 Complete Reaction Structure

B) *Creation of the Separation System*

➢ Identify Difference: *Difference in Concentration*

– Screen for Critical Features: *None*

• Identify Task Category: *SolidtoLiquidPhaseChange*

• Identify Primary Alternative: *Crystallize*

❑ Evaluate Preconditions: *Crystallize*

a) No vapor phase as input → *OK*

b) No multiple liquid phase as input → *OK*

c) No solid phase as input → *OK*

d) The composition of the feed mixture must lie within a region of the solid-liquid equilibrium that allows the desired component to form pure crystals. → *OK*

e) The melting point of the solvent must be less than the melting point of the solute → *OK*

f) The concentration of the solute must be greater than the solubility of the solute → *Violation*

g) Solubility of the product must be less than the solubility of all other components → *OK*

◆ Identify White Knight Task Category: *Solubilty Reduction*

◆ Identify Primary Alternative: *Cool*

⅄ Evaluate Preconditions: *Cool* → *OK*

◆ Apply Operations and Generate New Current State: *Cool*

❑ Evaluate Preconditions: *Crystallize* → (f) → *Violation*

◆ Identify White Knight Task Category: *Solubilty Reduction*

◆ Identify Primary Alternative: *Mix* with MSA → Select MSA → Ethyl Acetate

            - Evaluate Preconditions: *Mix* → *OK*
        - Apply operation and Generate New Current State: *Mix*
      - Evaluate Preconditions: *Crystallize* → *OK*
    - Apply operation and Generate New Current State: *Crystallize* (VIII solid)

- Identify Difference: *Difference in Concentration*
  - Screen for Critical Features: *None*
    - Identify Task Category: *SolidLiquidPhaseSplit*
    - Identify Primary Alternative: *Filter*
      - Evaluate Preconditions: *Filter*
        a) single solid and liquid input → *OK*
    - Apply operation and Generate New Current State: *Filter* (remove liquid)

- Identify Difference: *Difference in Concentration*
  - Screen for Critical Features: *None*
    - Identify Task Category: *LiquidfromSolid*
    - Identify Primary Alternative: *Wash-Solid* → *Select Solvent* → Ethyl Acetate/EtOH
      - Evaluate Preconditions: *Wash-Solid*
        a) $S_{\text{solid in MSA}}$ = low → *OK*
        b) Solid input → *OK*
        c) Liquid input → *OK*
    - Apply Operation and Generate New Current State: *Wash-Solid*

➢ Identify Difference: *Difference in Concentration*

  – Screen for Critical Features: *None*

    - Identify Task Category: *LiquidfromSolid*

    - Identify Primary Alternative: *Dry*

      □ Evaluate Preconditions: *Dry*

      a) solid and liquid input → *OK*

    - Apply Operation and Generate New Current State: *Dry* (VIII solid)

➢ Identify Difference: *None*

➢ Abstract Operations: *None*

Stage 9 Complete

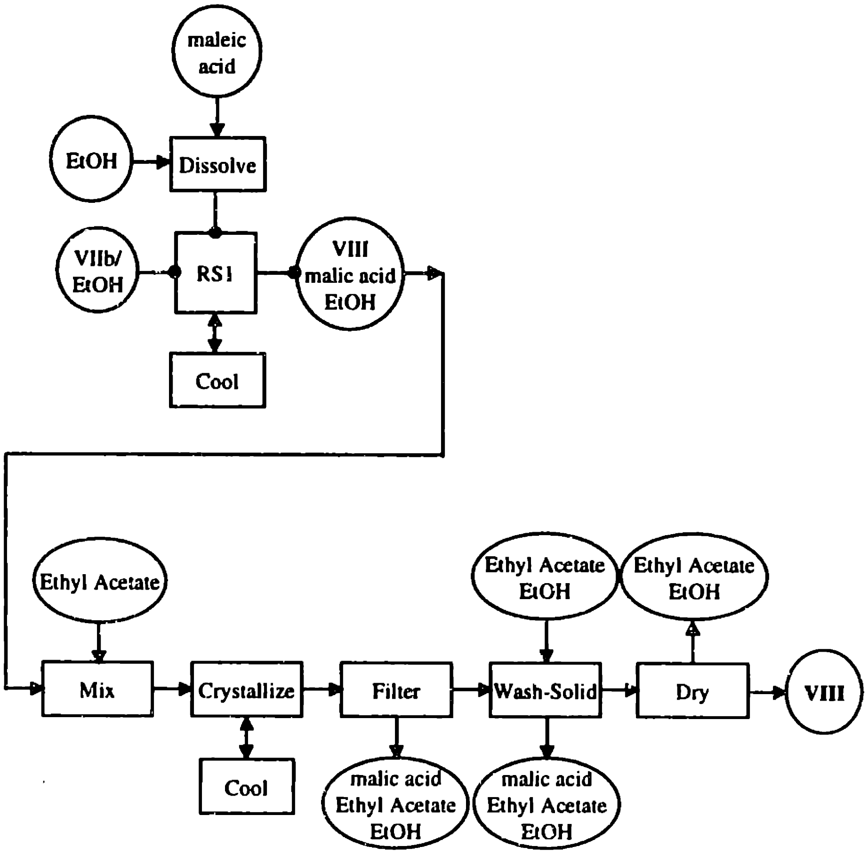


Figure 5.35 Stage 9 Complete Product Train (Yield = 93%)

5.11 Observations

In this chapter, we have synthesized a processing scheme for the production of a pharmaceutical compound using the MEA-NMP methodology. The case study has demonstrated the ability of the MEA-NMP methodology to synthesize a process which closely resembles that developed by the chemist. In order to reproduce the process developed by the chemist, during the operation selection step, we selected the specific operation chosen by the chemist. Obviously, had we selected alternate operations from within the Task Categories, we would have developed an alternative process. This is the simplest method for generating alternative processes. It should be clear that an optimization routine could be constructed using the Task Categories to construct a superstructure of alternatives in order to identify the optimal process.

During the development of the Stages within the case study we made some additional observations. First, as we have stated, the MEA-NMP methodology is capable of generating processes that closely resemble that of the chemist. This is important since currently within industry, the chemist's process is taken as the base case process and the ability of the MEA-NMP to produce processes with resemble the chemist's processes should provide some confidence in the viability of the methodology. Second, we can generate many alternative designs though the selection of different operations from within the Task Categories. The Task Categories provide us with a reduced set of operations which can perform the desired task. This reduces significantly, the search space of alternative operations that can be applied to resolve the differences between the current state and the goal state. Finally, we have seen the flexibility of the methodology in allowing the developer to insert operations into the process plan which may not be identified by the MEA-NMP methodology, due to the simplicity of the operation models. This was demonstrated in Stage 2, where the developer found in necessary to add aging, heating and cooling operations. We should note that by adding more sophisticated models of the operations and by expanding the Task Category selection logic many of the operations which where not identified in this case study, would be identified. Also, a number of the operations, such as *age*, *heat* and *cool* are closely related to the operating conditions of the operation being applied. By using operating condition information, these operations can be identified by the methodology. In Chapter 6, we discuss the application of the MEA-NMP methodology to the creation of superstructures for optimization.

Chapter 6

MEA-NMP: The Impact on Optimization

In Chapter 4 we discussed the application of the MEA-NMP methodology to create a base-case design, but the ultimate goal of process synthesis is to develop an optimal design for producing the desired product. This invites the question, can one actually develop an optimal design, i.e. is the identification of an optimal design a realistic proposition, given the available information? And if so, how would one go about developing an optimal process, and if not, how does one define the realistic scope of the design problem? In Chapter 3 we discussed the formulation of the batch process synthesis problem mathematically and provided an overview of a number of optimization techniques, which could be used to solve it. We also showed that the batch process synthesis problem is a large combinatorial problem that cannot be realistically handled with conventional optimization methods due to the immense complexity induced by the integer decision and the nonconvexities of the process models. In this chapter, we will show how the MEA-NMP approach to process synthesis can be extended in order to create superstructures of feasible alternatives only, thus reducing the complexity of the problem and facilitating the selection of the optimal batch process design.

In Section 6.1 we begin by showing how the MEA-NMP methodology can simultaneously create a superstructure of operations along with the base-case design. The superstructure makes it possible to consider the application of combinatorial optimization techniques to the process synthesis problem. It is this superstructure to which the optimization routines will be applied in order to identify the optimal design. In Section 6.2 we will present the potential problems that are inherent with attempting to find an optimal design along with the problems that are faced in using the MEA-NMP methodology in conjunction with optimization routines. In Section 6.3, we will discuss how the optimization problem would be formulated using MEA-NMP in conjunction with optimization. In Section 6.4 we discuss the role of logic in the MEA-NMP methodology. Superstructure reduction is discussed in Section 6.5 and finally, in Section 6.6 we perform a comparison of alternative processes.

6.1 MEA-NMP Superstructure Generation

The generation of a superstructure is a central concept in the optimal synthesis of batch processing schemes. A superstructure is a representation of the synthesis problem that contains all possible alternative solutions to the problem embedded within it. The most general superstructure would contain all possible alternative components for the construction of all possible solutions to the problem embedded within it. It is the superstructure that defines the search space of alternatives for the optimization problem. We use the superstructure to attempt to define and reduce the search space to which the optimization routines will be applied. An example of a general superstructure constructed using the state task network format is given in Figure 6.1. From the figure we see that the superstructure contains eight (2^3) possible flowsheets that can bring us from State-1 to State-G.

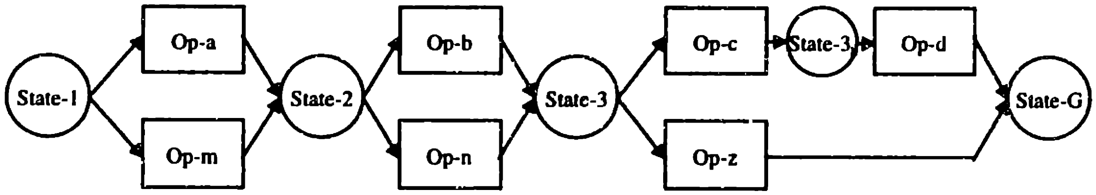


Figure 6.1 General Superstructure.

However, the construction of a meaningful superstructure is not a trivial task. In other words, one should not include all permissible operations and connect them in all possible ways, since the resulting superstructure contains not only the optimal but a very large number of logically inconsistent and thus, infeasible alternatives. Indeed, in this Chapter we will show that not all combinations are feasible solutions. In creating the superstructure for the batch synthesis problem, we would like to include *all* the feasible and *only the* feasible alternatives. This will be accomplished by the expanded application of the MEA-NMP methodology.

All previous optimization strategies for the synthesis of process designs have used ad hoc methods incorporating knowledge of the process in order to create a superstructure of feasible alternatives. Prior to this thesis, only one method has been proposed for the systematic synthesis of the process superstructure. Friedler et al [1] have proposed a method for the systematic construction of "maximal structures" which they claim are superstructures. Actually, the method of Friedler is more of a screening method than a superstructure generation method. The method of Friedler will be discussed in more detail in the superstructure reduction section. The problem with an ad hoc construction of the superstructure is that there is no systematic way to ensure that only the infeasible alternatives have been eliminated and that all of the feasible alternatives remain. There is a strong possibility that along with eliminating infeasible alternatives, the ad hoc methods may inadvertently eliminate non-obvious feasible alternatives due to bias or incomplete knowledge of the system. In the following two sub-sections, we present a method for the systematic construction of process superstructures and illustrate its use on the Carbinol case study.

6.1.1 Systematic Construction of the Superstructure

The MEA-NMP methodology allows us to systematize the construction of the superstructure of alternatives based on the type of operation to be performed. The superstructure is constructed simultaneously with the construction of a base-case design. The base-case design provides an upper bound on the minimum cost for the process. Any alternative process that is generated and has a cost greater than the base-case can be eliminated. As each alternative operation is selected during the construction of the base-case design, a parallel superstructure flowsheet is constructed using the task category from which the operation was selected as shown in Figure 6.2. In the figure, we have left out the intermediate states but we can envision that each arc connecting two tasks passes through an intermediate state. Since the task categories are subsets of the entire set of operations, they provide a convenient method for the reduction of the search space of alternatives by adding to the superstructure only the operations that are capable of performing the required task. Notice that we did not say that the task categories contain only the feasible alternatives since the feasibility can only be assessed after the preceding operation has been applied and the current state defined. This leads to another very important point, the intermediate states are created dynamically through the application of the operations, which implies that each node in the plan cannot be optimized independently. The use of the task categories avoids the potential for prematurely eliminating non-obvious feasible alternatives by forcing the optimization to examine all the alternatives that have the potential to perform the required task.

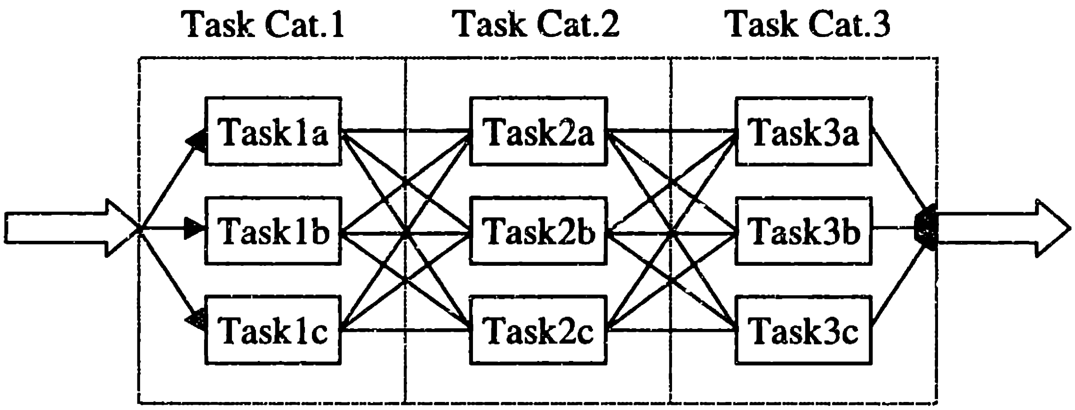


Figure 6.2 Task Category Construction of the Superstructure.

If we compare Figure 6.1 with Figure 6.2 we see that in some cases multiple operations can be used to perform a given task. This is a common occurrence in process synthesis and the MEA-NMP methodology is able to handle this situation. Operations at different levels of abstraction can be contained within a task category, thus an abstract operation can be placed into the plan and later it can be replaced with its detailed operations or the abstract operation can be viewed as a task sub-category and the detailed operations can be added directly into the plan at the time of construction. Illustration 6.1 provides an example of the use of sub-categories and abstract operators.

Illustration 6.1

An example of the addition of an abstract operator into the plan followed by its replacement with the detailed operations is given by the operation *solvent replacement.* The task category *SolventReplacement* contains an operation called *solvent replacement* which simply exchanges one solvent for another without regard to how it can be performed. Later, we can go back and select from the list of operations that can actually achieve a solvent replacement and replace the abstract operator with the detailed operations such as *concentrate and mix* or *extract.* We see in the combination of *concentrate and mix,* that in order to achieve a solvent replacement using detailed operations we have an option to use a combination of operations that produce the required effect. In order to achieve a solvent replacement using the *concentrate* operation we need to follow it with the *mix* operation. Figure 6.3 shows the abstract solvent replacement operation and below it the detailed operations that can be used to actually perform the solvent replacement.

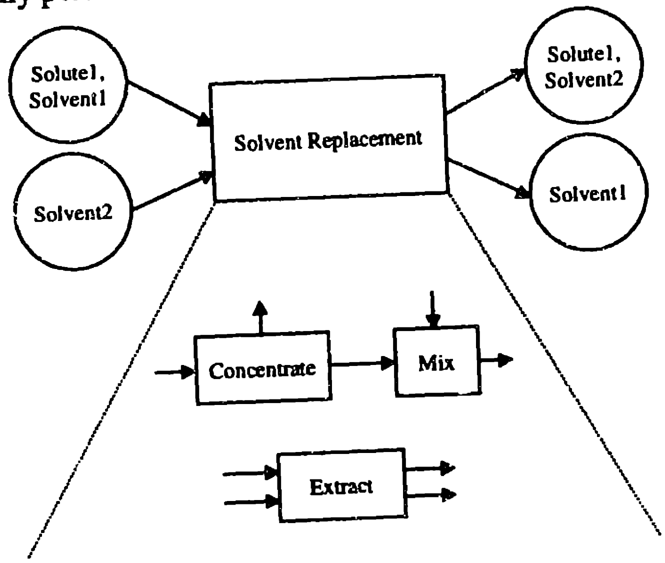


Figure 6.3 The Solvent Replacement Operation and its Detatiled Operations.

6.1.2 Generation of the Superstructure for the Carbinol Stage Using MEA-NMP

Let us now show how the superstructure is constructed simultaneously with the base-case design. To do this we will revisit the Carbinol stage that was developed in Chapter 4. Reviewing the MEA-NMP methodology (see Figure 4.16), we see that after we identify the existence of a difference and screen for critical features, the next step is to identify a task category which contains the operations that can eliminate the difference. The first task category identified in the Carbinol stage is the *MolecularStructureTransformation* task category used to eliminate the identity difference between the raw materials and the goal state. We then select the *react* operation and apply the interactive generation of the preconditions. The interactive generation allows the simultaneous generation of the preconditions and the identification of the white knight operations to eliminate the precondition violations. In many cases, in the reaction system construction, we identify a specific operation in order to eliminate a precondition violation without first selecting a task category. This does not hinder the construction of the superstructure, we simply place the operation into the superstructure. This corresponds to having a task category that only contains a single operation and simplifies the optimization by fixing the decision. Figure 6.4 contains the superstructure for the product train of the Carbinol stage produced from the example in Chapter 4. Notice that the *dissolve* operation has been placed in the superstructure before the *MolecularStructureTransformation* task category. Reviewing the development of the Carbinol stage in Chapter 4, we collect all the task categories that were used in the identification of the operations for the base-case design. It is important that we maintain the ordering in which the task categories were selected because this provides the structure of the superstructure. Each block within the superstructure represents either a task category or a specific operation. The name of the task category is underlined and the operations that belong to the task category are contained within the blocks. Notice in the superstructure that the task category *SolubilityReduction* is followed immediately by the sub-category *SolventReplacement*. What this implies is that the optimization of the solubility reduction operation can include either the abstract operation solvent replacement or the optimization can be performed over the combined list of operations from *SolubilityReduction* and *SolventReplacement*. Also, notice that a number of states are dashed. These states are added only if an operation within the corresponding task category requires an additional component. For example the *dissolve* operation requires the specification of a solvent.

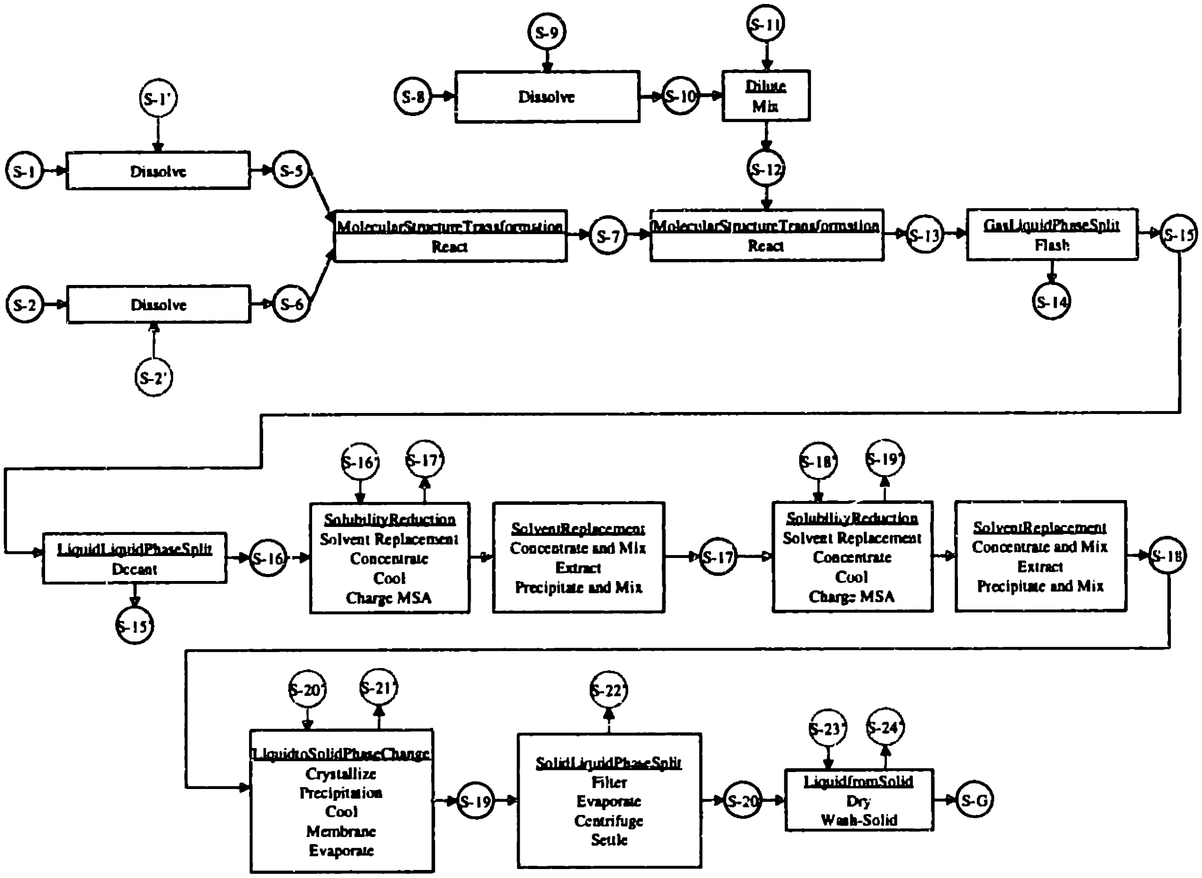


Figure 6.4 Superstructure for the Carbinol Stage.

Examining the figure, we see that a number of task categories contain only one operation which greatly reduces the number of combinations of possible flowsheets. Even with this relatively simple superstructure there are 1,960 alternative processes. To see how the superstructure has reduced the search space for the problem, lets construct the worst case scenario which assumes we perform the search over the entire space of alternatives. The total number of different batch processing operations that can be performed at each node is approximately twenty-five. In the worst case, we would need to consider every operation for each node in our plan. Thus, we must perform an optimization over the twenty five operations for the thirteen nodes in the plan. This is an optimization problem that needed to search over $1.490*10^{18}$ alternatives. Clearly this is an extreme example. If we limit the search to just the nodes that have multiple operations, allowing the fixed operations such as the *dissolve* operation above to remain fixed, the number of alternative flowsheets drops to $6.103*10^{9}$. This still exceeds the scope of any optimization method. These examples show that the superstructure is an important element in bringing the process synthesis problem into a tractable format. It is clear that the structuring of the problem using task categories provides a substantial reduction in the magnitude of the optimization problem. Additionally, the MEA-NMP methodology also creates a constraint on the maximum number of nodes that are to be included in the plan. The size of the base-case limits the size of the superstructure. An unconstrained optimization may produce flowsheets of substantially larger number of operations.

6.2 Problems with Superstructure Optimization

While some researchers have claimed success with superstructure optimization, the number of practical problems that have been solved is small. As far as we have been able to determine, no one has yet to perform a full process synthesis optimization of an industrially relevant batch chemical process. Even though construction of the superstructure using the task categories substantially reduces the size of the optimization problem, the problem is still a large combinatorial optimization. There are also other factors which must be considered before we can successfully apply an optimization routine and develop an optimal design. The two main factors providing difficulty in performing the optimization are (i) the problem size and complexity and (ii) obtaining the necessary data required for the optimization. The remainder of this section will focus on these two issues and discuss the implications of these problems with respect to optimal process synthesis.

6.2.1 The Problems Caused by Problem Size and Complexity

From the Carbinol stage, we see that the task categories reduce the problem size from that of billions of alternatives to an optimization of under two thousand alternatives. This is still a large number of alternatives but is within the scope of today's optimization capabilities. The problem lies in the structure of the problem which we are trying to solve. If we combine the problem of selecting the operations to achieve the desired goal, the nonlinear models that compose each operation, the various types of constraints for each operational model, the operating conditions as well as the sizing of the equipment, we see that the resulting problem is highly nonlinear and nonconvex. It is the nonconvexities that diminish our ability to determine an optimal solution to the problem. The nonconvexities add the complexity to the problem of determining whether the solution is a local optimum or a global optimum. Currently, there are optimization techniques that can guarantee that an optimal solution will be found if the problem contains nonconvexities, but they are not robust and are quite cumbersome. All superstructure optimization problems that have been solved up to now have been small and highly structured. The specific structure of certain problems allows the global optimization, even if they contain some nonconvexities [2] but for the general case there is no convincing evidence on the efficiency and robustness of the proposed algorithms.

> *Illustration 6.2*
>
> The HDA continuous process has been studied extensively by various authors [3],[4]. Grossmann [4] has developed a simple superstructure for the process based on the design decisions of Douglas [3]. This superstructure has almost 200 different process flowsheets embedded within it. Using simplified nonlinear models, the MINLP that was formulated consisted of "13 0-1 variables, 672 continuous variables and 678 constraints (140 nonlinear equations, 567 linear equations, 71 linear inequalities)." Depending upon the solutions algorithm used to solve the optimization the solution time was between 2 min and 8 min using an IBM-3083. While the HDA process does contain recycle streams it is considered a relatively small problem that is highly structured allowing the optimal solution to be computed based on the proposed superstructure.

Illustration 6.2 also shows how the complexity of the process can create difficulties when trying to perform an optimization. Almost all optimization problems are formulated using simplified models and a reduced set of constraints. If this is the case, then can we really call the resulting design an optimal design? In order to determine the true optimal design, we must take into account as much detail as possible for the process including reaction models, equipment specifications, raw material specifications, models of operating conditions as well as operation models. We must also be able to clearly state the objective function, which we would like to solve along with the constraints on the problem. These must be highly detailed and if we are performing an optimization for batch processes, the models for the operations and operating conditions must also be dynamic to capture the unsteady state of most batch operations. The field of dynamic optimization is taking on increasing interest [5]. We need to understand that the designs that we call optimal are only optimal with respect to the level of modeling applied and the assumptions made in formulating the optimization problem. The designs that are called optimal are simply good designs (although, since we cannot solve for the true optimal design, we do not actually know how good they are compared to the true optimal design). We must be cautious to ensure that we understand how our assumptions and simplifying models impact our design and then ask whether this design is good enough for our purposes.

6.2.2 Problems Due to Insufficient Data

It should be clear that, along with all the complexity of the models involved and the tasks to be achieved, a large amount of data is required to support the models. Even simplified models require substantial amounts of data in order to accurately model the behavior of the tasks and predict the resulting state of the system. For example, many processes contain a crystallization operation, yet the methods for predicting the properties of solids, such as solid solubility and the presence of eutectics, are poor. Ideally, we would like to have experimental data over the operating range of interest for our system. Unfortunately, the information we have available is rarely complete. The synthesis chemist usually has just a few data points which were collected during the synthesis experiments. To collect additional data points can be quite expensive and time consuming. Since the capability of prediction tools cannot handle the systems of interest, we must rely on extrapolating the available experimental data and rely on the knowledge of the developer to provide estimates for the required properties. Clearly, the quality of the data will impact the ability to predict the resulting states of the system as well as impact the modeling of the operations and will impact the ability to determine the optimal design. The lack of data is perhaps the most significant problem impeding the use of optimization in process synthesis.

Illustration 6.3

In order to model a crystallization operation we need to have an estimate of the solid-liquid equilibrium with respect to temperature. Rarely do we just have a single component dissolved in a solvent. Usually, crystallization is the desired method of separation because a number of solids are dissolved in solution and we would like to isolate a single pure compound. Figure 6.5 shows the polythermal solid-liquid phase diagram for the p-Xylene, m-Xylene, carbon tetrachloride system. We see from the figure that the system contains three binary eutectics and a ternary eutectic. Without the general structure of this diagram it would be very difficult to develop a crystallization scheme to isolate p-Xylene from m-Xylene. The SLE data is only part of the information required to perform the optimization of a crystallization operation. We need information on the kinetics of the system with respect to crystal formation and growth rate, which are generally functions of the operating conditions of the system as well as the starting state of the system. All of this information is not readily available and the lack of this information makes the optimization not a realistic proposition.

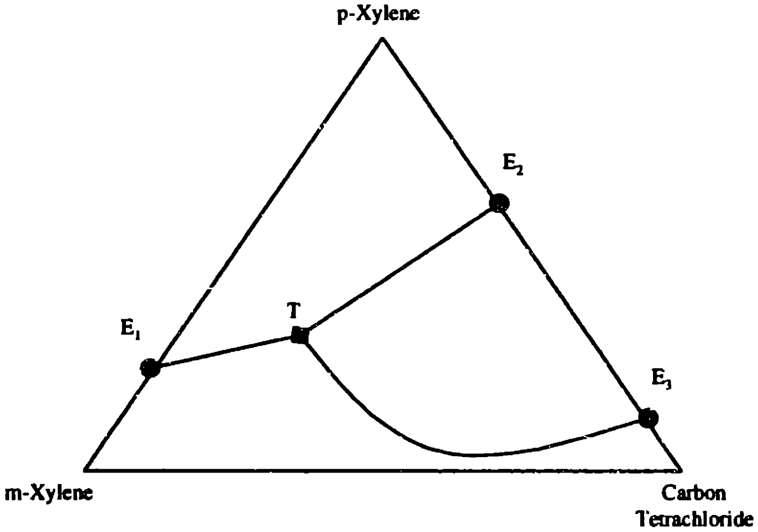


Figure 6.5 p-Xylene, m-Xylene, carbon tetrachloride system from Rajagopal et al [6].

6.2.3 Implications of the Problems on Optimal Process Synthesis

We have examined the problems caused by complexity problems as well as by the lack of sufficient data. What this implies for optimal batch process synthesis is that in order to achieve an optimal design, significant effort must be made to limit the complexity of the problem as well as to obtain the required data. This section will review how this impacts the search for the optimal design.

6.2.3.1 The Trade-Offs involved in Determining an Optimal Design

As discussed above, determining the optimal design requires the availability of large amounts of data. If this data is unavailable at the time the design task is being performed, this data must be collected by conducting experiments. The experimental collection of data is both costly and time consuming. This creates our first trade-off, do we settle for a sub-optimal design or do we stop and do the experiments and then continue with the design process to try to obtain an optimal design? Another interesting feature of this trade-off is that there is no way to incorporate these costs into the objective function for the design process even though these are significant costs with respect to the profitability of the process we wish to develop. For example, it has been estimated that each day a pharmaceutical product spends in the development phase costs approximately \$1,000,000 [7]. We see that there is a substantial cost associated with the time it takes to collect any additional data apart from the actual cost of collecting the experimental data itself. It should be clear that it would not require very long for the cost of collecting additional experimental data to overshadow the benefits of a detailed optimization. This point leads directly to the next trade-off, increased complexity versus degree of optimality. The more detailed we make our model of the system we wish to optimize, the more difficult it becomes to solve the optimization problem. This is due to the increase in number of variables and the possible increase in nonconvexity of the system. Of course, the added detail makes the solutions we find more accurate. We need to weigh this added accuracy against the cost of attempting to solve a more complex optimization problem. Additionally, more detailed models generally require more data to feed the models. This then increases the cost of data collection. On the other hand, we can choose to use simplified models to construct the problem at the expense of losing accuracy in our solution.

What do these trade-offs mean to the optimization problem? First is that solving a detailed complex optimization problem is probably a waste of time and money. Due to the nonconvexities in the problem, we have no way of knowing whether or not we have achieved the optimal design. Without having some termination criterion, the only way to be sure we have obtained the optimal design would be to enumerate all alternatives, which is clearly impossible. Also, the cost and time required to collect all the required data to perform the detailed optimization is prohibitive. What we can do however, is use our knowledge of the data that is available about the specific process to further reduce the superstructure that is developed along with the base-case design in the MEA-NMP methodology. Once we have a narrowly defined superstructure for which we know that we have the required data or the ability to predict the required data, we can then apply any of the various optimization routines and determine a near optimal solution. If we are able to find a solution to the optimization problem, it is an optimal solution within the search space we have defined. Of course, if the models that we use are nonconvex, we are still not guaranteed to have found the global optimal solution but the time spent constructing the problem and searching for the solution will have been greatly reduced. We must then determine if the solution obtained is of sufficient quality to process with the implementation. In the next section we will discuss the formulated of the objective function for the process synthesis optimization problem.

6.3 Formulation of the Optimization Problem Using MEA-NMP

In this section, we will go through the steps for formulating the optimization problem beginning with the superstructure generated using the MEA-NMP methodology. The purpose of this section is to describe how one would go about formulating the reduced optimization problem. The detailed optimization will not be performed. Floudas [2] and Grossmann et al [4, 8, 9] have published numerous examples of superstructure optimization in process synthesis and the reader should consult these sources for examples of solutions for superstructure optimization problems.

In order to complete the formulation of the optimization problem, we need to develop the objective function, develop the models and constraints for the problem and attempt to reduce the superstructure. In the next sub-section we will describe the formulation of the objective function. Following the development of the objective function, we will discuss the construction of the models describing the operations and constraints.

6.3.1 Formulation of the Objective Function

The objective function is the function we wish to minimize or maximize in our optimization problem. Generally, objective functions are formulated to either maximize profit or minimize cost. When comparing alternative processes for the same product, formulating the objective function to minimize cost is preferred. The minimization of cost eliminates the product value from the objective function. This is necessary when dealing with such high value added products like pharmaceuticals. The value of the product is generally so high that it makes the cost of production seem irrelevant. When comparing projects that produce different products or in the cases where the product value has the same order of magnitude as the cost of production, as may be the case for some toll manufacturers, it is better to use the maximization of profit as the objective function. Since we will be comparing processes for the production of a single product, we will formulate the objective function in terms of cost minimization.

6.3.1.1 The Components of the Objective Function

Generally, when costing a batch process it is desirable to express the costs in terms of dollars per kg of product. In order to obtain the \$/kg of product, we need to consider many cost components. Some of the main cost components are raw material costs, the equipment costs, utility costs, overhead and labor costs and process waste costs. Each of these cost components is made up of a number of sub-components. For example, for the construction of a grass roots facility to produce porcine somatotropin, Cooney [10] provides tables with the break down for the Fixed Capital Estimate and the Annual Operating Cost at 1991 prices. These tables are reproduced below in Table 6.1 and Table 6.2, respectively.

A.	Total Plant Direct Cost (TPDC)		$	% of DFC
		(PC = Physical Cost)	$ 6272000	13.8 %
1.	Equipment Purchase Cost	(PC)	$ 2509000	5.5 %
2.	Installation	(0.40 * PC)	$ 2195000	4.8 %
3.	Process Piping	(0.35 * PC)	$ 3763000	8.3 %
4.	Instrumentation	(0.60 * PC)	$ 188000	0.4 %
5.	Insulation	(0.03 * PC)	$ 941000	2.1 %
6.	Electrical	(0.15 * PC)	$ 4704000	10.4 %
7.	Building	(0.75 * PC)	$ 941000	2.1 %
8.	Yard Improvement	(0.50 * PC)	$ 3136000	7.0 %
9.	Auxiliary Facilities	(0.50 * PC)	$ 24649000	54.4 %
		TPDC =		
B.	Total Plant Indirect Cost	(TPIC)		
10.	Engineering	(0.25 * TPDC)	$ 6162000	13.6 %
11.	Construction	(0.35 * TPDC)	$ 8627000	19.0 %
		TPIC =	$ 14789000	32.6 %
C.	Total Plant Cost	(TPDC + TPIC)	$ 39438000	87.0 %
12.	Contractor's fee	(0.05 * TPC)	$ 1972000	4.3 %
13.	Contingency	(0.10 * TPC)	$ 3944000	8.7 %
D.	Direct Fixed Capital (DFC)			
		DFC = TPC + 12 + 13 =	$ 45354000	100%

Table 6.1 Fixed Capital Estimate – Porcine Somatotropin (1991 prices)[10].

A. DFC-Dependent Items		\$/yr	% of TAOC
1. Depreciation		\$ 4309000	19.5 %
2. Maintenance Material	(0.03 * DFC)	\$ 1361000	6.2 %
3. Insurance	(0.01 * DFC)	\$ 454000	2.1 %
4. Local Taxes	(0.02 * DFC)	\$ 907000	4.0 %
5. Factory Expense	(0.05 * DFC)	\$ 2268000	10.3 %
		\$ 9299000	42.1 %
B. Labor-Dependent Items			
1. Operating labor	(19 * \$32000)	\$ 608000	2.8 %
2. Maintenance labor	(0.03 * DFC)	\$ 1361000	6.2 %
3. Fringe benefits	(0.40 * (1 + 2))	\$ 788000	3.6 %
4. Supervision	(0.20 * (1 + 2))	\$ 394000	1.8 %
5. Operating Supplies	(0.10 * 1)	\$ 61000	0.3 %
6. Laboratory	(0.15 * 1)	\$ 91000	0.4 %
		\$ 3303000	15.1 %
C. Administration and Overhead			
	$(0.6 * (B_1 + B_2 + B_3))$	\$ 1418000	6.4 %
D. Raw Materials		\$ 543000	2.4 %
E. Process Chemicals & Consumables		\$ 3634000	16.5 %
F. Utilities		\$ 252000	1.1 %
G. Waste Treatment		\$ 628000	2.8 %
H. Running R&D		\$ 3000000	13.6 %
I. Running Royalties		\$ 0	
J. Sales Cost		\$ 0	
Total Annual Operating Cost	(TAOC)	\$ 22077000	100 %

Table 6.2 Annual Operating Cost - Porcine Somatotropin (1991 prices)[10].

We see from Table 6.1 that many of the *Direct Fixed Capital* (DFC) costs are estimated as a percentage of the physical cost of purchasing the equipment. Thus, it is important that we have reliable estimates of the cost of equipment for the process. Generally, equipment costs for batch processes are best obtained from the venders of the equipment but for the early stages of process development and design, it may be sufficient to use correlations such as Guthrie's cost correlations [11]. From Table 6.2, we see that the

Total Annual Operating Cost (TAOC) contains many more components than just raw material, labor, utility and waste treatment costs. Depending upon the costing methods of the particular firm, the break down of the costs may be as in Table 6.2 or may have additional or fewer components and, most certainly, each company has its own multiplying factors depending upon their costs.

In this thesis, we are looking to provide quick estimates of the costs for the process so that we can rapidly screen a number of alternatives. Thus, we need not include all the detail shown in the tables above in the formulation of the objective function. The cost components we will include in the objective function include the raw material costs, the equipment costs, utility costs, overhead and labor costs and process waste costs. Before we formulate the objective function, we must have an understanding of how to obtain each of these costs and the proper metrics to use when evaluating the value of each of these components. Let us examine these cost components and then show how they are combined to form the objective function.

Raw Material Costs

One of the primary components of the objective function is the raw material cost. The raw material prices are fixed values or price functions that are taken from databases of prices extended to the company by its suppliers. The raw material costs for the process are a function of the amount of raw material consumed by the reaction steps and the amount of raw material that can be recovered and reused. We see that *conversion*, *selectivity* and *material recovery* costs all impact the contribution of raw material costs in the objective function and also constitute various trade-offs. For example, if the process is run at a low conversion, a large amount of raw material remains at the end of the reaction steps. In many cases it will be beneficial to recover and recycle the unreacted raw materials which will increase the load on the raw material recovery system. If the conversion is high, and the selectivity is low we have a large amount of raw materials going to form by-products which increases the cost of the separation system and waste disposal. The objective function must reflect these trade-offs. Equation 6.1 gives the amount of raw material needed per batch. Equation 6.2 contains the cost function for a raw material.

$$F_{Raw\,Material}=\frac{P}{x f_r S S_r g \frac{MW_{Prod}}{MW_{Raw\,Material}}} \qquad (6.1)$$

where:
$F_{Raw\ Material}$ = Raw material feed (kg/batch)
P = Production ($kg_{product}$/batch)
x = Conversion
f_r = Product fractional recovery
S = Selectivity
S_r = Stoichiometric ratio
g = Feed ratio
MW_{Prod} = Molecular weight of the product
$MW_{Raw\ Material}$ = Molecular weight of the raw material

$$C_{Raw\,Material}=(F_{Raw\,Material})M_{Raw\,Material} \qquad (6.2)$$

where:
$C_{Raw\ Material}$ = Cost of raw material
$M_{Raw\ Material}$ = Price of the raw material

If we look at Table 6.2 we see a component called Process Chemicals & Consumables. This component is similar to the raw materials but it is used to describe materials such as solvents and catalysts which are not recovered and recycled to the process. In many biochemical batch processes we do not recover and recycle the solvents and catalysts since there is concern about cross contamination. For most non-pharmaceutical processes we can recover and recycle materials such as solvents. Equation 6.3 contains the cost component related to any material which can be recovered and recycled. We should note that this cost component does not include the required equipment or utilities for the recovery of the material. Those costs will be included in their corresponding components respectively.

$$\begin{aligned} C_{Process\ Chemical} = {} & \text{(Amount used in process)(Price of material)} \\ & - \text{(Amount recovered)(Value of recycle quality material)} \end{aligned} \qquad (6.3)$$

Equipment Costs

The equipment costs are a function of the operations that are included in the process and are a function of the required volume in order to perform the operation. Additionally, the equipment costs are a function of the plant allocation i.e. is the process going to be allocated to an existing facility or is a new facility going to be constructed.

The equipment cost component of the objective function contains binary variables that indicate whether or not the piece of equipment is included in the process. The value zero indicates an operation and its corresponding piece of equipment is not included in the objective function, while a value of one indicates the inclusion of the operation and equipment in the design. The problem here is the concept of task merging. For continuous processes, generally, there is a direct one-to-one correspondence for operations and the equipment, i.e. each piece of equipment will perform a single operation. For batch processes this is not the case. Many operations can be performed in a single piece of equipment. For example, a reaction can be performed in the still of a batch distillation vessel or a crystallization can be performed in a filtration vessel. This means that we need to integrate task merging into the optimization so that we can also determine the optimal amount of equipment along with the optimal set of operations. The following illustration demonstrates why this is not a simple task.

> *Illustration 6.4*
>
> This illustration will show that task merging brings elements from the process scheduling problem into the process synthesis problem further complicating the process synthesis problem. Lets we assume that we have a simple process consisting of three operations, a reaction and two separation steps. Let us also assume that we can either perform all the operations in a single vessel, perform each of the operations in a separate piece of equipment or perform two operations in one piece of equipment and one operation in another. We see that the number of possible configurations is four. If one took a simple view of the process we could say that performing all the operations in a single piece of equipment would minimize equipment costs. It should be obvious that this is incorrect. Using a single piece of equipment could be the lowest cost if we were only producing a single batch of product, the problem becomes one of scheduling once we determine the need to produce multiple batches. The use of multiple pieces of equipment allows us to begin parallel processing by starting the next batch before the previous batch is complete. We cannot explicitly say which of the four equipment configurations is best without including the scheduling problem in the synthesis optimization problem.

In order to incorporate task merging into the optimization, rules must be developed which guide the task merging by indicating what types of tasks can be merged into a single piece of equipment. The optimization must then include process alternatives which merge tasks and those which do not in order to determine the optimal process.

How does one evaluate the equipment cost of a process? The answer is simple if we are developing a grass roots plant or re-equipping a facility, we simply determine the capital cost for the purchase of new equipment. The issue becomes more difficult if we are going to examine placing the process in existing facilities. At the process synthesis stage of development we may know that we are going to produce a specific product or we may have competing projects and we are trying to determine which project the company should commit to. We see that the task of deciding which project to proceed with along with which facility we should use for the project are interrelated. There are four possible scenarios:

1. Comparing competing projects that are going to be placed in new facilities.
2. Comparing competing projects that are going to be placed in a specific existing facility.
3. Comparing the placement of a single project in different existing facilities.
4. Comparing competing projects that could be placed in different facilities.

When evaluating a process we must make a distinction between existing facilities and new facilities. Note that when we say a new facility, this refers to either the construction of a new plant or the complete re-equipping of an existing facility. Obviously, we could also make a distinction between the construction of a new facility and the re-equipping of a facility since a new facility would have additional costs related to the construction of the building. While these additional costs can be quite significant, the distinction will not change the analysis for comparing processes. We make the distinction between the existing facilities and new facilities because the costs associated with each are different. For a new facility, we must determine the cost of purchasing all the required equipment. Charging the entire cost of the new equipment to a single project in a multipurpose batch plant would unfairly place the burden of paying for all the equipment upon a single project even though that project will only be using the equipment for a fraction of the equipment lifetime. The project should only be charged for the fraction of the cost of the equipment which the project actually uses. This assumes that the equipment required for the project is not specific only to that project and can be used for other projects. We can try to reconcile this problem simply, by taking the cost of the equipment and multiplying it by the ratio of time spent using the equipment for this project to the life expectancy of the equipment. While this can provide a quick approximation of the cost of the equipment, the actual costing of the equipment is more complicated and must include factors like the depreciation in value of the equipment over the life of the equipment, the desired capital recovery rate, the cost of capital, etc. Since the inclusion of these factors increases the complexity of the analysis without changing the results, it will be sufficient to use the simplified cost of equipment during the process synthesis. Equation 6.4 contains the simplified equipment cost component and at this stage of process development we choose not to include installation costs since this is simply a percentage of the purchased cost and for all processes it would be the same percentage of cost. As mentioned earlier, during the early stages of process development we can use cost correlation models based on equipment type and volume in order to obtain the purchased cost for the equipment items or alternatively if we have a database of equipment items and cost on hand we can use those values.

$$C_{Equipment} = PC_{Equipment}\left(\frac{\text{Time for Process}}{\text{Life Expectancy}}\right) \tag{6.4}$$

where:
$PC_{Equipment}$ = Purchased Cost

Determining the cost of equipment usage for an existing facility is more complicated. Many companies develop an internal rental charge for the usage of equipment while others develop a usage charge for an entire facility. Which one of these methods is best for evaluating the cost of a given process? Some would argue that the rental charge per piece of equipment is the way to go since the process is only charged for the equipment that is actually used. This is true, but if a project is going to take place in a facility and no other projects can be run in that facility during that time period, the project is in essence tying up all the equipment for that facility. The problem with using rental costs of any type is that rental charges may carry with them the costs of inefficiencies of previous processes and then that process is unfairly charged to make up for short falls of previous processes.

In order to avoid making the decision of which type of costing structure to use at the conceptual stage, we suggest costing all processes based on a fraction of new equipment cost as described above. This will provide a fair basis for comparing competing projects and designs. Of course, once we have narrowed the number of possible projects under consideration, we must go back and assess the actual cost to the company based on the nature of the project i.e. constructing a new facility or using an existing facility. For example, lets assume that we have two competing projects and the value of the final products and the cost of the raw materials is approximately the same for both. We may determine that on a fraction of new equipment basis, that Project A has a lower cost then Project B. We can envision a situation when we attempt to actually allocate Project A and Project B to existing facilities within the corporation, we may have a facility that is completely ready to accommodate Project B while all of the facilities may require significant modification in order to accommodate Project A. In this case, even though on a new equipment basis Project A has a lower cost, we would want to select Project B since the actual cost to the company will be lower.

To summarize, for each of the four scenarios presented above, we will construct the optimization problem using a fraction of new equipment cost basis. When comparing competing projects that are going to be placed in new facilities, we can use the objective function as is without performing a secondary optimization. For the case of comparing competing projects that are going to be placed in a specific existing facility, we can first perform the optimization using the fraction of new equipment cost basis and then once the number of projects has been narrowed to just a few under consideration, we can apply the actual costs incurred by placing those projects in the specific facility. If we are comparing the placement of a single project in different existing facilities, we may want to evaluate different designs for the process using the fraction of new facilities equipment cost in order to have a direct comparison basis and then after we have selected the most promising designs we can begin the search over the existing facilities to determine which is more economical. Finally, if we are comparing competing projects that could be placed in different facilities, we again use the fraction of new facility equipment cost as a basis for comparing alternatives of the different processes and then once we have selected the most promising we can then go ahead and re-evaluate the processes based on the actual costs associated with the different facilities.

Utilities Costs

The utilities costs in batch processes are rarely a significant component of the cost function. More important than the cost of the utilities, is the availability of the utilities. If a process calls for more utilities than are available, the process is infeasible for that facility. For a preliminary evaluation of the utility costs, we can use quick estimates of utility usage based on the heating and cooling duties of the operations selected in the process multiplied by the cost of the type of utility that can provide the required energy flow. If the costs appear particularly high for a given alternative, we can then go back an perform a more detailed utility cost analysis before selecting the process.

A better solution would be to add utilities usage as a constraint on the optimization problem. For example, we can set the maximum usage of a given utility at a specific value, such as the maximum amount of high pressure stream available to the process is x kg/hr. The utility availability is a function of the facility, with some facilities having different utilities and different amounts of utilities available. Again, as with the facility allocation for the equipment, once we have optimized the process we must search over the various facilities that are available to ensure that the utility demand for the process can be met.

In the development of the objective function, we can derive estimates of the utilities required for each operation based on the operating conditions for that operation and the amount of material involved. We can then take the required utilities and apply a price using a table such as Table 6.3 provided by Douglas [3]. Equation 6.5 contains the utility cost component.

$$C_{Utilities} = \sum_{Operations} P_{Utility}(Q_{operation}) \tag{6.5}$$

In Table 6.3, Douglas provides us with factors relating the price of utilities to the price of fuel, which is readily available. By linking the utility prices to the cost of fuel and energy, we can always have a reliable cost estimate that fluctuates with market conditions. The prices in Table 6.3 reflect the fuel and energy prices of 1988 and should be updated to today's prices when evaluating current processes.

Utility	**Factor**	**Price** (1988)
Fuel (oil or gas)	1.0	\$4.00/10^6 BTU
Steam		
600 psig at 750°F	1.30	\$5.20/1000lb
Saturated Steam		
600 psig	1.13	\$4.52/1000lb
250 psig	0.93	\$3.72/1000lb
150 psig	0.85	\$3.40/1000lb
50 psig	0.70	\$2.80/1000lb
15 psig	0.57	\$2.28/1000lb
Electricity	1.0	\$0.04/kwhr
Cooling Water	0.75	\$0.03/1000 gal

Table 6.3 Utilities Costs (1988) [3].

Overhead and Labor Costs

The overhead costs are a collection of costs associated with the process which are not captured by the other cost categories. Generally, the overhead costs include the costs associated with performing the project in a given facility. These costs pay for maintenance of the facility, maintaining the necessary services to the facility and all the auxiliary services such as emergency response services. The overhead costs can be represented as a fixed cost per day value for a given facility and are a function of the time a process remains in a facility. For the early stages of the design process, we can use an average overhead cost value for all facilities and then revise this number once we allocate the process to an existing facility.

The labor costs for a process are estimated in two ways. First, a facility can have a fixed amount of labor associated with running any process within that facility so that the process is charged for the total number of personnel associated with that facility. For example, if a facility has two production engineers and five operators then the process would be charged for all seven personnel, even if the process is a relatively simple one requiring only four operators. The second manner in which labor costs can be estimated is in terms of personnel required for each operating step. For example, a filtration may be a labor intensive step requiring a fraction of an engineer and two operators, while a distillation may only require one engineer and one operator. In this case the personnel are valued as utilities that can be added or removed as needed for each processing step. The facility personnel are managed similar to utilities in terms of availability and we can set constraints on the objective function limiting the maximum number of personnel available for the process. While assigning a labor cost to each step may provide a manner for analyzing the costs per operating step, generally, we have fixed number of personnel assigned to a given facility and the process is charged a fixed amount for the entire team of personnel. This is the approach we have chosen to use. Equation 6.6 provides the overhead and labor cost function.

$$\begin{aligned} C_{Overhead\,\&\,Labor} = {} & BatchCycleTime(FacilityCost\,/\,day) \\ & + NumberofOperators(OperatorCost) \end{aligned} \tag{6.6}$$

Process Waste Costs

Given today's environmental protection standards, waste disposal costs are becoming an important component of the cost of a batch process. While, relative to continuous processes, the total amount of waste produced by batch processes is far lower than by continuous processes, the amount of waste per kilogram of product produced is substantially higher for batch processes. For batch processes it has been reported that the processes can generate anywhere for 10 to 10,000 kilograms of waste per kilogram of product [12]. In addition to the direct cost of disposing of the waste produced by the batch process, there are indirect costs that can be attributed to failure to deal with waste production in an appropriate and evermore regulated manner. Failure to handle wastes properly can result in fines and in the worst case, plant shut down which can cost the company millions of dollars. Also, improper disposal of wastes have come back to haunt companies and the government in the form of massive waste site clean-up projects.

In recent years, the United States Environmental Protection Agency (EPA) has promoted a hierarchy of options with respect to the handling of process waste: i) process modification to avoid the generation of the waste, ii) recover and recycle, and iii) destruction and disposal. The first option is to develop processes that don't generate much waste. This requires chemists and engineers to work together to develop processes that convert less raw material to undesired by-product and are more efficient by using less auxiliary materials, such as solvents. Obviously, the best solution from a waste stand point is not to produce the waste in the first place. Unfortunately, this is usually not very easy to do. The next option in the EPA hierarchy is to recycle. Much of the waste produced by batch processes is in the form of solvents. Due to the high value of the products produced in the pharmaceutical and specialty chemical industry, the recovery and recycle of the solvents has not been a priority. Only recently with the passage of strict environmental regulations and the increasing cost of waste treatment and disposal has there been an increase in the area of recovery and recycle of reusable materials. Finally, there is the destruction and disposal of the waste. This is the least desirable option from an environmental standpoint, since usually the destruction of the waste simply reduces the volume of the waste, but we are still left with waste that must be disposed of in some type of storage facility such as a landfill. The problem with this solution is that this provides the potential for future impact, for instance leaking storage containers, and also future storage problems once the storage facilities become filled.

As part of determining the cost of waste treatment for a process, we need to establish prices for different types of treatment. The different prices for waste treatment can be constructed around different categories of waste and their corresponding treatment options. Of course, not all treatment options are applicable for each type of waste. For example, we cannot send biotoxin containing wastes into a biopond without destroying the activity of the pond. At the early stages of process development, we can use simplified categories for waste treatment in order to develop a preliminary cost estimate based on waste load and generic type of treatment. Table 6.4 contains some cost data for generic forms of waste treatment obtained from Seymour [13].

Generic Treatment Costs	
Sludge with Inorganics	\$0.20/lb
Sludge with Organics	\$0.32/lb
Sewer discharge	\$0.20/1000 lb
Biological Oxygen Demand (BOD)	\$0.25 – 0.35/lb

Table 6.4 Generic Treatment Costs [13].

We must use caution in relying on the values in Table 6.4 for estimating the waste treatment costs. The actual cost of waste treatment is a strong function of the compounds in the waste, the amount of waste and the applicable treatment technology. Table 6.5 contains a sample of some remediation technologies and estimates of their associated cost, obtained from The Federal Remediation Technologies Roundtable web site [14]. Determining the applicability of these technologies can only be determined when the waste material composition is known. Additional information, including the required data for determining the applicability of the technologies listed in Table 6.5 and many others can be found at http://www.frtr.gov/matrix/section1/toc.html.

Treatment Technologies	Cost
Soil, Sediment and Sludge	
• Slurry Phase Biological Treatment	\$130 - \$ 210 /m^3
• Chemical Extraction	$110 - $ 440 /metric ton
• Chemical Reduction/Oxidation	\$ 190 - \$ 660 /m^3
• Dehalogenation	$ 220 - $ 550 /metric ton
• Solidification/Stabilization	$ 110 /metric ton
• Incineration	$ 220 - $ 6600 /metric ton
Ground Water, Surface Water and Leachate Treatment	
• Bioreactors	Installation of unit with Surface Area of 9,300 – 13,900 m^2 ≅ \$ 80 - \$85K
• Granulated Activated Carbon/Liquid Phase Carbon Adsorption	0.4*10^6 l/day ≅ \$0.32 - \$1.70/1000 l
• Ion Exchange	$ 0.08 - $ 0.21 /1000 l
Air Emissions/Off-Gas Treatment	
• Biofiltration	$ 5 - $ 10 /kg contaminant
• High Energy Corona	$ 10 /lb contaminant
• Oxidation	$ 8 - $ 40 /day + $25000/200scfm unit
• Vapor Phase Carbon Adsorption	$ 2 - $ 3 / lb carbon + $1000/100scfm unit

Table 6.5 Sample Remeditaion Treatment Technologies and Costs [14]

As with most of the cost components in process synthesis waste treatment costs consist of trade-offs. Reducing waste through process development increases the design costs, while decreasing the recycle and treatment costs. Recycling can increase the processing costs through increased equipment and utility usage, while decreasing the waste disposal costs through the reduction the volume of waste that needs to be disposed and decreases the chemicals costs through the reuse of the recovered materials. Allowing the waste to leave the process decreases the development costs and recycling equipment costs but increases the chemical usage cost as well as increasing the disposal costs.

The process waste costs can be represented in the objective function by creating general categories for different types of wastes based on the phase and the composition of the waste streams produced. We can then assign an average cost for the treatment or disposal of the waste by outside sources. Equation 6.7 represents the cost of waste treatment. The process waste cost is then a function of the waste category and the amount of waste within that category. Processes that produce more waste will incur a higher cost. The cost and savings obtained through recycling will be reflected in the equipment and materials costs. The equipment costs will be higher but the materials costs will be lower. An improved process based on design modifications will be reflected through lower waste costs. We cannot a priori say anything about how design modifications will be reflected in materials costs or equipment costs because, depending upon the modification, each of these costs may go up or down. One problem with this method of accounting for waste costs is that the cost for the development of the process by the process engineers and chemists cannot be reflected in the objective function so that the time spent developing more efficient processes is not represented in the cost of the process.

$$C_{ProcessWaste} = \sum_i F_{Waste\,i}(P_{Waste\,i}) \qquad (6.7)$$

$F_{Waste\,i}$ = Amount of waste i

$P_{Waste\,i}$ = Price of treatment for waste i

6.3.1.2 The Objective Function

Now that we have discussed the individual components of the objective function we can formulate the complete objective function. The objective of process synthesis is to produce the desired product at the required specifications at the minimum cost. This statement of the objective outlines how the objective function must be formulated. The objective function that is to guide the process synthesis is the minimization of the cost of production subject to the constraint that we produce the product at the required specifications. The functional form of the objective function is given in Equation 6.8.

$$\begin{aligned}&\min\ Cost = \\ &\min\ (C_{Raw\,Materials} + C_{Equipment} + C_{Utilities} + C_{Overhead\,and\,Labor} + C_{ProcessWaste})\end{aligned} \qquad (6.8)$$

Each of the cost components are functions of the *manipulated variables.* The manipulated variables are variables representing operating conditions such as temperature, pressure and reflux ratio, as well as equipment size. They are also a function of the *decision variables.* The decision variables represent degrees of freedom and, in the case of process synthesis, also represent which tasks are included in the process. Many of the variables within the cost functions are dependent upon the value of constraints. These variables are called the *dependent variables* and are solved using the constraints. For example, if we know that the product degrades at a given temperature we can set a constraint that limits the value of the temperature of all operations to be below the decomposition temperature. The individual cost components in Equation 6.8 are functions of variables that are determined through the modeling of the individual tasks in the process. For example, the raw material cost is a function of the conversion, selectivity and the percentage of the product that is recovered by each separation step. The percentage recovered at each separation step is determined through the models that lie behind the operations.

6.3.2 Constraints

Whenever we perform an optimization it is always subject to constraints. Constraints are a very important part of the optimization problem since they set the limits of acceptability for the resulting processes. Any process that violates the constraints is unacceptable. The formulation of the constraints forces the optimization to propose solutions that meet the constraints. In Chapter 3 we have shown the general mathematical formulation of the constraints. In process synthesis we have two classes of constraints, *global* constraints which must be met by the process and *local* constraints which must be met by a specific operation or set of operations. Examples of global constraints are the purity requirement of the final product or the specification of a maximum operating temperature to protect against product degradation. While an example of a local constraint is the maximum pressure specification of a vessel.

6.3.3 Models

In order to optimize the process we must be able to model the system. For each operation we must construct models that provide us with the material balances, energy balances, phase equilibrium and composition information for the system. These models are functions of the feed composition, operating conditions and the decision variables. Through the manipulation of the variables, the state of the system is altered in order to determine the optimal values of the variables and minimize the cost. The general mathematical formulation of the models was discussed in Chapter 3. Examples of the detailed models are given in Chapter 7. Illustration 6.5 provides an example of the model requirements for a distillation operation.

> *Illustration 6.5*
>
> For the case of modeling a distillation operation the models require the following components, phase equilibrium equations, total material balance equations, component material balance equations and enthalpy balance equations.

6.4 MEA-NMP and Logic

As we have shown, the MEA-NMP methodology simultaneously constructs a superstructure of process alternatives along with the base-case design. This process of constructing the superstructure and the base-case design is based on logic. The selection of the task categories is based on rules that have been formulated with respect to characteristics of the current state of the process materials and the desired goal state of the product. The white knight task category identification is also logic based, with the logical implications having the form: if not precondition, PC1, then white knight task category, WK1. Raman and Grossmann [15, 16] have shown how rules can be converted into propositional logic, which in turn can be converted into algebraic statements (See Illustration 3.4 in Chapter 3).

In the logic-based MINLP formulation proposed by Raman and Grossmann [16], the superstructure is modeled through logical implications and the inclusion of individual units is modeled through a series of disjunctions. The generation of alternatives through the use of the MEA-NMP methodology, allows us to generate the logical statements proposed by Raman and Grossmann. Illustration 6.6 shows how the MEA-NMP methodology formulates logical statements using the superstructure generated for the Carbinol stage in Figure 6.4.

Illustration 6.6

The construction of the superstructure is, in essence, the construction of the logical implications, which will govern the search for the optimal process. Through an examination of the superstructure developed for the Carbinol stage, we can quickly identify the logical implications embedded within it. The logical implications are necessary for the logic-based MINLP formulation proposed by Raman and Grossmann [16]. Figure 6.6 contains a simplified version of the Carbinol superstructure shown earlier in Figure 6.4.

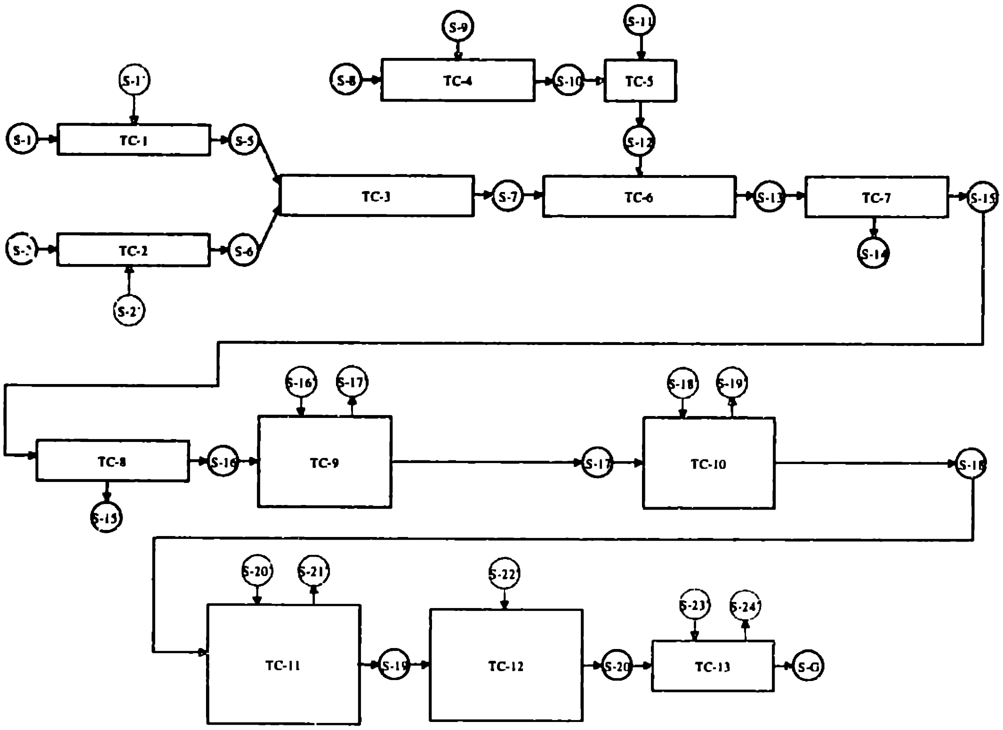


Figure 6.6 Simplified Carbinol Stage Superstructure

From the superstructure in Figure 6.6, we can identify the logical implications which describe the connections among the operations based on their task categories. Table 6.6 contains the list of logical implications.

Logical Implications	
TC-1⇒ TC-3	TC-8⇒ TC-9
TC-2⇒ TC-3	TC-8⇒ TC-7
TC-3⇒ TC-1∧TC-2	TC-9⇒ TC-10 ∨ (¬TC-10 ∧ TC-11)
TC-3⇒ TC-6	TC-9⇒ TC-8
TC-4⇒ TC-5	TC-10⇒ TC-9
TC-5⇒ TC-6	TC-10⇒ TC-11
TC-5⇒ TC-4	TC-11⇒ TC-10 ∨ (¬TC-10∧TC-9)
TC-6⇒ TC-7	TC-11⇒ TC-12
TC-6⇒ TC-3∧TC-5	TC-12⇒ TC-13
TC-7⇒ TC-8	TC-12⇒ TC-11
TC-7⇒ TC-6	TC-13⇒ TC-12

Table 6.6 Logical Implications for the Carbinol Stage

For the formulation of the optimization problem, these logical implications are converted into algebraic statements which represent the constraints to be imposed on the superstructure. For example, the implication TC-9⇒ TC-10 ∨ (¬TC-10 ∧ TC-11) is transformed into $y_9 \leq 1$ and $y_9 - y_{10} - y_{11} \leq 0$, where the y's are binary variables.

Once we have constructed the logical implications, we can go one step further and generate the logical statements for each task category and the individual operations contained within each task category. The logical statements for the task categories are the binary statements which state whether or not the task category is included in the superstructure. For example, we can express the logical statement of the task categories as a logical disjunction of the form TC-n ∨ ¬(TC-n). By not including a task category we eliminate an entire set of operations simultaneously. The logical statements for the *SolubilityReduction* task category are given in Table 6.7. We should note that the statements shown in the table are incomplete, containing only material balance equations, and should include additional equations to model the operation as well as the corresponding cost.

In addition to the disjunctions for the task categories, we also must construct the logical statements for the operations contained within each task category. The logical statements for the operations are the binary constraints that state whether an operation is included in the process or it is not. Figure 6.7 represents the superstructure of the operations contained within the *SolubilityReduction* task category. We see that the task category is connected to the rest of the superstructure by streams S16 and S17. These streams can be viewed as continuous variables for use in the optimization problem. S16' and S17' are streams that exist only when required, i.e. charging of a mass separating agent. Using the figure, we see that the information from the figure is captured within the disjunction for the *SolubilityReduciton* task category in Table 6.7. The top portion of the table contains the logical statements for representation of the inclusion of the *SolubilityReduction* in the process superstructure. The bottom of the table represents not including the task category, written as ¬(*SolubilityReduction*). Within the *SolubilityReduction* statement, we see the logical specification which states that we can select at most one operation from those within the task category. The logical specification can easily be converted into algebraic constraints using the method outlined in Chapter 3, yielding $y_1+y_2+y_3+y_4 \leq 1$, $y_1 \leq 1$, $y_2 \leq 1$, $y_3 \leq 1$, $y_4 \leq 1$. The logical specification is followed by the mixing and splitting balances around the mixing and splitting points for the operations within the task category (See Figure 6.7). Finally, we have the logical statements for the individual operations in the form of disjunctions containing the statements which are used depending whether the operation is included or not.

Solubility Reduction

Logical Specification:

$$(Op1 \veebar Op2 \veebar Op3 \veebar Op4) \vee \neg(Op1 \vee Op2 \vee Op3 \vee Op4)$$

Mixing and Splitting Balances:

$$S16 = S16a + S16b + S16c + S16d$$
$$S17 = S17a + S17b + S17c + S17d$$
$$S16' = S16'a + S16'b$$
$$S17' = S17'a + S17'c$$

Operation Logical Statements:

Operation1 (Op1):

$$\begin{bmatrix} \text{SolventReplacement} \\ S16a + S16'a - (S17a + S17'a) = 0 \end{bmatrix} \vee \begin{bmatrix} \neg(\text{SolventReplacement}) \\ S16a = S17a = 0 \\ S16'a = S17'a = 0 \end{bmatrix}$$

Operation2 (Op2):

$$\begin{bmatrix} \text{Charge MSA} \\ S16b + S16'b - S17b = 0 \end{bmatrix} \vee \begin{bmatrix} \neg(\text{Charge MSA}) \\ S16b = S17b = 0 \\ S16'b = 0 \end{bmatrix}$$

Operation3 (Op3):

$$\begin{bmatrix} \text{Concentrate} \\ S16c - (S17c + S17'c) = 0 \end{bmatrix} \vee \begin{bmatrix} \neg(\text{Concentrate}) \\ S16c = S17c = 0 \\ S17'c = 0 \end{bmatrix}$$

Operation4 (Op4):

$$\begin{bmatrix} \text{Cool} \\ S16d - S17d = 0 \end{bmatrix} \vee \begin{bmatrix} \neg(\text{Cool}) \\ S16d = S17d = 0 \end{bmatrix}$$

¬(Solubility Reduction)

Balances:

$$S16 - S17 = 0$$
$$S16' = 0$$
$$S17' = 0$$

Table 6.7 Logical Statements for the SolubilityReduction Task Category

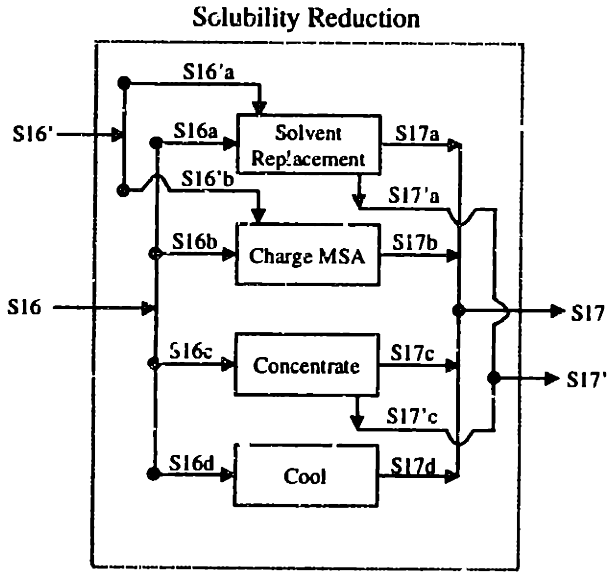


Figure 6.7 Superstructure for the Solubility Reduction Task Category

Thus, we see from the two tables above that the MEA-NMP methodology can be used to derive the required elements for the logic-based MINLP formulation of the optimization problem.

6.5 Superstructure Reduction

Earlier in this chapter we have shown how the superstructure can be formulated using the MEA-NMP methodology and how superstructures can significantly reduce the search space of alternatives for the optimization problem. Even though the systematic construction significantly reduces the search space of alternatives, the problem can remain quite large for industrially relevant problems. In this section, we will discuss the three methods that have been proposed to screen the superstructures and further reduce the search space of alternatives for the process synthesis optimization.

6.5.1 Heuristic Superstructure Reduction

Rudd et al [17], Douglas [3] and others have proposed using heuristics for process synthesis. There is no reason that these heuristics cannot be employed to perform superstructure reduction. Until recently, heuristics have been the only method for the reduction of superstructures. Heuristics provide the advantage of quickly removing alternatives from the superstructure. We use our engineering expertise and the rules of thumb developed by others to quickly screen the alternatives within our superstructure and eliminate those alternatives that violate the heuristics. The trouble with using heuristics is that there are always exceptions to the rules and we may unknowingly eliminate the optimal solution from consideration. Another problem is knowing what to do when there are contradicting heuristics. In order to reduce the possibility of removing the optimal solution from the superstructure, the heuristics must be applied carefully using as much of the available information as possible. In the case of contradicting heuristics we have two options, first it to ignore both of the heuristics and do not eliminate any alternatives from the superstructure and second, is to determine which heuristic has priority and eliminate alternatives based on the heuristic that has higher priority while ignoring the heuristics which has lower priority. Illustration 6.7 details why it is important to use heuristics with care.

> *Illustration 6.7*
>
> A common separation system heuristic is: *Avoid the use of additional compounds*. If we were to blindly apply this heuristic we would eliminate many important alternatives from the process such as extraction. Careful usage of this heuristic can provide benefits. For example, we can modify the heuristic slightly and develop a new heuristic: *Avoid the use of additional materials that can form azeotropes with any process components.* While this does not guarantee that we do not eliminate the optimal design, we generally like to avoid the generation of azeotropes since they usually involve a greater number of processing steps to separate them, which tends to either increase the processing costs or waste disposal costs. Of course there are some cases where an azeotrope can be a useful element in the separation of difficult to separate components and we must not allow the heuristics we develop completely eliminate these alternatives.

6.5.2 Mathematical Screening Methods

The main criticism of the heuristic method is that the screening is not rigorous and depends too much on the experience of the developer. Recently there have been a few attempts to add rigor to the screening of process superstructures. As far as we have been able to determine there have only been three methods proposed, the aggregation and decomposition method proposed by Daichendt and Grossmann [18], the graph-theoretic method by Friedler et al [1] and the screening model method by Allgor [19]. Each of these methods will be briefly described without any attempt to evaluate one against the other.

6.5.2.1 Aggregation and Decomposition Method

The aggregation and decomposition method proposed by Daichendt and Grossmann [18] performs a preliminary screening of the superstructure through the use of aggregated models. "These are simplified representations of the MINLP models, may be lower dimensional, and are based on linear underestimations of the cost and convex over estimations of the feasible region."[18]. The method attempts to bound the solution to the optimization problem between a base-case design determined from the original superstructure as the upper bound on the cost and an underestimated lower bound determined by taking integer cuts of the aggregated model.

The aggregation of the models for the optimization problem are generated through the convexification and combination of the original models, the linearization and combination of the constraints and the linearization of the fixed-charge cost functions. the resulting problem is convex and can be solved to a global optimum. This allows the determination of a underestimated lower bound. As the screening takes place any tasks that are not selected during the prescreening are removed from the superstructure thus reducing the size of the superstructure. Once the screening is complete the resulting problem ie superstructure is transformed back into the original model, this is called decomposition. The resulting problem is the reduced superstructure and can be solved using standard optimization techniques described in Chapter 3. For a detailed description and examples of the aggregation and decomposition method for superstructure reduction the reader should refer back the original papers by Daichendt and Grossmann [18, 20].

6.5.2.2 Graph Theoretic Method

The graph theoretic method for superstructure reduction was proposed by Friedler et al [1]. They claim their approach generates superstructures or as they refer to them *maximal structures*. They define the maximal structure as "the largest structure that must be taken into account in process synthesis; it also contains all combinatorially feasible process structures." I claim that the approach presented by Friedler et al is actually a superstructure screening method. I make this claim based on the assumptions made by Friedler et al in order to formulate the problem. They begin the problem by assuming that they have a list of process units and process streams available. I would contend that this set information actually describes the primary superstructure and Friedler et al's method actually performs a screening of the original superstructure to determine the reduced superstructure termed a maximal structure that contains the feasible structures.

Friedler et al's method is based on the principles of graph theory. They have developed a set of axioms that were derived to define the combinatorially feasible solution structure. The axioms are defined in [21]. These axioms are applied using an algorithm for generating the maximal structure adapted from Pidgin Algol. This maximal structure generation algorithm is given in [1]. The algorithm consists of two parts a reduction part and a composition part. The reduction part removes the units and materials that must not belong to the maximal structure. The composition part selects only the units and materials that must belong to the maximal structure. We should then be left with a structure that contains only the feasible alternatives. For a detailed description and examples of the graph-theoretic method the reader should refer to Friedler et al [1].

6.5.2.3 Screening Model Method

The screening model method proposed by Allgor [19] is the only superstructure reduction method that has been developed exclusively with batch process development in mind. This method is similar to the method to that proposed by Daichendt and Grossmann which was discussed in Section 6.5.2.1. The method is based on the concept of developing screening models. "Screening models yield a rigorous lower bound on the cost of the process, providing both design targets and a valid way in which to prune or screen discrete alternatives that cannot possibly lead to an optimal solution." [19] The screening models combine changing the process structure, the determination of operating conditions and the allocation of equipment simultaneously. The screening models developed must overestimate the feasible region of alternatives and underestimate the design objective. All of the optimization variables must be considered simultaneously. The formulation of the screening models allows the constraints of the equipment allocation and scheduling to be expressed in their original form, the constraints that define the dynamic performance of the processing tasks are relaxed and the differential-algebraic equations describing task performance are replaced by algebraic equations. Also, time averaged material balances are imposed. The development of the screening models thus convexifies the problem as was done by Daichendt and Grossmann. Once the problem is defined using the screening models it is solved using MINLP solution techniques. Once we have solved the problem, we are left with a solution that provides us with a definition of the processing structure i.e. a set of operations that should be included in the superstructure, allocation of these operations to equipment and initial values for the compositions of state nodes in the resulting superstructure. From the solution of the screening model problem, we can the go back and convert the problem back into a problem with the original models and apply some of the standard optimization techniques to try to determine the optimal design. A detailed description and examples of the screening model method is given in [19] and the interested reader should refer to this source.

6.6 Comparison of Alternative Processes

Now that we have discussed the construction on the process superstructure using MEA-NMP, the potential problems with superstructure optimization and methods for reducing the size of the resulting superstructures, let us examine a few alternative processes generated using the superstructure for the Carbinol Stage given in Figure 6.4. For examples of detailed solutions to MINLP optimization problems the reader can refer to the following sources [4, 16, 22, 23].

In Chapter 4, we generated a base-case process for the Carbinol Stage of the case study. The complete base-case process is given in Figure 4.44. If we examine the superstructure for the process, given in Figure 6.4, we see that a portion of the process is fixed due to the developer inserting operations directly into the plan, for example the *dissolve* operations, or because there is only a single alternative within the task category such as the *react* operation within the *MolecularStructureTransformation* task category. Thus, we can focus our attention to the portion of the superstructure that contains alternative operations. Let us now show how selecting a different alternative impacts the process that is developed and its impact on the cost of the process.

6.6.1 Impact of a Single Alternative

To show how a single alternative can impact the process, let us replace the *concentrate and mix* alternative selected from the first *SolubilityReduction* task category shown in Figure 6.4, with an *extract* operation. The selection of the *extract* operation and the removal of the *concentrate and mix* operation amounts to setting the binary variable in the objective functions corresponding to the extract operation to one and the binary variable corresponding to the *concentrate and mix* operation to zero. Examining the preconditions of the *extract* operation, there are no violations so the *extract* operation is a feasible alternative. We then apply the *extract* operation and generate the postconditions. If we assume that the solvent selected for the extraction is efficient and extracts most of the product material into the solvent, the second *SolubilityReduction* task category becomes unnecessary. This does not force us to change the superstructure since during the optimization, we can simply set all the binary variables for operations within the *SolubilityReduction* task category to zero. The rest of the original operations selected during the base-case development are feasible, thus the new process flowsheet is given in Figure 6.8.

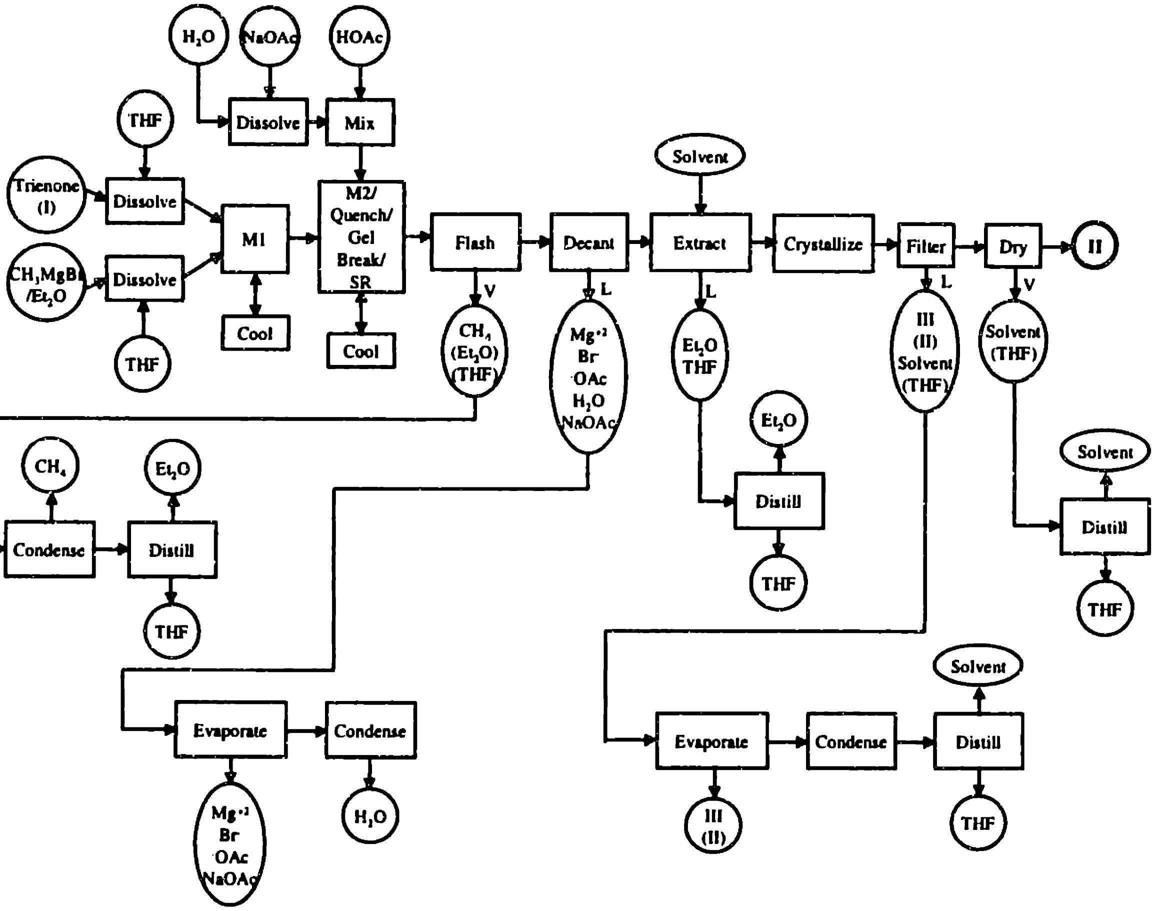


Figure 6.8 Carbinol Stage using Extract to Replace Concentrate and Mix

From Figure 6.8 we see that using an *extract* operation to replace the *concentrate and mix* operation has reduced the number of operations in the plan. We have eliminated a subsequent *concentrate and mix* operation as well as solvent recovery series of operations.

What impact does selecting the *extract* alternative have on the cost of the process? Because we have fewer operations in the plan does not simply imply that this process will be lower in cost. Before we can determine which process has the lower cost, we must examine a number of trade-offs that arise based on the selected alternated. These trade-offs include the trade-offs between the cost of the equipment required for the concentration versus that of the extraction, the time required for the alternative operations, the different amounts of utility usage, the reduction or increase in process materials used such as solvents and the impact the use of different process materials will have on waste treatment and materials recovery. Now let us examine a number of possible scenarios that demonstrate how these trade-offs impact the cost of the process.

By comparing the alternative processes given in Figure 6.8 and Figure 4.44, we see that the replacement of the *concentrate and mix* with *extract* has caused the removal of two *concentrate and mix* operations, two *condense* and a *distill* operation used in solvent recovery. On the surface it would appear that this could result in a significant reduction in equipment for the process, but lets take a closer look. We must remember that the operations based process requires us to fit the selected operations to specific pieces of equipment. Since this is the case, the number of operations does not have a direct correspondence to the amount of equipment required for the process. As we have mentioned earlier, the scheduling problem for the batch process can be quite complex and must be included as part of the optimization, especially for multibatch processes. For the remainder of our discussion here, for the sake of simplicity, let us assume that we are dealing with a single batch process. An example of fitting multiple operations into a single piece of equipment is performing the *concentrate and mix* operations that are contained in the original process in the same piece of equipment, thus it is the amount of time the equipment is in use which changes, not the amount of equipment. Then, if the amount of time it takes to perform the two *concentrate and mix* operations is shorter than the time required to perform the *extraction*, the equipment cost for the *concentrate and mix* operations could be lower than that for the *extraction*, assuming that the unit price for the two pieces of equipment are the same. With respect to the objective function, varying the time of equipment usage changes the cost in the equipment component.

Even though the utilities cost has been shown to be a relatively small portion of the cost of a process, selecting alternative operations does have an impact on the amount of utilities required for the process and as we have mentioned earlier, it is usually not the cost of utilities that limits a process but instead the availability of utilities. When choosing a specific facility, we can add constraints on the amount of various utilities that are available or we can set an extremely high price for utility usage over a fixed amount thus, forcing processes which use large amounts of utilities to reflect that in the cost. For now lets examine how selecting an alternative operation impacts the utility cost. It should be apparent that the *concentrate* operation requires either heating i.e. evaporating the solvent or vacuum which is reflected through compressor charges, while the extract operation generally requires only mixing. From a utilities point of view, the extract operation will contribute minimally to the utility portion of the cost function, while the utility cost component of the concentrate operation will be significantly higher.

The selection of the *extract* operation can also have an impact on the raw material and process material costs associated with the process. The efficiency of the *extract* operation i.e. the percentage of the product the extracting solvent can extract, will effect the yield. If the extraction is highly efficient, the product recovery may be similar to that of the concentrate and mix operation. If the efficiency is low, then we must either perform additional operations to recover additional product material from the raffinate or we must increase the amount of raw material used to make the desired amount of product. Both of these options have the potential to increase the cost of the process, either by adding additional equipment or increasing the processing time or by increasing the amount of raw material used as well as increasing the waste treatment costs by sending more product to waste treatment. The concentrate and mix route avoids the loss of product to waste treatment, unless the product is volatile.

As we have mentioned while discussing the impact of the alternatives on the raw material cost, the extract operation has the potential to increase the waste treatment cost by sending some product to waste treatment. On the other hand, the selection of the extract operation also has the potential to lower the solvent recovery and waste treatment costs. If we look at the original process in Figure 4.44, we see that the concentrate operations create two vapor phase waste streams which contain solvent that could be recovered and recycled. The extract operation only creates a single liquid phase waste stream from which solvent can be recovered. Thus, the resulting solvent recovery system resulting from the implementation of the extract operation contains two fewer *condense* operations and two fewer *distill* operations than does the plan with the concentrate and mix operations. The resulting impact on the cost of the process is that the plan containing the *extract* operation will require less equipment as well as lower utility usage for the solvent recovery, thus having a lower solvent recovery cost. Along with determining the cost for solvent recovery we must determine the cost involved by not recovering solvents and simply disposing of the wastes produced. This is generally unfavorable but in some cases could be more economical. For example, if the waste stream that is created is a water based mixture that can be sent directly to the sewer system, the cost of recovering the water may be much higher than simply disposing of the water. The cost of not recovering material is the initial cost of purchasing the material as well as the cost of the particular treatment and disposal methods for that material. So we see that simply disposing of large amounts of process materials can contribute significantly to the cost of the process.

In addition to the recovery and disposal issues, the plan containing the *extract* operation will generate fewer fugitive emissions since it does not vaporize any solvent. While we have not chosen to add this as a consideration in our costing of the process, this is an important issue when evaluating if the facility is within environmental regulatory compliance.

Above we have discussed some of the main implications of alternatives on the cost of the process. As we have shown, there are a number of trade-offs that exist with the different alternatives and we cannot determine the optimal process until we perform the detailed optimization including all of the cost information with respect to raw materials, process materials, equipment costs, task allocation and scheduling, utilities costs and solvent recovery and waste treatment costs.

6.6.2 Impact of Alternatives on the Superstructure

Since the MEA-NMP methodology derives the superstructure through the development of a base-case design, there is the potential for different decisions made during the development of the base-case to impact the formulation of the superstructure. Thus, by developing an alternative base-case process, it is possible to generate alternative superstructures. This can occur when selecting operations to remove precondition violations or when selecting an operation that generated a different resulting state, which causes precondition violations in later operations in the plan. Since each operation has its own set of preconditions, there is the potential for different operations within a given task category to require different white knights to remove violations of the preconditions. The addition of white knights during the base-case construction changes the superstructure that is developed. Thus, the optimization we perform over a given superstructure does not necessarily provide the optimal process. We must compare the optimal process developed from the current superstructure to optimal processes developed through the optimization of alternative superstructures. We can generate an example by looking at Figure 6.4. For the base-case process, we have selected the *crystallize* operation from the *LiquidtoSolidPhaseChange* task category. Instead of selecting the *crystallize* operation, we could have selected the *precipitate* operation. In this case, the process plan would be unaffected up to and including the *precipitate* operation. After the application of the precipitate operation, the resulting solid no longer has the proper purity, resulting in an additional concentration difference. Thus, after the application of the *filter* operation and the *dry* operation, the concentration would no longer meet the product specifications. The MEA-NMP methodology would then identify the *SolidSolidSep* task category to be added after the *LiquidfromSolid* task category, expanding the superstructure as shown in Figure 6.9.

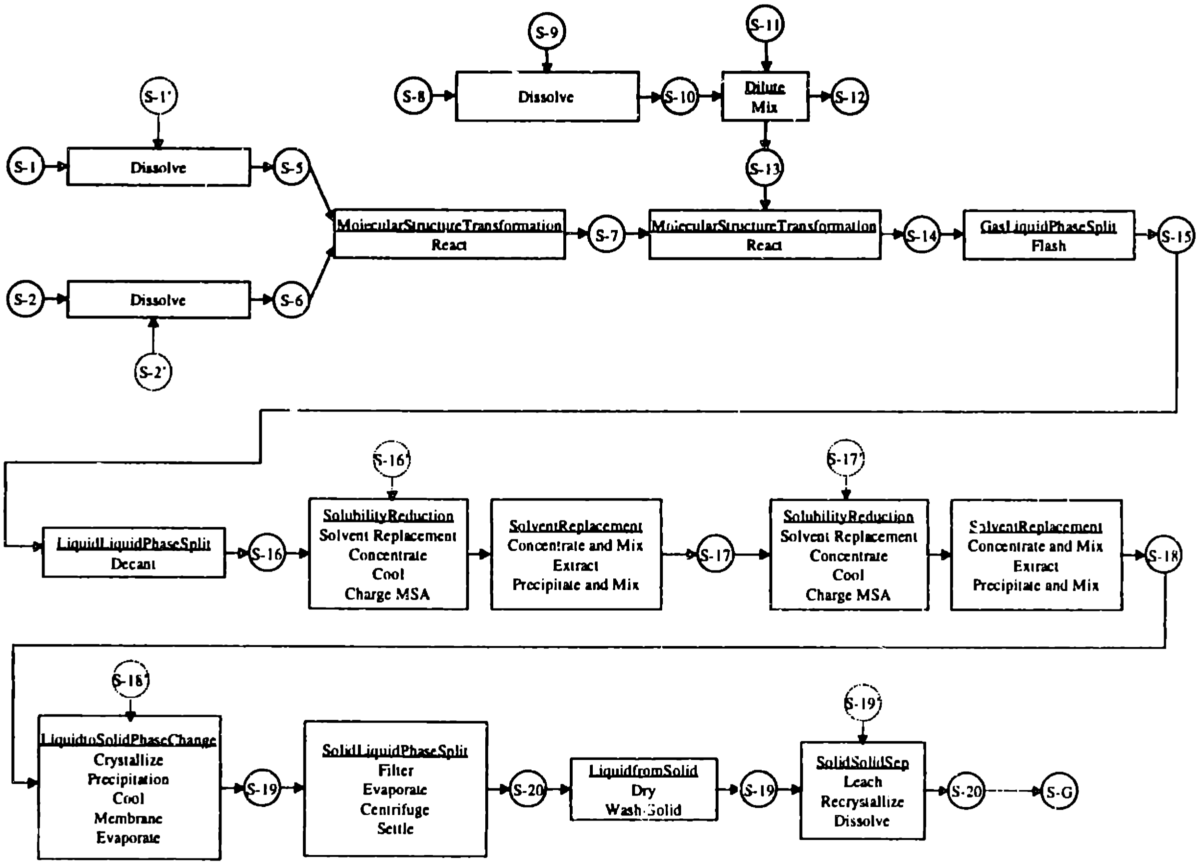


Figure 6.9 Superstructure resulting from the Replacement of Crystallize with Precipitate

From Figure 6.9, we see that the resulting superstructure is different from that given in Figure 6.4, requiring that, in order to determine the optimal design, we must optimize the superstructure in Figure 6.9 and then compare the optimal process found using this superstructure with the optimal process found using the superstructure in Figure 6.4.

This chapter has given an overview of the construction of the optimization problem based on the reduced superstructure proposed using the MEA-NMP method. We have also provided a description of the components and formulation of the objective function. Examples of superstructure reduction methods have also been given. We have shown how alternatives impact the cost of the process as well as effect the form of the superstructure. Biegler et al [24] and Floudas [2] provide excellent examples of detailed optimization problem formulation once the superstructure has been constructed. The interested reader should refer to one of these sources for a detailed description and examples of the mathematical construction of the optimization problem based on superstructures.

6.7 References:

1. Friedler, F., *et al.*, *Graph-Theoretic Approach to Process Synthesis: Polynomial Algorithm for Maximal Structure Generation.* Computers and Chemical Engineering, 1993. **17**(9): p. 929-942.
2. Floudas, C.A., *Nonlinear and Mixed-Integer Optimization.* Topics in Chemical Engineering, ed. K.E. Gubbins. 1995, New York: Oxford University Press. 462.
3. Douglas, J.M., *Conceptual Design of Chemical Processes.* McGraw-Hill Chemical Engineering Series. 1988, New York: McGraw-Hill Publishing Company. 601.
4. Grossmann, I.E., *Mixed-Integer Optimization Techniques for Algorithmic Process Synthesis*, in *Process Synthesis*, J.L. Anderson, Editor. 1996, Academic Press: Boston. p. 172-246.
5. Allgor, R.J. and P.I. Barton, *Mixed-Integer Dynamic Optimization I: Problem Formulation.* Computers & Chemical Engineering, 1998.
6. Rajagopal, S., K.M. Ng, and J.M. Douglas, *Design and Economic Trade-Offs of Extractive Crystallization Processes.* AIChE Journal, 1991. **37**(3): p. 437-447.
7. Rubin, R.H., *et al.*, *15.136J, Principles and Practice of Drug Development*, . 1997: Massachusetts Institute of Technology.
8. Kocis, G.R. and I.E. Grossmann, *A Modeling and Decomposition Strategy for the MINLP Optimization of Process Flowsheets.* Computers and Cnemical Engineering, 1989. **13**: p. 797.
9. Daichendt, M.M. and I.E. Grossmann, *Integration of hierarchical decomposition and mathematical programming for synthesis of process flowsheets.* Computers and Chemical Engineering, 1997. **22**(1-2): p. 147-175.
10. Cooney, C.L., *Biochemical Process Synthesis and Analysis*, in *Downstream Processing Short Course.* 1998: MIT. p. 8.1-8.65.
11. Guthrie, K.M., *Capital Cost Estimating.* Chemical Engineering, 1969. **76**(6): p. 114.
12. Linninger, A.A., *Waste Production by Batch Processes*, . 1996.
13. Seymour, C.B., *Conceptual Design of Dedicated Batch Versus Continuous Processing in Multireaction Step Processes*, in *Deparment of Chemical Engineering*. 1995, University of Massachusetts Amherst: Amherst, MA. p. 394.
14. Roundtable, F.R.T., *Remediation Technologies Screening Matrix and Reference Guide*, . 1998, United States Army Environmental Center: Aberdeen Proving Ground, MD.

15. Raman, R. and I.E. Grossmann, *Symbolic Integration of Logic in Mixed-Integer Linear Programming Techniques for Process Synthesis.* Computers & Chemical Engineering, 1993. **17**(9): p. 909-927.

16. Raman, R. and I.E. Grossmann, *Modelling and Computational Techniques for Logic Based Integer Programming.* Computers & Chemical Engineering, 1994. **18**(7): p. 563-578.

17. Rudd, D.F., G.J. Powers, and J.J. Siirola, *Process Synthesis.* 1973, Englewood Cliffs, New Jersey: Prentice-Hall, Inc. 320.

18. Daichendt, M.M. and I.E. Grossmann, *Preliminary Screening for the MINLP Synthesis of Process Systems. I. Aggregation and Decomposition Techniques.* Computers and Chemical Engineering, 1994. **18**: p. 663.

19. Allgor, R.J., *Modeling and Computational Issues in the Development of Batch Processes*, in *Department of Chemical Engineering*. 1997, Massachusetts Institute of Technology: Cambridge. p. 401.

20. Daichendt, M.M. and I.E. Grossmann, *Preliminary Screening for the MINLP Synthesis of Process Systems. II. Heat Exchanger Networks.* Computers and Chemical Engineering, 1994. **18**: p. 679.

21. Friedler, F., *et al.*, *Graph-Theoretic Approach to Process Synthesis: Axioms and Theorems.* Chemical Engineering Science, 1992. **47**(8): p. 1973-1988.

22. Kocis, G.R. and I.E. Grossmann, *Global Optimization of Nonconvex Mixed-Integer Nonlinear Programming (MINLP) Problems in Process Synthesis.* Ind. Eng. Chem. Res., 1988. **27**: p. 1407.

23. Turkay, M. and I.E. Grossmann, *Logic-Based MINLP Algorithms for the Optimal Synthesis of Process Networks.* Computers and Chemical Engineering, 1996. **20**(8): p. 959-978.

24. Biegler, L.T., I.E. Grossman, and A.W. Westerberg, *Systematic Methods of Chemical Process Design.* Prentice Hall International Series in the Physical and Chemical Engineering Sciences. 1997, Upper Saddle River, NJ: Prentice Hall PTR. 796.

Chapter 7

BatchDesign-Kit: The Computer-Aided Environment to Support the Synthesis and Evaluation of Batch Processing Schemes

While the formulation of a systematic methodology for the development of batch processes is an important advancement in the area of process development, the true benefit of such a methodology can be realized through its implementation as a computer-aided process development tool. A tool of this type has the potential to enhance the ability and efficiency of our human experts by acting as an assistant. The system can keep track of decisions made by the expert, possibly supply alternatives that the expert may not have considered and perform tedious tasks and calculations allowing the expert to focus their attention on the interesting aspects of the development process. The *BatchDesign-Kit* (BDK) process development tool was developed specifically for this purpose, to act as an assistant, as well as, a knowledge and information repository. The goal of the BDK project was to develop an industrial grade software tool for the synthesis and evaluation of batch processing schemes, beginning with the synthesis of batch processing recipes that process chemists carry out and moving through process development to the pilot plant and to the process design stage with plans that could be taken directly into manufacturing.

It should be apparent from our discussion of the MEA-NMP methodology in previous chapters that the MEA-NMP methodology requires significant information support in order to be effective. We have implemented a prototype of the process synthesis tool called the *Process_Synthesizer* as a layered application within the scope of BDK, in order to make use of the many features available within BDK.

The integration of the *Process_Synthesizer* with BDK creates a computer-aided tool, which facilitates the entire development process through a single platform. In this chapter we will detail the requirements for the implementation of the *Process_Synthesizer* and the MEA-NMP methodology. We begin this chapter with a description of the necessary facilities which are required for the implementation of a process synthesis methodology as a computer-aided system, in Section 7.1. In Section 7.2, we will describe the BDK system and show that BDK provides the necessary support for the implementation of the *Process_Synthesizer* process synthesis tool. We will also provide the background leading to the development of the BDK system as well as the MEA-NMP methodology. Additionally, we will discuss the current *Process_Synthesizer* implementation. Finally, in Section 7.3 discuss the potential impact of BDK on the pharmaceutical and specialty chemical industries and new drug development.

7.1 Implementation Requirements for Process Synthesis

A computer-aided environment to support the needs of batch process development requires substantial information and computational facilities. In addition, several specific requirements must be met by such a system in order to effectively perform process synthesis within the framework of the MEA-NMP methodology. Within such a computer-aided environment, we must be able to model the behavior of materials and the material transformations induced by the various operations under different operating conditions. We must also have the ability to represent knowledge such as rules and procedural methods. To support the models and knowledge systems, we must have efficient means of accessing and storing large amounts of data such as physical properties, health and safety properties, regulatory information along with property estimation techniques for determining the values of required properties that are not readily available through databases. Finally, we must have the tools for evaluating the processes we develop in order to compare alternative processes and determine when the development process should be terminated. Table 7.1 contains a list of required components for the successful implementation of the MEA-NMP methodology.

System Requirements for Implementation

- Operation-Based Language - a language for describing batch processes in terms of transformations induced on the state of materials by the various operations, rather than in terms of the unit operations that perform the tasks, as is used in continuous processes.
- Material Models - models to describe the state and behavior of the materials contained within the system, including phase distribution and the ability to model their behavior in response to operating condition changes.
- Operation Models - models of the transformations that will be performed on the materials within the system in order to produce the desired product. For example distillation, extraction and mixing.
- Reaction Models - models which determine the products and by-products of reactions based on user defined reactions and stoichiometry and conversions. Must also have the ability to model the reaction utility requirements based on user supplied heats of reaction.
- Property Databases - databases which allow the rapid access of material, health and safety properties and allow modification by the user.
- Cost Databases - databases which contain cost information such as material, utility and equipment costs.
- Equipment Allocation Facilities - the facilities for defining physical pieces of equipment along with databases containing the equipment configurations at different production sites. The system must also support either the manual or automatic allocation of the operations based process into the physical equipment of specific plants. Additionally, it must be able to assess the costs associated with different plants.
- Process Assessment Tools - tools which help the developer evaluate the process that has been developed. These tools must be able to function at different points within the process development providing preliminary evaluation information in order to assist the developer with the decision process. Some of these tools include production cost estimation tools, scale-up and production scaling facilities and visual aids such as Gantt Charts and Process Sequence Diagrams.
- Material Assessment - a system to help the developer assess the materials used in the process based on health, safety and regulatory issues.
- Graphical User Interface - the system must be user friendly and display information in both graphical and textual formats for example

the system must be able to clearly display the flowsheet and batchsheet for the process. This is also an important feature for decision tracking during the development process.

Table 7.1 System Requirements for Process Synthesis Implementation

The BatchDesign-Kit was designed to be a fully integrated process development tool combining reaction-route development, process synthesis and plant design. BDK was designed using a modular approach such that the various components can be used independently or as a complete process development system. This modularity in the structure of BDK has allowed the MEA-NMP methodology to be added to BDK in the form of a layered application called the *Process_Synthesizer.* It also allows the *Process_Synthesizer* to take advantage of all the facilities, tools and data support structures that are contained within BDK. It is the *Process_Synthesizer* which extends BDK's applicability as an advanced flowsheet and assessment tool to a full process development tool, providing support to the chemists and the development engineers over the entire batch process development task.

Figure 7.1 depicts a schematic of the batch process development plan using BDK. Notice that aside from the identification of the target product and the earliest chemical synthesis work, all other aspects from reaction route specification through to the start of manufacturing can be carried out within the BDK environment. In the following sections we will describe the details of the BDK system including all the features that are required in order to perform process synthesis in a computer-aided environment. We will also show how the *Process_Synthesizer* is an integral part of a complete process development tool.

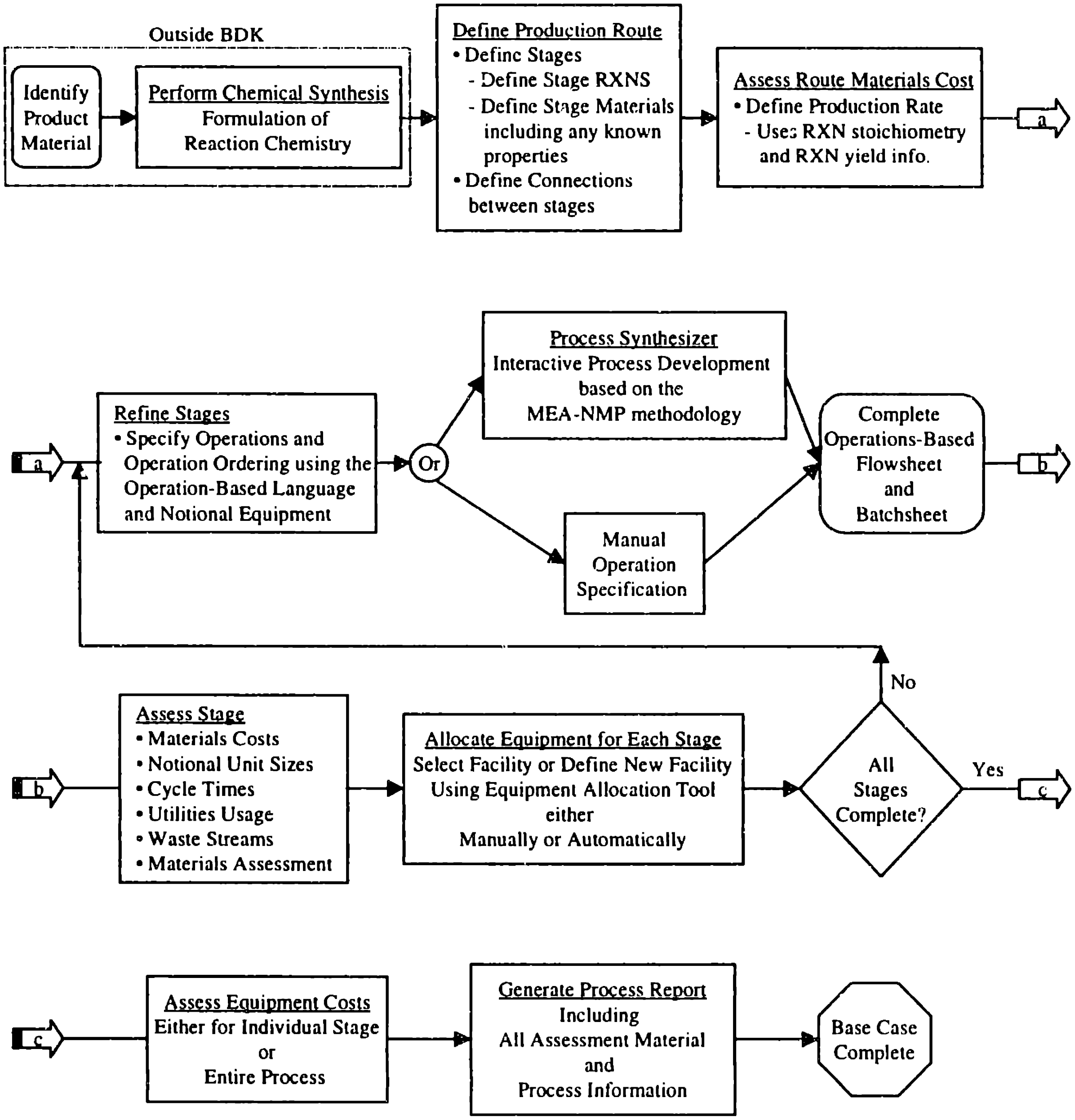


Figure 7.1 Schematic of the Batch Process Development Route Using BDK

7.2 BatchDesign-Kit

BatchDesign-Kit is a computer-aided environment intended to support the various activities associated with batch process development and the identification of the manufacturing process for the production of pharmaceutical, agricultural and specialty chemicals. It is intended to be used from the earliest stages of process conception through the later stages of process development and design. Its use at the early stages of the development of a new technology allows for the development of more cost effective and environmentally friendly processes by selecting the most appropriate reactions, solvents, reaction processes and separation processes. Through the facilities provided in BDK, BDK enhances the development of many process alternatives and allows a rapid screening of these alternatives. Different process alternatives can easily be screened in terms of cost and product quality, as well as solvent requirements and generated wastes. This increase in the speed of alternative screening allows for a more rapid development of better processes, thus increasing productivity and decreasing time to market. The completely integrated system also allows for better record keeping of the design process which is a valuable benefit for both the FDA approval process, the accumulation of corporate intellectual assets, as well as for the revamping of processes if changes are required.

The goal of BDK is to provide a fully integrated computer-aided environment for the development of batch processes through the facilitation of information sharing among all parties involved in the development process. Furthermore, it is intended to act as an assistant to the human experts, removing the burden of performing tedious tasks through automation, allowing the experts to have more time to focus on problems that only they can solve and examine a greater number of process alternatives with the goal of decreasing the time-to-market and deploying better processes into manufacturing through increases in efficiency in the development process.

Some of the main features of BDK include:

(a) A knowledge-based process synthesis tool that takes the information provided by the chemist and assists the process engineer in constructing the process flowsheet by helping to chose the operations to be performed.

(b) The operations-based language which facilitates communication between process chemists and engineers.

(c) The ability to construct many different process versions for a single project.

(d) The ability to construct many different stages within each process version.

(e) The ability to construct a flowsheet and batch sheet simultaneously, while at the same time having both the flowsheet and batch sheet interact, such that changes made to one are automatically reflected in the other.

(f) Cost analysis tools.

(g) Environmental impact and waste assessment tools.

(h) A complete list of commonly used process operations.

(i) Automatic report generation tools to cut the time spent preparing the documentation of the process and many other important tools.

These features are representative of the tasks performed by the chemists and engineers working in process development. Before we discuss the functionality of the BDK system we will first discuss the *Operations-Based Language* and the *Material Model*, which provide the basis for the BDK system. The main features included in BDK will be discussed in the subsequent sections. The description of the BDK system follows closely from the descriptions provided in Linninger et al [1, 2].

7.2.1 Background

The BatchDesign-Kit project was started in 1992 by Professor George Stephanopoulos at the Laboratory for Intelligent Systems in Process Engineering at MIT. The BDK project was an extension of a previous research effort known as *Design-Kit* [3]. Unfortunately, the *Design-Kit* project was significantly ahead of its time. The software tools (e.g. object-oriented LISP for programming and development of knowledge-bases) available during the mid to late 1980's, although sufficient to adequately capture the scope of the *Design-Kit* project, were not broadly based and used in process systems engineering. In addition, since these tools emerged from the laboratories carrying out research in artificial intelligence, they were tainted by the skepticism that accompanied the failure of "strong-AI" to deliver on its promises. Now almost ten years later, the computer facilities have advanced enough and an understanding of the capabilities and limitations of AI has emerged allowing the development of a project with the scope of BatchDesign-Kit.

Following in the footsteps of the Design-Kit project we have developed the BatchDesign-Kit. The BatchDesign-Kit has remained true to the basic principles for a computer-aided process engineering tool as specified by the developers of Design-Kit. A computer-aided process engineering tool enhances human productivity by allowing (i) high-level communication between user and computer, (ii) full and transparent integration among its various facilities, and (iii) easy maintenance, evolution and expansion of its capabilities [3]. These principles will become apparent in the following description of BatchDesign-Kit.

The BatchDesign-Kit design team consisted of Professor George Stephanopoulos, Shahin A. Ali, Dr. Andreas A. Linninger, Eleni Stephanopoulos, Dr. H. Enrique Salomone, Dr. Chonghun Han and Brian Harrison. A large number of major pharmaceutical and specialty chemical companies including Merck, SmithKline Beecham, Ciba, Bristol-Myers Squibb, Sandoz, Searle, Pfizer, along with the Office of Naval Research and the US Army sponsored the research. The goal of the BDK project was to produce a process development tool that increases the rate of process development by automating tedious tasks that are typically performed by process engineers, while at the same time extending the ideas of process design and development back to the lab and the process chemists. BDK was also designed to be a knowledge repository, allowing all participants in the design process to add to and learn from the design experience of others, contained within a centralized system. The core of the BDK project has been completed and the computer implementation is currently in use at several of the sponsor company sites.

7.2.2 Operations-Based Language

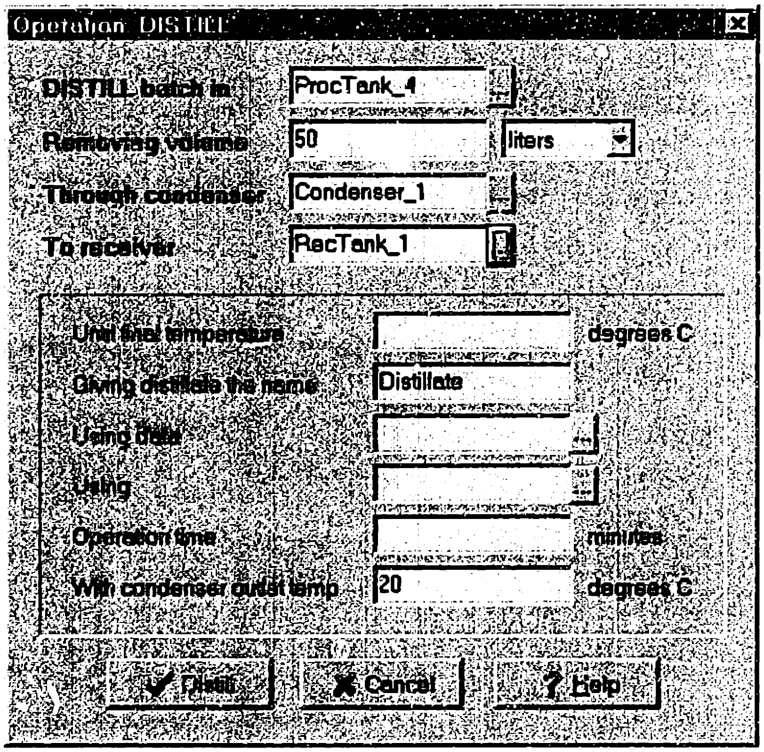
The key to the BDK system is the promotion of a paradigm shift within the realm of process development, away from the traditional units-based approach to an operations-based approach. Traditionally, process synthesis has taken place with the chemist developing a synthesis route and then passing it on to the process engineer who scales it up to a production size facility. During this process the chemist thinks in terms of operations while the process engineers typically think in terms of unit operations. The first step toward bettering communications between the chemist and the process engineer is to have them speak the same language. We have proposed the use of an operations-based language system, which composes the core of BDK. When we decompose any chemical process into its constituent elements, it consists of operations that act upon materials in order to effect a change in the state of the materials. These operations take place within pieces of equipment. For continuous processes this does not pose any difficulty since, in most instances, there is a direct one to one correspondence between a piece of equipment and the operations that take place within that piece of equipment. On the other hand, in batch processes many different sets of operations can be performed using the same piece of equipment. Thus, we see that it is only natural to decouple the operations selection process from the equipment selection process. What we really want to do is choose the operations and then fit the operations into various pieces of equipment. If we do not decouple the operations selection task from the equipment selection task, and instead begin by directly selecting equipment, we will fail to capture all of the aspects of the process and we will forgo a number of feasible designs.

An added advantage of the operations-based language is that the chemists already think in terms of operations. This is the language they use to construct their synthesis routes. The problem then becomes getting the process engineers to think in terms of operations. This is not as big a problem as it may appear to be. When the operations-based language concept is explained to the engineers, they generally agree with the operations-based synthesis approach. What we have experienced is that although the engineers agree to using the operations-based language, they quickly tend to revert to thinking in terms of units. So, how have we overcome this dilemma?

The key was developing the language in terms that sound familiar to both chemists and engineers. By providing a strict definition of the operations embedded within BDK in the form of templates, the user selects the operation that they wish to perform and provides the necessary information to complete the operation. The system then generates a *notional unit* on the flowsheet, as well as the corresponding description on the batch sheet. A list of predefined operations contained within BDK is given in Table 7.2. Figure 7.2 contains an example of an operation template for the distill operation. The notional unit merely represents the operation on the flowsheet, it has no direct correspondence with an actual piece of equipment. A batch process flowsheet is a graphical representation of the operations and the streams that connect the operations. The batch sheet is a step by step textual representation of the process which contains all the information to perform the process. A partial flowsheet and corresponding batch sheet for the Carbinol stage are shown in Figure 7.3, notice the notional units upon the flowsheet.

Material Transfer Operations	Operations on Liquids
Charge	Concentrate
Charge-from-Recycle	Distill
Clean	Distill-Continuously
Recycle	Vacuum-Distill
Transfer	Decant
Transfer-Batch-Through-Heat-Exchanger	Extract
Transfer-Intermediate	Extract-Continuously
	Stir
Heat Transfer	Reactions
Age	pH-Adjustment
Cool	Thermal-Quench
Heat	Reactive-Quench
Heat-with-Reflux	React
Operations on Gases	Column Operations
Evacuate	Column-Elution
Pressurize	Column-Loading
Pressure-Purge	Column-Regeneration
Vacuum-Purge	
Sweep	
Vent	
Operations on Solid	Synchronization
Centrifuge	Parallel-Start
Crystallize	Parallel-End
Dry	Sequential-Start
Filter	Sequential-End
Filter-Continuously	
Filter-in-Place	
Wash-Cake	

Table 7.2 BDK Operations List



DISTILL batch in: ProcTank_4

Removing volume: 50 liters

Through condenser: Condenser_1

To receiver: RecTank_1

Until final temperature: degrees C

Giving distillate the name: Distillate

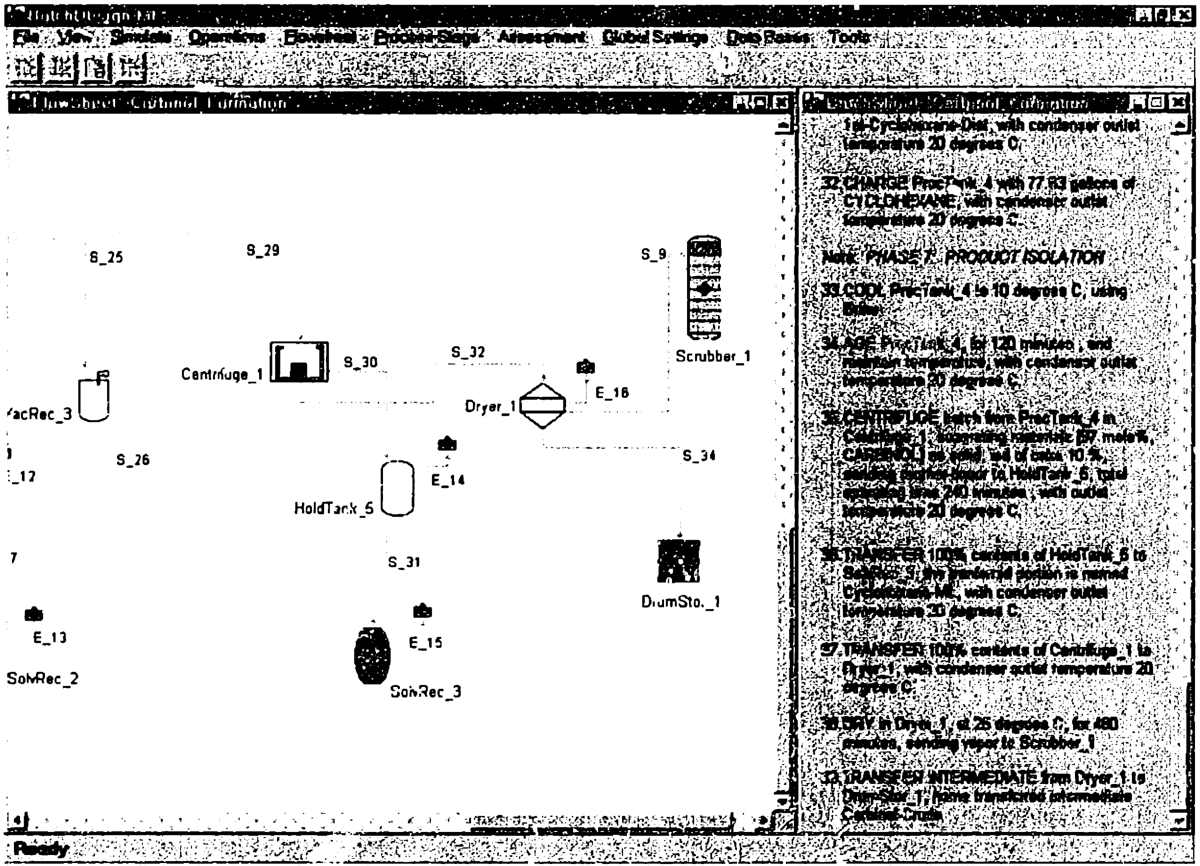
Using data:

Using:

Operation time: minutes

With condenser outlet temp: 20 degrees C

Figure 7.2 BDK Distill Operation Template



[illegible], with condenser outlet temperature 20 degrees C.

32. CHARGE ProcTank_4 with 77.93 gallons of CYCLOHEXANE, with condenser outlet temperature 20 degrees C.

Note: *PHASE 7: PRODUCT ISOLATION*

33. COOL ProcTank_4 to 10 degrees C, using Brine.

34. AGE ProcTank_4, for 120 minutes, and maintain temperature, with condenser outlet temperature 20 degrees C.

35. CENTRIFUGE batch from ProcTank_4 in Centrifuge_1, separating [illegible] 97 mole%, CARBINOL as solid, [illegible] 10 %, sending mother-liquor to HoldTank_5, total operating time 240 minutes, with outlet temperature 20 degrees C.

36. TRANSFER 100% contents of HoldTank_5 to SolvRec_3; the transferred portion is named Cyclohexane-ML, with condenser outlet temperature 20 degrees C.

37. TRANSFER 100% contents of Centrifuge_1 to Dryer_1, with condenser outlet temperature 20 degrees C.

38. DRY in Dryer_1, at 25 degrees C, for 480 minutes, sending vapor to Scrubber_1

39. TRANSFER INTERMEDIATE from Dryer_1 to DrumStor_1; name transferred intermediate Carbinol-Crude

Figure 7.3 BDK Flowsheet and Batch sheet

Once all the operations have been specified and the flowsheet is complete, the user can allocate the operations to specific pieces of equipment either manually or by having the system do it automatically. By displaying a notional piece of equipment on the flowsheet we satisfy both the requirements of the chemist who is familiar with operations, while at the same time satisfying the engineer who during the planning of the operations is accustomed to seeing units on the flowsheet. The icons for the notional units have been drawn to look like the equipment where usually the corresponding operation is carried out. This provides a quick manner to determine the types of operations which are being performed and allows the engineer to get a quick overview of the process.

Hand in hand with the operations-based language comes the increased communication potential between the chemist and the engineer. BDK provides a facility for rapid and iterative communication between the two parties. The chemist can input the reaction chemistry and chemical synthesis flowsheet. Since the chemist and the engineer are using the same system, without disturbing the chemist's synthesis development, the engineer can "look over the chemist's shoulder" to begin looking for possible synthesis steps that may cause problems later in the process development. The engineer can then provide feedback and ask questions while the chemist is developing the process recipe. This is very important since frequently, by the time the engineer receives the process from the chemist, the chemist has already begun working on another process and cannot afford the time to go back and re-examine the previous process. An example of the potential for process improvement through increased communication is given in Illustration 7.1.

> *Illustration 7.1*
>
> Traditionally, the chemist has chosen solvents based upon personal preference. Generally, it may be possible to use a number of solvents for the same processing step. If the engineer were to have access to the process while the chemist is still working on it, the engineer could say to the chemist "Would it be possible to use a different solvent because the one you have chosen gives us trouble in operation X?" At this point it would be much easier and effective for the chemist to make the change and continue with the synthesis development. While this is a simple example, it clearly shows the importance of the iterative feedback loop as early as possible in the development process making the entire process synthesis task more efficient.

The operations-based language highlights the most important aspect of the BDK system, if we can get everyone to speak the same language we can increase communication, develop better designs and decrease the time spent in process development.

7.2.3 Modeling and Simulation

In conjunction with the operations-based language, BDK has modeling and simulation capabilities to determine the effect of the operations. As the flowsheet is constructed BDK simulates the operation being performed on the given set of materials using the underlying models of the operations. The results of the simulation are used by the *Process_Synthesizer* (BDK's conceptual design tool, see Section 7.2.5) as a source of information to allow the operations selection process for the base case design to proceed. The information is also available to the developer so that the developer can assess the design up to the current point as the process is begin developed. The models used are simple design models. The simple design models are sufficient for the early stages of design when processes are being screened using coarse evaluation parameters. Once the base case design is complete, it can then be passed to dynamic simulation programs such as ABACUSS© or gProms©, which have the capability to perform the detailed simulation. This saves time and computation costs since using coarse screening can narrow the number of detailed simulations to just a few.

The compounds within BDK are modeled using what we call the *Material Model.* Since all operations are acting upon materials and it is the transformation of the raw materials into the product materials which is our main concern, models of the materials are essential to the simulation. BDK has links to both commercial and user defined databases of material properties as well as models for estimating these properties. Thus, when applying an operation reliable predictions of the result can be given.

7.2.3.1 BDK Operation Models

As mentioned previously, the BDK operation models are simple models. For vapor-liquid separation operations such as distillation, models like the Rayleigh distillation model (an ideal distillation model) are used. For other operations such as solid-liquid or liquid-liquid separation operations, we do not have property estimation capabilities so we must ask the user to specify the compound distribution among the different phases. For example, for the crystallization operation we must specify the percentage of each compound that is removed as a solid and for the extract operation we ask the user to specify the partition coefficient values. We see that these are relatively primitive models to use for simulation but frequently, during the conceptual design stage in which we are screening alternatives, these simple models are sufficient in order to determine if the process will be profitable and to eliminate unfavorable alternatives.

7.2.3.2 BDK Material Model

The material model [4, 5] is another essential and fundamental portion of the BDK modeling capabilities. The material model is used to model the state of materials in the system. Compounds and mixtures of compounds are known as materials and the material model is an abstract data model used to express the state of the materials (chemicals, mass, temperature, pressure, composition). Since we are performing operations on the materials, the material model must reflect the changes the operations impart on the state of the materials. The material model is used to link and access many data sources which provide the system and user with compound data such as physical property information, health, safety and environmental impact related properties. The physical property information is accessed through the *Material Properties Manager* which is described in the *Material Properties Manager* section.

Table 7.3 provides a partial list of some of the properties contained within the material model. Along with being able to access property information the material model contains procedures for computing properties such as enthalpy and density as well as being able to determine the properties of mixtures using mixing rules. In addition the property calculation capabilities of the material model contains a set of abstract operators which allow the determination of the material behavior when the operations change the process conditions. For example, the material model can determine if a material will form several phases and the distribution of the components into the different phases based on VLE and pure component properties.

Property Group	Property
Physical	IUPAC Name, Alias, Critical Pressure, Critical Temperature, Molecular Structure, Enthalpy of Formation, Total Enthalpy, Heat Capacity, Vapor Pressure, Gibbs Free Energy of Formation, Viscosity
Safety	Auto-Ignition Point, Flash Point, Flammability Limits, Explosive Limits, Corrosivity
Health	Mutagenicity, Carcinogenicity, Teratogenicity, Skin Irritancy, Eye Irritancy, LC-50, LD-50, Blood Agent, Lung Agent
Environmental	EPA Hazardous Waste Number, Octanol Water Partition Coefficient, Soil Adsorption Coefficient, Biological Oxygen Demand, Chemical Oxygen Demand, Anaerobic Biodegradability, EC-50, Bioconcentration Factor

Table 7.3 Sample Properties of the Material Model

The Material Properties Manager

The *Material Properties Manager* is an independent program within the suite of BDK development tools. It is used to manage the database of materials and material properties. While the management of the materials is performed outside of the main BDK environment, the *Material Properties Manager (MPM)* can be called from within BDK and the material property databases are accessed by BDK to provide the required material properties.

The *MPM* accesses the user defined databases as well as other external databases such as the DIPPR© database. The database access is transparent to the user since the Material Properties Manager displays the material properties to the user using a standard format.

The data contained in a compound record is divided into four sections:

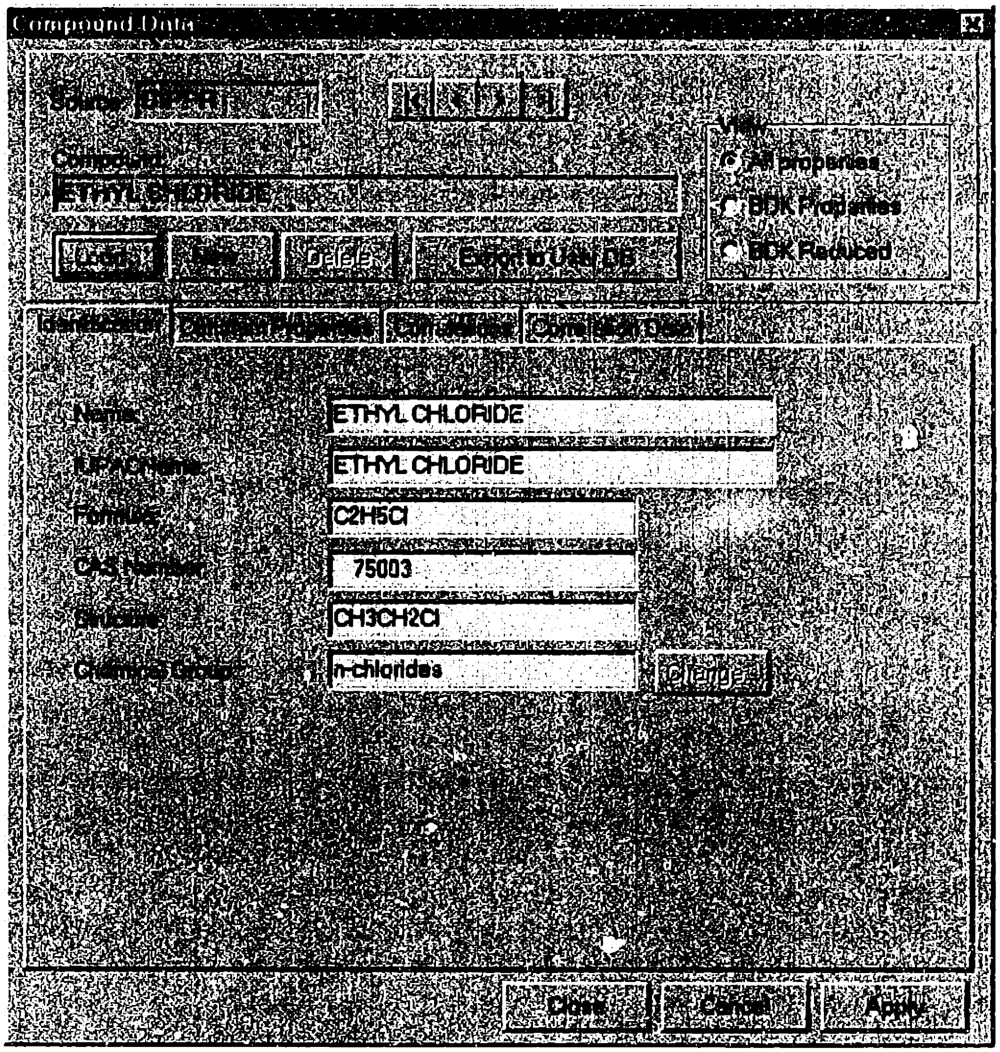
(i) Identification - contains all the ways which a compound can be referenced categorization of the compound.

(ii) Constant Properties - contains all the available constant-valued physical property data i.e. molecular weight, critical temperature and pressure, etc.

(iii) Correlations - contains a list of properties which have value that are functions of temperature and pressure. The data included are the units, a code referring to the function used to calculate the value of the property. A list of equations are provided in the BDK Reference Manual [2]. Some of the properties include density, heat capacity and viscosity.

(iv) Correlation Data - contains the correlation data, or functionality, of the selected physical property from the Correlations section. The correlation data displays the correlation equation and the coefficients used in the correlation equation. The user can edit the coefficients using this dialog.

Within the *MPM*, the data contained in the compound record can be edited, manipulated and stored. It also allows the definition of new compounds along with their corresponding properties. A screen snapshot of the Material Properties Manager is given in Figure 7.4.



Source: DIPPR

Compound: ETHYL CHLORIDE

Name: ETHYL CHLORIDE

IUPAC Name: ETHYL CHLORIDE

Formula: C2H5Cl

CAS Number: 75003

Structure: CH3CH2Cl

Chemical Group: n-chlorides

Figure 7.4 Material Properties Manager

7.2.3.3 BDK and the STN

In Chapter 3 we discussed how the set of process operations and the set of material states are used to construct the State Task Network. The STN is the key that allows us to link the methodological approach to process synthesis with BDK. When a process is manually created using BDK, we are inherently implementing a STN. We are selecting the initial material states and the tasks to be used to generate the states of the process with the goal of generating a state that matches the desired product state. The models contained with BDK are the tasks and the BDK material models are the states. The models move the system from one state to the next by taking the preceding states and generating new states. Illustration 7.2 demonstrates how BDK represents the underlying STN.

Illustration 7.2

For the illustration, let us use a simple process that produces a liquid compound C and a liquid by-product D from a reaction of liquid raw materials A and B. Within BDK the material states are generated automatically based on the models of the tasks selected, the operating parameters supplied by the developer and the current state of the materials in the system. When constructing a process using BDK, we begin with the *charge* operation. The charge operation allows us to generate the start states for the process by automatically generating the *sources* for the process materials. The sources for the materials is a supply of the materials at storage conditions, usually assumed to be standard conditions. We begin by charging raw material A. For the *charge* operation, we must specify the material being charged, the amount and the notional unit to which the material will be charged. BDK automatically generates the source for A and connects the source to the notional unit, which in this case is a reactor. The material state is represented by an round icon. Figure 7.5 contains the BDK flowsheet representation for charging A. The corresponding STN is given in Figure 7.6.

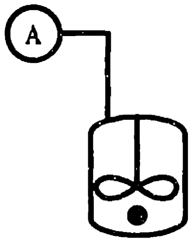


Figure 7.5 BDK Flowsheet for Charge A

Figure 7.6 STN for Charge A

The next operation in our example is a *charge* operation for material B. Let us assume that A and B react spontaneously, so adding B to A will cause a reaction. Both BDK and the STN require the explicit statement of the *react* operation even though the reaction takes place immediately when the reactants are brought into contact. So, the charge operation is selected and the destination of material B is the notional reactor containing A. At this point the BDK flowsheet shows an additional source connected to the notional reactor, given in Figure 7.7, and the STN is shown in Figure 7.8.

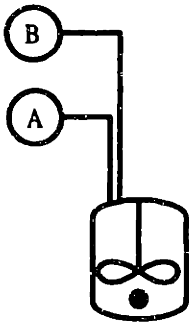


Figure 7.7 BDK Flowsheet Charge B

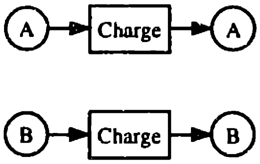


Figure 7.8 STN Charge A and B

Now we tell BDK that A and B react by selecting the *react* operation. The model behind the react operation performs the reaction based on the conversion of the key reactant and the stoichiometry. In this case let us assume that we have 100% conversion and we have added no excess reactant, thus all the raw materials are converted to products. The reaction model acts upon the material models and generates the new state, which is a mixture of C and D. Included in the new state information is the amounts of C and D produced, the amount of unreacted raw materials, the phases and phase distribution of the materials as well as the final temperature and pressure. While the change is not reflected on the BDK flowsheet explicitly, we detect the change in state of the system by displaying the properties of the material contained in the notional reactor. The STN also reflects the change as shown in Figure 7.9.

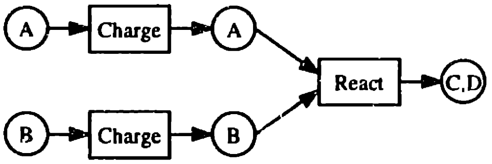


Figure 7.9 STN Including the React Operation

The final step is to separate C from D. Let us assume that batch distillation can be used perform the separation and that C is the light component. The batch distillation model requires the developer to set the operating parameters as well as the notional equipment for the distillation. The resulting BDK flowsheet is shown in Figure 7.10 and the corresponding STN is given in Figure 7.11.

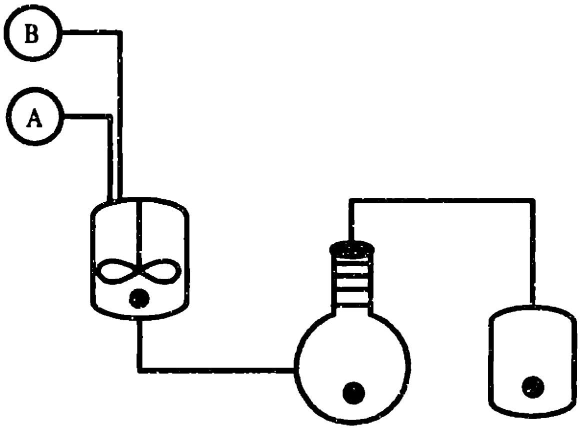


Figure 7.10 BDK Process Flowsheet for the Production of C

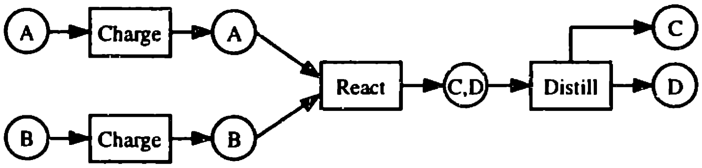


Figure 7.11 STN for the Production of C

It should be clear from Illustration 7.2 that the differences between the initial state and the goal state drive the development of the process. The process is made up of tasks and the selection of the tasks is based on the models of the operations and materials. The development of the process is essentially the development of the STN and underlying the BDK process flowsheet is the STN of the process. The importance of the models used in the process development can not be underestimated since the results of the application of the models directly impacts the final format of the STN and thus, our final BDK process flowsheet. In the next section we will discuss the BDK environment in detail.

7.2.4 The BDK Environment

BDK supports the process development activity from the initial stages of a process idea through the definition of a manufacturing system. BDK uses a four step procedure for the computer-aided development of a process design. The four steps are:

1. Project Definition and Route Planning
2. Stage Refinement

3. Facility Allocation
4. Process Analysis based on Material Assessment and Treatment Selection

7.2.4.1 Project Definition and Route Planning

All knowledge accumulated through the course of the development of a new process, from the initial reaction sequence to the final manufacturing scheme, is contained in the BDK *Project* and its associated databases. The project is made up of one or more process *routes* or *process versions*, which are the alternative synthesis methods to produce a single product. Figure 7.12 is a screen snapshot of the *Project Dialog* which displays the project, route, stage hierarchy.

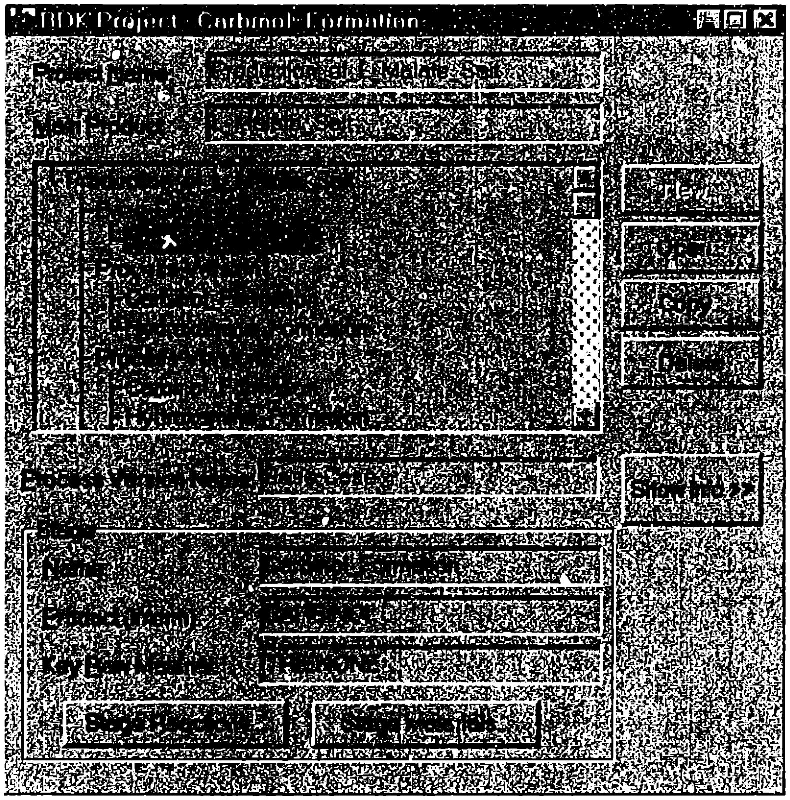


Figure 7.12 BDK Project Dialog

Each route is made up of a stage network where each stage generates a specific intermediate. These intermediates serve as precursors to the subsequent stage. In planning the route, the designer must specify the reaction pathways and the overall yield of the constituent stages. Reaction path identification is currently not supported by BDK thus the reaction path must be specified by the developer. Each stage allows for parallel competitive reactions or sequential reactions. Along with the definition of the reactions, the user must also specify the process materials. The process materials are the compounds that are going to be used in the process. These materials include the raw materials, solvents, catalysts, products and by-products. These compounds are either taken from the databases that accompany BDK or they can be specified by the user. The management of the compounds is done through a tool called the *Material Properties Manager*. The Material Properties Manager will be discussed in a following section. BDK generates the route structure and closes the overall mass balances for the stage network. Figure 7.13 provides an example route structure for the pharmaceutical process that includes the Carbinol stage.

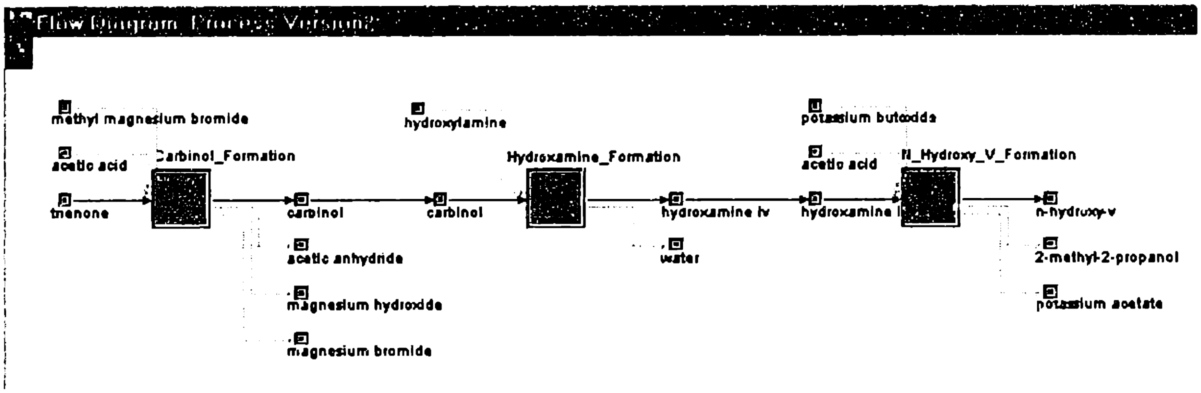
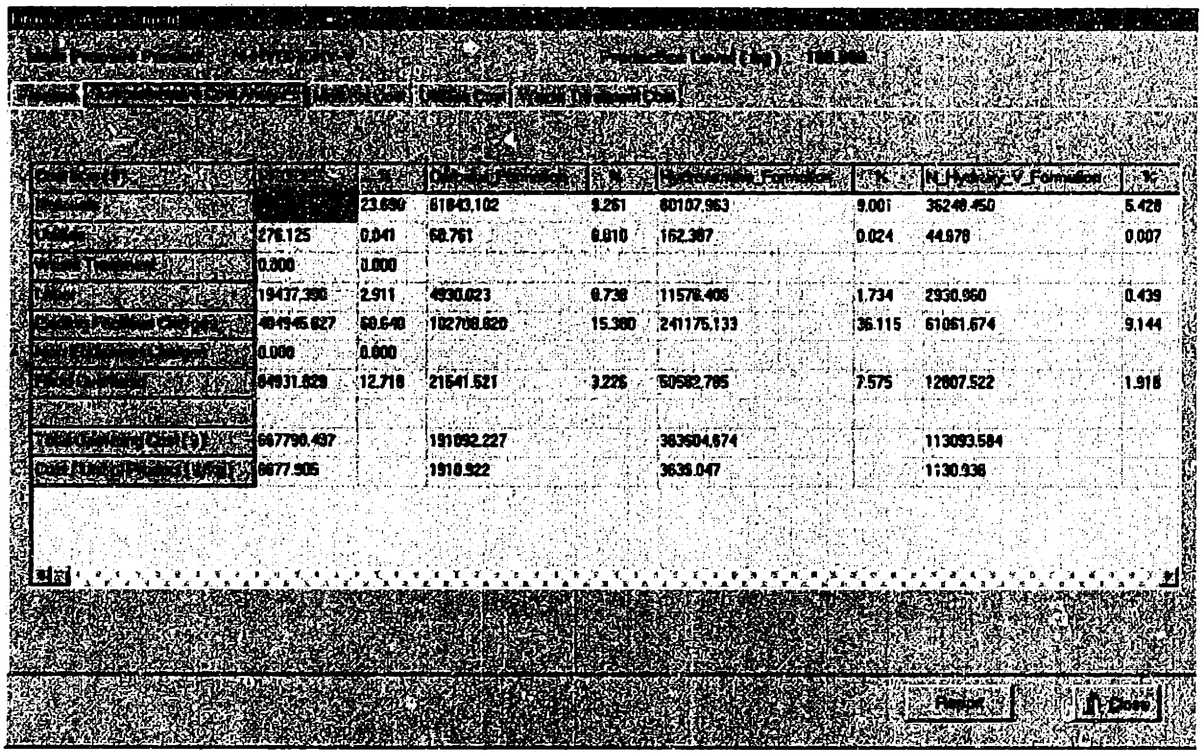


Figure 7.13 BDK Route Structure

Early in the design, different promising production routes can be submitted to a preliminary economic evaluation based on projected or required yield with respect to the stage intermediate and raw material prices. With a preliminary calculation of material costs, the user can do production planning based on raw material prices and initial material impact screening based on stage materials and by-products. Figure 7.14 provides an example of the BDK economic analysis dialog generated from a route level analysis.



[illegible]	[illegible]	%	[illegible]	%	Hydroxylamine_Formation	%	N_Hydroxy_V_Formation	%
[illegible]	[illegible]	23.690	61843.102	9.261	80107.963	9.001	36248.450	5.428
[illegible]	276.125	0.041	68.761	0.010	162.387	0.024	44.978	0.007
[illegible]	0.000	0.000						
[illegible]	19437.390	2.911	4930.023	0.738	11578.406	1.734	2930.960	0.439
[illegible]	404945.627	60.640	102708.820	15.380	241175.133	36.115	61061.674	9.144
[illegible]	0.000	0.000						
[illegible]	84931.829	12.718	21541.521	3.226	50582.785	7.575	12807.522	1.918
[illegible]	667790.497		191092.227		363604.674		113093.584	
[illegible]	6677.905		1910.922		3636.047		1130.936	

Figure 7.14 BDK Route Level Economic Analysis Dialog

7.2.4.2 Stage Refinement

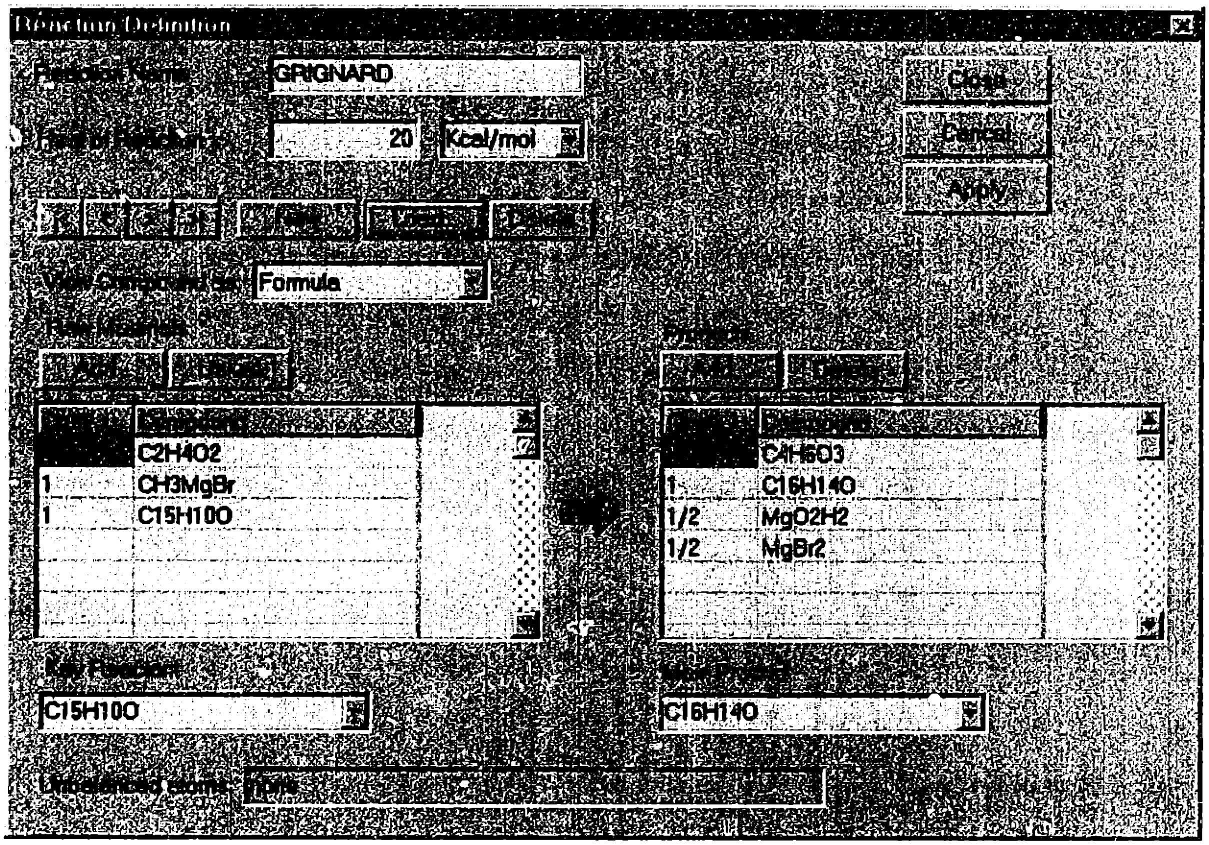
After the structural layout of the process route and its stage network, promising routes have to be refined to determine the required operational steps. In general, a stage consists of three abstract tasks: (i) the preparation of the raw materials, (ii) the reaction steps and (iii) product recovery and purification through separation. These abstract tasks are articulated through the specification of the necessary operations to accomplish these tasks. The operations-based language is the medium of communication used for the stage construction. The specification of the operations can be performed manually by the developer or the developer can use the process synthesis tool that has been developed as part of this thesis. The process synthesis tool, called the *Process_Synthesizer* will be discussed in Section 7.2.5. The stage is refined through the selection of operations and filling in the required information into the operation dialog. BDK automatically constructs the batch sheet and the flowsheet simultaneously as well as simulates the operation generating the resulting material state.

Preparation of Raw Materials

The preparation of raw materials consists of defining the operations that act upon the raw materials in order to either purify them or transform them into a state that is more readily used in the reaction steps. For example, methylmagnesium bromide is highly reactive so it is generally shipped dissolved in a solvent such as ethyl ether. If ethyl ether is not a solvent that is appropriate for the reaction we wish to perform then we must perform a solvent replacement step in order to prepare the raw material for reaction. Another common raw material preparation step is to dissolve solid raw materials before reaction.

Reaction Steps

The reaction steps should need little explanation. These are the operations that perform the reaction as well as supporting operations such as cooling during reaction or age until the reaction is complete. The developer is required to specify the reactions. The reactions can specified during the route construction or during the stage construction and are stored in a database. Figure 7.15 contains a screen snapshot of the Reaction Dialog for the Grignard reaction.



Reaction Name: GRIGNARD

Heat of Reaction: 20 Kcal/mol

View Compound as: Formula

Raw Materials

	C2H4O2
1	CH3MgBr
1	C15H10O

Products

	C4H6O3
1	C16H14O
1/2	MgO2H2
1/2	MgBr2

Key Reactant: C15H10O

Main Product: C16H14O

Unbalanced atoms: none

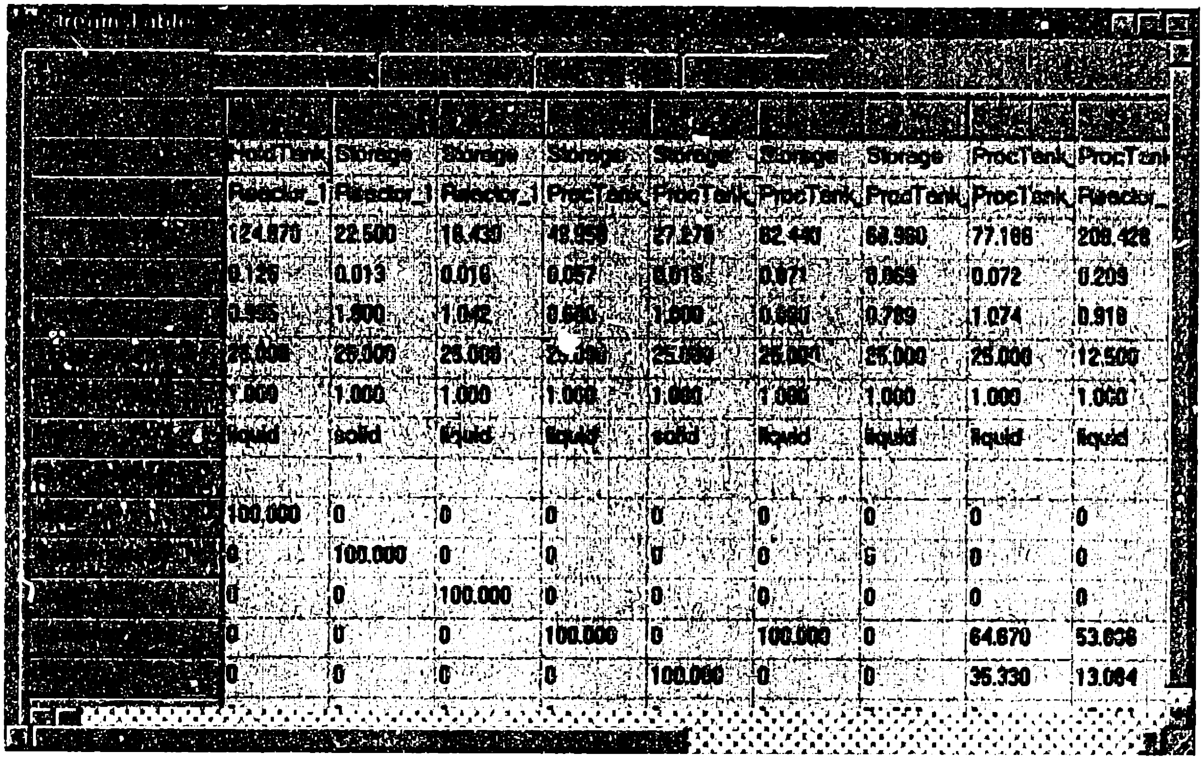
Figure 7.15 BDK Reaction Definition Dialog

Product Recovery and Purification

Finally we have the product recovery and purification task. This is generally where the largest number of operations are used and the separation systems can be quite complex. If we go back and examine the carbinol stage from Chapter 4, we see that most of the operations are used to purify the product. These operations include Distill, Filter, Crystallize, etc. of course, like all operations, the separation operations can require support operations such as Heat, Cool, Age, etc.

The end result of the stage refinement is the production recipe including the batch sheet and flowsheet. The information produced includes a list of process operations, operating conditions, material and energy balances along with the notional equipment and their material content. Figure 7.3 provided a screen snapshot of the BDK process flowsheet and batch sheet. If we look carefully we can notice the correspondence of the operations on the batch sheet and the notional equipment on the flowsheet.

At this point the developer has many options for process analysis. The developer can generate stream tables to examine material flows and study the process wastes that are produced. The BDK stream table is given in Figure 7.16. The developer can bring up the process sequence diagram to begin the analysis of potential equipment usage issues by displaying a two-dimensional depiction of the notional equipment and the duration of the operations that are taking place within these pieces of equipment. Figure 7.17 contains a screen snapshot of a process sequence diagram for the carbinol stage. The equipment is arranged horizontally and the operations are listed under each notional equipment item. Also contained on the process sequence diagram is the estimated volume required by each operation and the duration of the operation along with the connectivity between the operations which allows the reader to visually determine the ordering of the operations. Additionally, the developer can generate a Gantt chart which displays the duration of each process operation within physical equipment. The Gantt chart (not shown) allows the developer to begin an analysis of the scheduling of operations and equipment.



[illegible]	Storage	Storage	Storage	Storage	Storage	Storage	ProcTank	ProcTank
Reactor_1	Reactor_1	Reactor_1	ProcTank	ProcTank	ProcTank	ProdTank	ProcTank	Reactor_
124.870	22.500	16.430	40.950	27.270	62.440	68.960	77.186	208.428
0.126	0.013	0.016	0.057	0.015	0.071	0.069	0.072	0.209
0.986	1.800	1.042	0.880	1.800	0.880	0.789	1.074	0.918
25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	12.500
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
liquid	solid	liquid	liquid	solid	liquid	liquid	liquid	liquid
100.000	0	0	0	0	0	0	0	0
0	100.000	0	0	0	0	0	0	0
0	0	100.000	0	0	0	0	0	0
0	0	0	100.000	0	100.000	0	64.670	53.836
0	0	0	0	100.000	0	0	35.330	13.084

Figure 7.16 BDK Stream Table

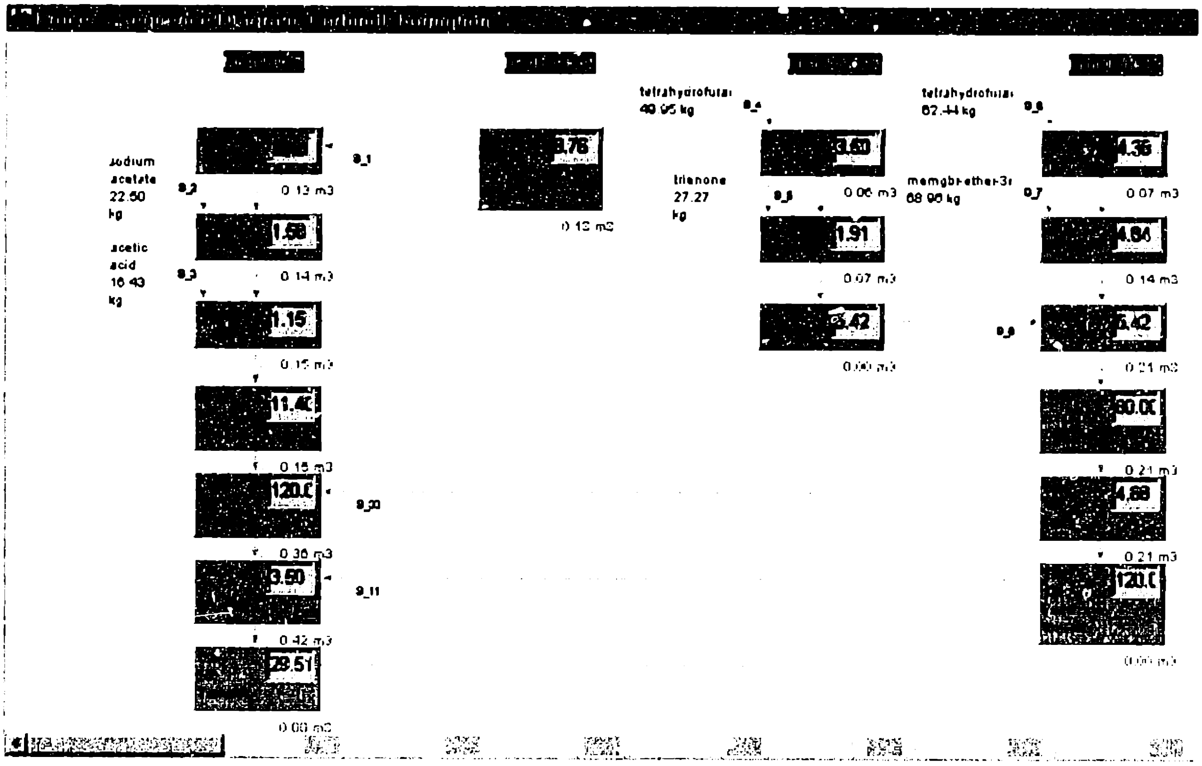


Figure 7.17 BDK Process Sequence

7.2.4.3 Facility Allocation

After the definition of promising processing schemes through the operations-based language, we have created the stages. The batch sheet now holds the sequence of operational steps, i.e. a state task network description of all tasks and materials. The flowsheet depicts a graphical representation of the notional units required by the chemical recipe and representing the process operations. Mapping these abstract units into existing production facilities is the objective of the *Facilities Allocation Manager.*

The Conceptual Model of Facility Allocaion

The allocation of the process recipe into a process facility is in essence a *matching* problem where the requirements of the operations in the process recipe are matched with equipment which have the necessary characteristics to carry out the operations. A conceptual model of the facility allocation problem has been formulated by Salomone et al [6] and is depicted in Figure 7.18.

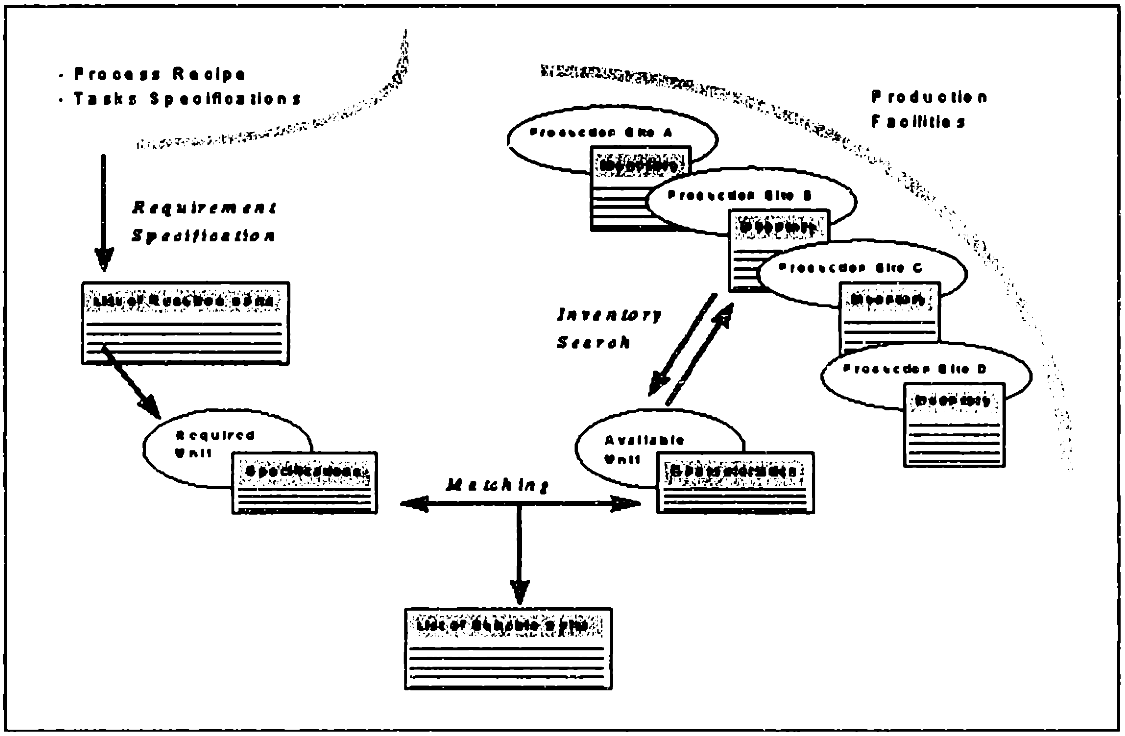


Figure 7.18 Facility Allocation Conceptual Model Salomone et al [6].

We begin the facility allocation problem with the process recipe, which includes the task and process material specifications. Along with the process recipe and the specifications, we also have a list of production facilities and a corresponding equipment inventory for each facility. The equipment inventory states the number and types of equipment as well as the corresponding characteristics for each piece of equipment such as volume, material of construction, etc. From the process recipe and specifications, we can develop a list of required units, which is a list of units with all the required features for performing the specified tasks. For example, if one of the tasks in the process recipe is a batch distillation, the required unit would be a batch distillation unit with the appropriate characteristics to perform the separation of the material in the process, these characteristics include the appropriate volume, material of construction and utility services such as vacuum.

Once we have a list of required units, we select one of our production facilities and apply a matching algorithm which compares each of the required units with the equipment, called the *Available Units*, on the facility inventory list by matching the characteristics of the required unit with those of the equipment. During the matching process, any equipment within the facility inventory that does not have the features required by the required unit is removed from consideration. Upon completion of the matching, we are left with a list of available units, called the *Suitable Units*, for each task. Each available unit in the suitable units list is a piece of equipment with all the necessary features required by the required unit. If more than one unit is available from the facility inventory, we can then define selection criteria, such as cost, to select the best unit from the list of suitable units for the task. If the suitable units list is empty for a given task, this indicates that the facility under examination does not contain a piece of equipment which meets the requirements of the task in order to perform the task. In this case, the allocation facility generates a generic type of equipment, called a *New Unit*, and assigns the new unit the list of required features as indicated by the required unit. The generation of the new unit allows the developer to determine the cost and type of equipment that must be added to the facility under consideration in order to perform the process. An outline of the logic used in the facility allocation is given in Table 7.4.

Facility Allocation Logic

1) Enumerate all Required Units
2) For each required unit:
    a) Specify the equipment class
    b) Specify all required features
3) Select a production site
4) Obtain all units of the requested type available in the specified site
5) For each required unit:
    a) Obtain the feasible set of available units
    b) Choose the best available

Table 7.4 Facility Allocation Logic from Salomone et al [6].

In order to accomplish the facility allocation, the above process model has been subdivided into five components, each performing a different part of the facility allocation task within the *Facilities Allocation Manager*. The five components are:

1. The *Production Site Manager* - allows for the creation of a database of production sites and the individual production areas within those sites.
2. The *Equipment Characterization System* - allows for the definition of equipment classes and types along with specific pieces of equipment and their properties.
3. The *Facilities Inventory Manager* - allows the allocation of equipment to specific areas of the different production facilities.
4. The *Scenario Manager* - contains the logic for the automated mapping of the notional equipment to actual equipment in a specific facility.
5. The *Facilities Cost Manager* - allows for the creation of a plant dependent cost database so that costs can be assigned to the various facilities.

In the following sections we will discuss each of these tools in more detail.

The Production Site Manager

The Production Site Manager allows us to model the production facilities at a specific location. The manufacturing plant, its plant subsections and specific production buildings are stored in a customizable relational database. A location path nomenclature is used for referencing of site locations and their subplants. Figure 7.19 shows a sample tree of the production facilities available to our example pharmaceutical firm. We see from the figure that a production facility can be decomposed into numerous subsections such as buildings, areas and lines.

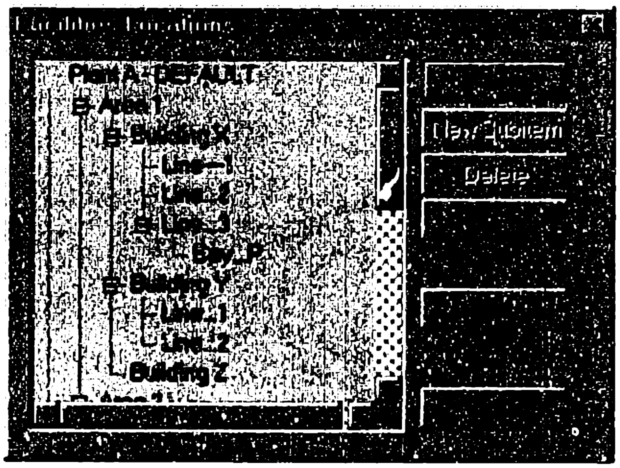


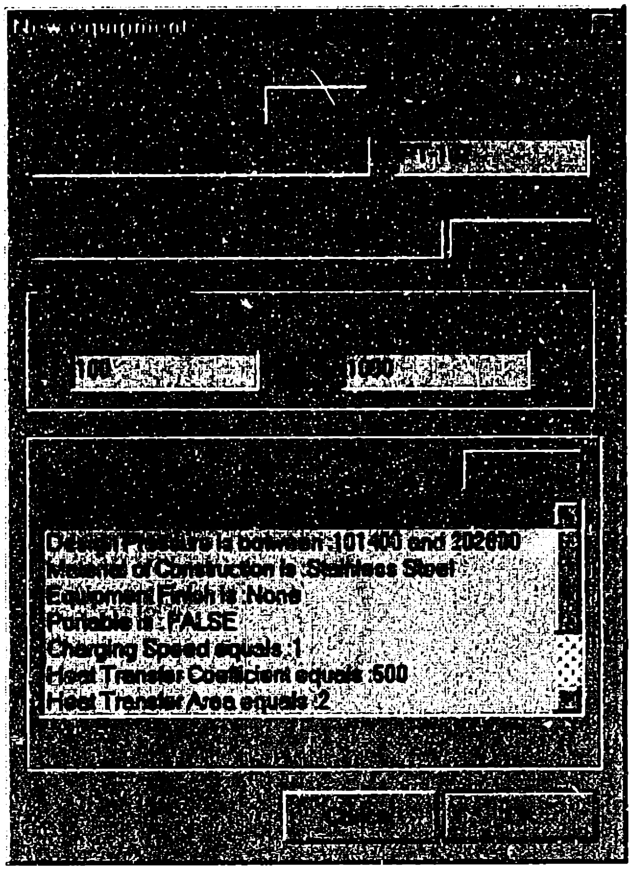
Figure 7.19 BDK Facilities Locations Dialog

The Equipment Characterization System

For the identification of equipment types as they are available in multi-purpose batch manufacturing facilities, an array of specifications influences the choice of equipment. These required features may vary from project to project, depending upon the choice of chemicals and the operating conditions. Different production facilities may offer particular services through specialized equipment types. This multitude of possible specifications and properties led to the need for a general characterization system that provides support for the definition of particular equipment features and the development of a class hierarchy of equipment types.

The Definition of Equipment Features

BDK distinguishes between general and equipment features. The general features are the features that are in common among all equipment types. These features include equipment type and the asset identification for the unique identification of each piece of equipment, location identifier in order to designate the location of the equipment to a specific manufacturing facility and volumetric capacity with upper and possibly lower operational bounds. The specific features provide the possibility to define special properties or services provided by a type of equipment. The scope and number of special features is not limited by the BDK system, but should include all relevant properties for equipment selection. Each property must be modeled by one of four types: (i) symbolic, for the definition of a string property such as the value Carbon Steel for the equipment material of construction property, (ii) single value for the definition of numeric single valued properties such as 50 m^3 for the property Heat Exchange Surface Area, (iii) range for the definition of a range of values acceptable for a given property such as 200K - 500k for the Operating Temperature Range property and (iv) Boolean for properties that accept only Boolean values such as the property Portable. Figure 7.20 is a screen snapshot of the BDK New Equipment Dialog showing the equipment features for the process tanks.



Design Pressure is between :101400 and 202600
Material of Construction is :Stainless Steel
Equipment Finish is :None
Portable is :FALSE
Charging Speed equals :1
Heat Transfer Coefficient equals :500
Heat Transfer Area equals :2

Figure 7.20 BDK New Equipment Dialog

Development of the Class Hierarchy of Equipment

Each of the features described by the equipment characterization system can assigned to a specific type of equipment. In order to structure the numerous equipment types available in multi-purpose plants, the user can generate a tree hierarchy of equipment classes. The instances of each equipment class share common features. The equipment lower in the hierarchy inherit the attributes from their parents. The equipment class hierarchy can be dynamically generated and is stored in a relational database for future use. Figure 7.21 shows the class hierarchy of equipment. Notice that all the equipment stems from the class Generic Type. From this generic equipment class we have general equipment categories such as Tanks and Filters and then within each of these categories we have more detailed classes until finally we have specific equipment types like Jacketed P/V-Reactor or Nutsche Filters.

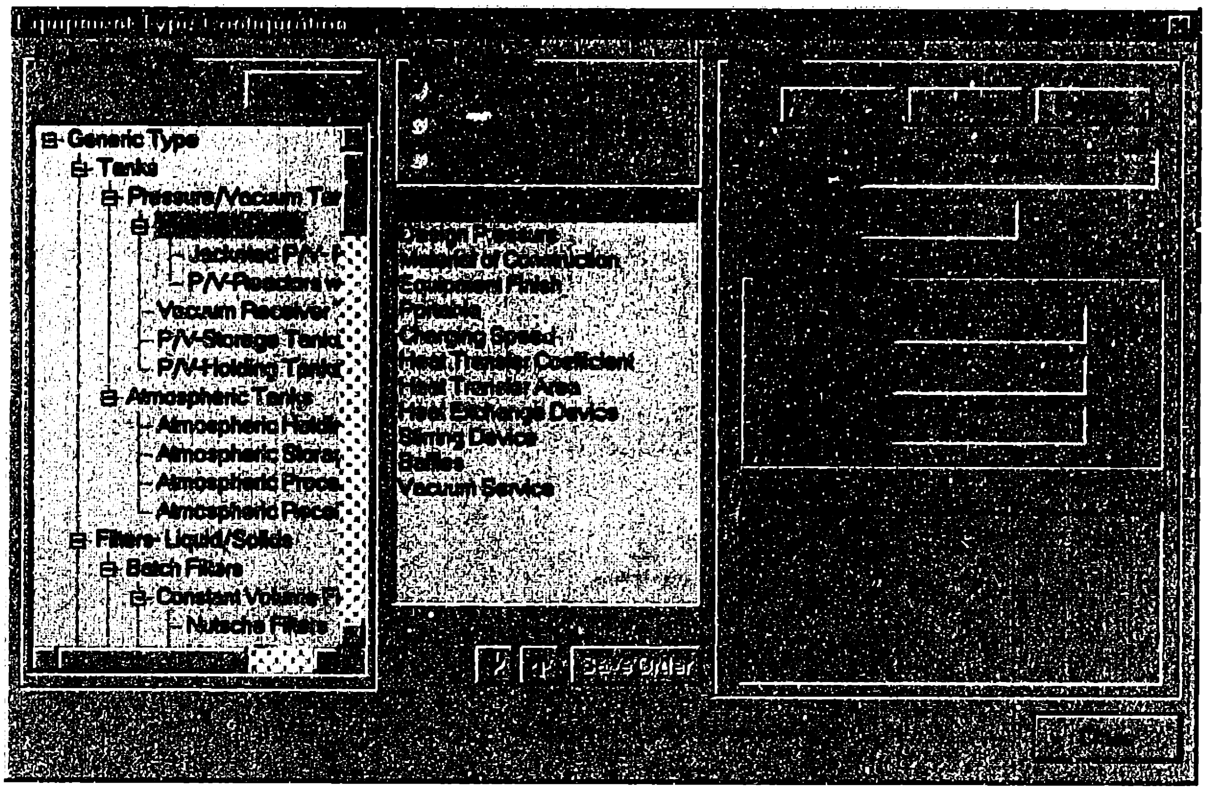


Figure 7.21 BDK Equipment Class Structure

Facilities Inventory Design

Once the definition of the equipment classes and the database of locations is complete, the designer can populate the production database with specific instances of equipment taken from the equipment inventory of the existing facilities. Figure 7.22 is an example of the equipment list for a specific location, Plant A-Area1-BuildingX-Line1.

Examining Figure 7.22 we see that vacuum receiver tank VR-X103 has been selected and its selected features and values are displayed.

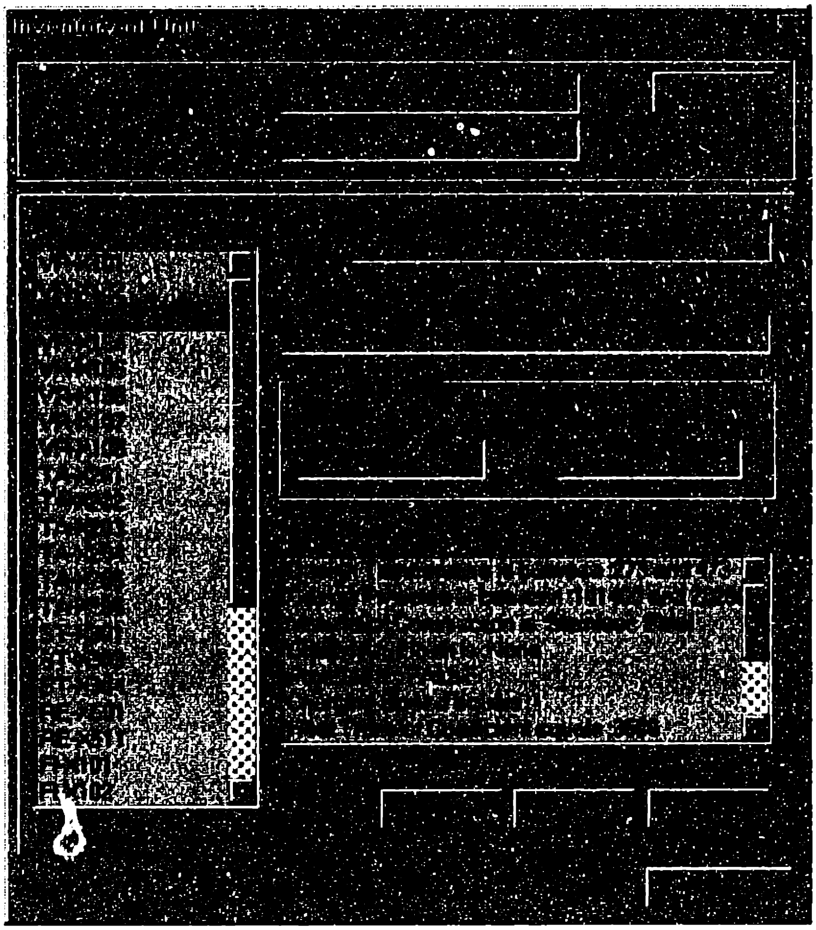


Figure 7.22 BDK Equipment Inventory

The Scenario Manager

With the model of the production site, we return to the batch sheet which lists the operations. The corresponding flowsheet only depicts the notional units. The designer can now allocate the physical equipment and investigate the feasibility associated with the implementation of the process at an existing production site. The mapping algorithm traverses four steps illustrated in Figure 7.23. The four steps are:

1. Definition of Equipment Requirements
2. Selection of the Site Location
3. Automatic User Defined Allocation
4. Display of Results

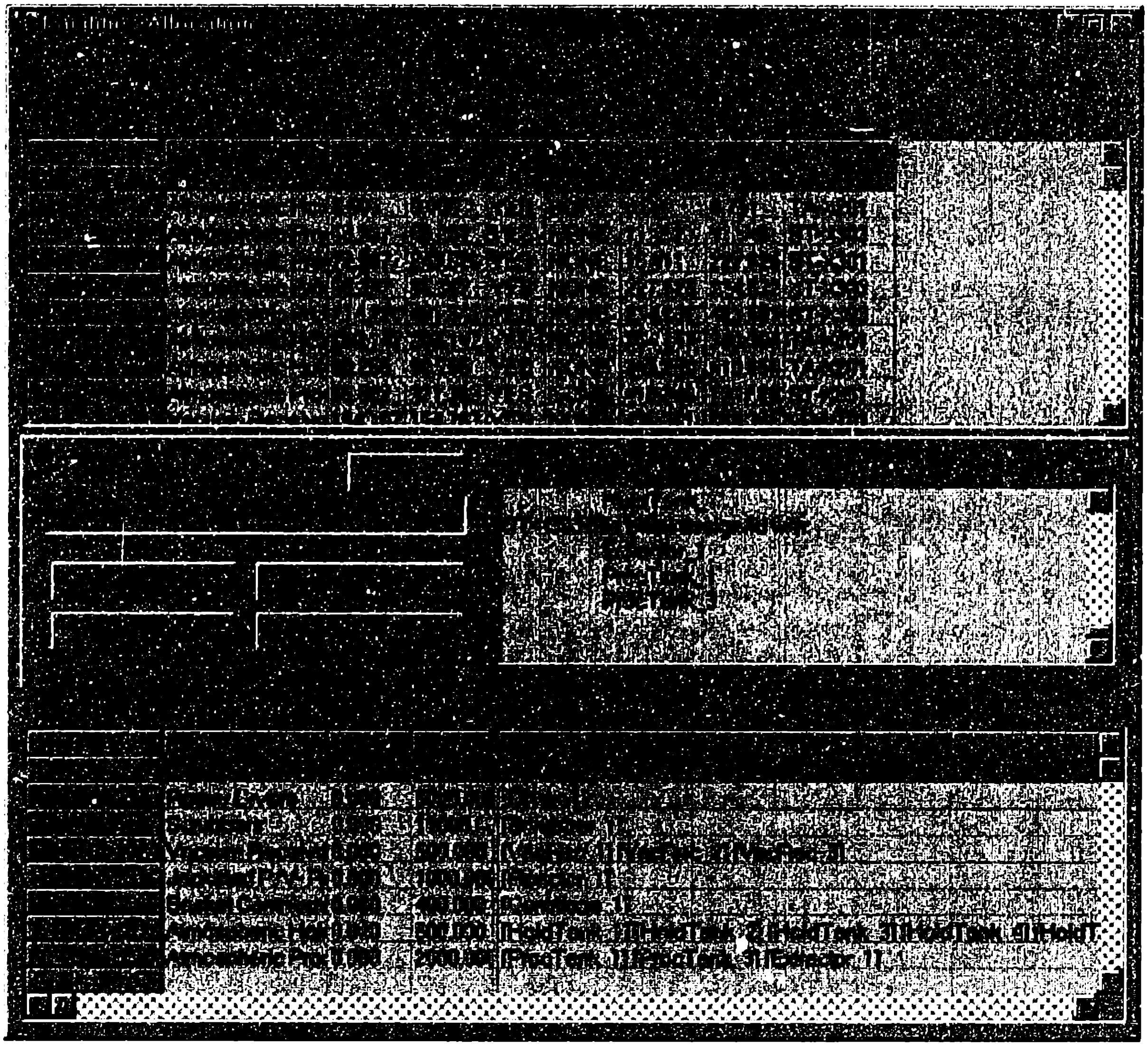


Figure 7.23 BDK Scenario Manager

Definition of Equipment Requirements

From the calculated material flow information, BDK computes a list of key requirements. BDK displays the Facility Allocation table and compiles the key requirements which are: (i) a list of notional vessels to be mapped into existing equipment, (ii) minimum and maximum volume requirements as calculated from the operations and (iii) equipment occupancy duration measured by the operation start and end times. In addition to these key requirements, the designer can specify supplementary features to better reflect the needs of a specific operation with respect to safety or hazardous material considerations.

Selection of Site Location

In this step the designer selects a plant or a subplant location to which the process will be allocated. This location is the location from which the equipment inventory will be used in the facilities allocation. The designer can select a number of different locations in order to perform a comparison and determine the most desirable location to place the process. Notice the location path in Figure 7.23. From this path it is easy to see the exact location to which the designer will allocate the process.

Facility Allocation

At this point the facility allocation is performed. There are two modes of operation for the facility allocation: automatic or manual. In automatic mode, BDK attempts to assign the operations to specific pieces of equipment such that all the requirements are met. The selection algorithm follows a combinatorial logic for generating the best configuration based on equipment availability and the use of a minimal amount of equipment. We have provided a description of this logic in a previous section and is described in more detail by Salomone et al [6]. Alternatively, in manual mode, the designer can manually allocate the process operations to specific pieces of equipment. This progression through the allocation can be monitored in the *Message Board* of the scenario manager as shown in Figure 7.23.

Allocation Results

The *Allocated Equipment Table* displays the results of the allocation algorithm, see Figure 7.23. If no equipment with the required features is available, BDK then assigns the operation to a "new unit". This indicates to the designer that he must either select another facility to perform the process or install an additional piece of equipment that has the required features to the existing facility. The results can also be view graphically in the form of a Gantt chart (not shown) which reflects the allocation of the notional equipment (horizontal) to the physical equipment (vertical), along with the duration of the operation (horizontal bars).

Facilities Cost Manager

When allocating the process to a specific location we would like to be able to get an estimate of the costs associated with that facility. Within BDK we have chosen to allow the estimation based on a Facilities Charge per day. This provides a first estimate of the costs associated with a given facility. Within BDK we maintain a database of the relevant costs for each facility or specific location within a facility. These include the labor costs for the facility in dollars per hour, the facilities charge in dollars per day and the fixed overhead rate which is a percentage of the total cost. We also allow the developer to maintain the utilities costs for the facilities and the waste treatment costs which can vary significantly from location to location. Figure 7.24 shows the Plant Dependent Costs dialog for a location within the BDK system.

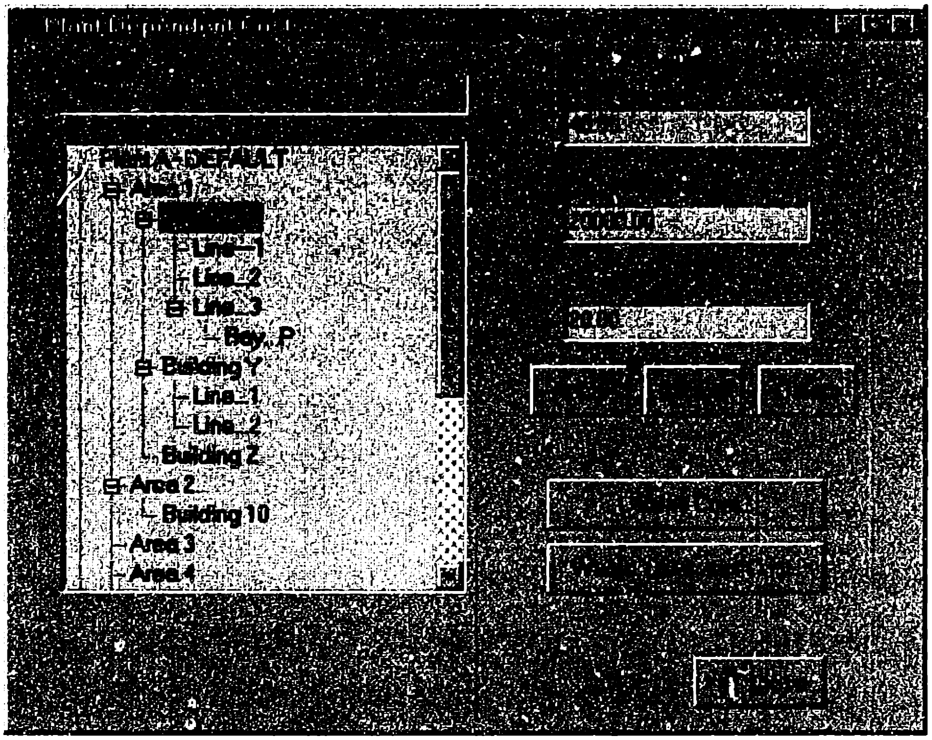


Figure 7.24 BDK Plant Dependent Costs Dialog

7.2.4.4 Materials Assessment and Waste Treatment Selection

Two additional tools to be incorporated into BDK are the *Materials Assessment* tool (MAT) and the *Waste Treatment Selection* tool (WTS). The MAT and WTS are both knowledge-based systems that analyze materials generated by the process. The MAT system performs an assessment of all process streams with respect to seven major U.S. environmental regulations. It identifies all the regulations that apply to a given stream to make the developer aware of the potential regulatory problems he may encounter. Using this information, the process developer can work with the chemist at an early stage to try to make changes to the process in order to remove any violated regulatory constraints. For example, the developer can work with the chemist to select a less toxic or more environmentally benign solvent to use in the process. The WTS examines all waste streams produced by the process and suggests appropriate treatment alternatives. For example, it suggests sending organic wastes to bio-treatment ponds, while organic waste contaminated with bio-toxins cannot be sent to the bio-pond so it suggests operations for the removal of the toxins before sending it to the bio-pond. The WTS and the MAT system both contribute to the interactive feedback concept discussed earlier, where screening can be used to make adjustments to the process as early as possible in an attempt to develop better designs. Also, the MTS can be used to guide the WTS through the selection of mixtures and compounds that should be sent to waste treatment. Both the WAT system and the MAT system have been implemented within BDK. While it is considered important, the MAT system has not been ported to the latest BDK platform.

The Materials Assessment

The objective of material assessment is to provide the designer with all relevant health, safety and environmental information regarding the compounds used within the process under consideration, so that the designer can screen process alternatives based on the regulatory impact. The materials assessment plays a crucial role in the development of a process since violations of regulations can be costly, not only from a financial point of view with fines and plant shut down, but also with respect to the corporate reputation in public relations with the surrounding community. Additionally, serious violations can leave the company vulnerable to a costly litigation process. Thus, it is important for process developers to be able to assess a process with respect to health, safety and environmental impact considerations. The earlier the developer is aware of potential regulatory violations, the more flexibility and options the developer will have to make process modifications in order to avoid future problems. Finally, many aspects of chemical manufacturing require various levels of permitting and reporting, the materials assessment tool can be a valuable assistant to provide information for obtaining the required permits and determining if reporting is required.

The assessment of materials is composed of two tasks: (i) assessment of all the individual compounds and (ii) assessment of process mixtures. The assessment of materials is composed of rule-based tasks and screening tasks formulated from federal, state and local regulations regarding the uses of chemicals. Since all the regulations typically have the same format, we have chosen to include only the seven major federal regulations listed in Table 7.5, allowing for the future customization of the material assessment in order to incorporate the state and local regulations. In the following sub-section we will give a brief overview of each of the regulations listed in Table 7.5.

Source of Regulation
• Clean Air Act (CAA)
• Clean Water Act (CWA)
• Resource Conservation and Recovery Act (RCRA)
• Toxic Substances Control Act (TSCA)
• Comprehensive Environment Response Compensation and Liability Act (CERCLA)
• Superfund Amendments and Reauthorization Act (SARA)
• Occupational Safety and Health Act Hazard Communication Standard (OSHA HCS)

Table 7.5 List of Federal Regulations

Sources of Federal Health, Safety and Environmental Regulations

An inherent part of chemical manufacturing is ensuring compliance with all applicable federal, state and local regulations governing the manufacturing process, including the chemicals used and produced, the handling of all materials used in the process, ensuring wastes are handled in an appropriate manner and ensuring the health and safety of the workers, the surrounding community and the environment. While we recognize that the state and local regulations are often more stringent than those imposed by the federal government, using the federal regulations as the basis for the MAT allows us to create the framework for the system and later the system can be extended to include any other set of regulations.

The federal government has created vast sets of regulations under the authority of numerous federal agencies. The size, scope and complexity of the sets of regulations makes the task of ensuring compliance difficult. As an example of the complexity of the regulations, each regulation has its own definition of a "hazardous" chemical. It is the goal of the Material Assessment Tool to act as an assistant to the developer in order to allow the developer to have the necessary information to ensure that the process developed is in compliance with all appropriate regulations. We should note that while it is feasible to create a system that contains knowledge about all the federal regulations, this is a large and time consuming task. With the MAT that we have developed, we have focused on those sections of the regulations that apply to the batch process industry. Also, since the compliance with regulations is such an important matter, any system developed requires extensive validation in order to ensure the accuracy of the results. Thus, the system we have developed should be considered a prototype system, since it is neither complete nor has been validated. A final problem with developing a system that is based on regulations, is that the regulations are constantly being updated thus, keeping the system current is also an important part of the task. In the following sub-sections we will provide a brief description of the purpose of each of the regulations listed in Table 7.5 as well as how each of the regulations in implemented in the MAT.

The Clean Air Act

The objective of the *Clean Air Act* (CAA), 42 U.S.C. § 7401, is to protect the air quality of local areas around a facility as well as the national air quality. The agency in charge of regulation and enforcement under the CAA is the Environmental Protection Agency. The CAA empowers the EPA to establish air emission limits for various air pollutants and to establish national air quality standards. Under the CAA an air pollutant is defined as any substance released into the outdoor air, which has the potential to endanger public health or welfare. This includes effects on soil, water, vegetation, weather, visibility, climate, harm to property and personal discomfort. [7] As with all regulations, the CAA is broken into sections called Titles. For the batch chemical production industry the most relevant portions of the CAA are Title III: Hazardous Air Pollutants and Title VI: Stratospheric Ozone Protection. Title III sets emission limits on 172 substances and 17 groups of chemicals based on the source industry. Both the pharmaceutical and agricultural chemical industries are listed. Title VI sets reporting and emission standards for ozone depleting chemicals. As required by the CAA, permits are required for air emissions and air emissions include fugitive emissions thus, an accurate accounting for all process materials is required. Permitting guidelines for the sources of air emissions are stated both in Title III and Title VI.

The implementation of the CAA into the MAT requires the ability to screen lists of materials stored in databases. Listed materials are the simplest to deal with since all that is required is a database search. Lists also provide an additional advantage of being easy to update and maintain. With the internet, automatic updating of regulatory lists is possible by linking to online databases. Once we identify the source industry, we can screen the process under development to see if there is the possibility of emissions of materials which are regulated under the CAA. Additionally, we can maintain an inventory of regulated materials used in the process, which we can use to assist in preparing permit applications for the facility.

The Clean Water Act

The *Clean Water Act* (CWA), 33 U.S.C. § 1251, was enacted in order to protect the waters of the United States from pollution with the goal of completely eliminating the discharge of pollutants into all navigable waters. [8] The CWA prohibits the discharge of any pollutant with a *National Pollutant Discharge Elimination System* (NPDES) permit. Under the CWA a pollutant is defined as *any dredged soil, solid waste, incinerator residue, filter backwash, sewage, garbage, sewage sludge, munitions, chemical waste, biological materials, radio active materials, heat, wrecked or discarded equipment, rock, sand, cellar dirt and industrial, municipal, and agricultural waste discharge into water.* [9] As we can see, this definition of a pollutant covers a wide range of materials. The CWA covers both *direct discharges*, those discharged directly into a body of water, and *indirect discharges*, those discharged into *Publicly Owned Treatment Works* (POTW). The discharge limits are based on the type of industry and the characteristics of the material discharged. For example, for fermentation products, the pH of any discharged stream be within the range of 6.0-9.0 standard units. The CWA strictly prohibits dilution as a method for meeting effluent limitations. The CWA mandates the reporting of discharges of 65 listed toxic pollutants and an expanded list of 126 priority pollutants and specifies pretreatment requirements for discharges to POTWs.

The implementation of the regulations contained in the CWA in the MAT requires both the ability to screen lists of materials as well as the ability to interpret rules as in the example given above. The rule-based screening of waste water streams requires the ability to estimate mixture properties such as Biological Oxygen Demand (BOD), Carbon Oxygen Demand (COD) and pH. Also, we need to have some accounting of the rate of discharge. For example, for fermentation products the limit on discharge is based on the daily average mass of BOD in a calendar month. Again, the information collected in order to perform the assessment will also be useful in completing any required permit applications.

The Resource Conservation and Recovery Act

The *Resource Conservation and Recovery Act* (RCRA), 42 U.S.C. § 6901, was enacted in order to regulate solid and hazardous wastes and encourage resource recovery and the management of non-hazardous wastes. The two subtitles which are most relevant to the batch chemical processing industry are Subtitle C and Subtitle D. Subtitle C regulates the generation, transport, treatment, storage and disposal of solid hazardous waste. The important aspects of Subtitle C relevant to batch processing include (i) a tracking system designed to track all hazardous wastes from their point of generation through final disposal and (ii) a permitting system to regulate the storage, treatment and disposal facilities on a site-specific basis. [10] Subtitle D regulates non-hazardous wastes. As with most of the regulations, the definition of a solid hazardous waste under the RCRA is broad. The RCRA uses a detailed set of rules in order to determine if a waste is regulated under its provisions. This set of rules can be divided into four steps:

(1) Determine if the wastes meet the RCRA definition of a solid waste.

(2) Determine if the solid waste is a listed hazardous waste.

(3) Determine if the solid waste is a characteristic hazardous waste.

(4) Check to see if the solid waste is exempt from regulation.

In step one we determine if the waste is a solid waste as defined in the RCRA. The definition of a solid waste does not simply involve the determination of the phase of the material. Solids, liquids and gases are all included in the definition of an RCRA solid waste as defined in 40 C.F.R. § 261.2. We should also note that even materials which are intended to be reused or recycled can be regulated under the RCRA thus, we must be sure to check all materials leaving the process to ensure that we are not inadvertently excluding materials from consideration. If a waste does not meet the definition of a solid waste, it is exempt from RCRA regulation.

Once we have determined that a material is a solid waste a defined by the RCRA, we must then determine if the waste is a hazardous waste. There are two ways for a material to be considered hazardous, either the waste is a listed hazardous waste or the waste is a characteristic hazard. We begin by determining if the solid waste is a listed hazardous waste. Unfortunately, this is not a simple as looking up a compound on a list. The lists included within the RCRA require the identification of certain hazardous compounds and the source the waste. The solid hazardous waste list is more a list of rules rather than a list of specific waste materials. For example, solid wastes meeting the definition in the following rule are considered a solid hazardous waste under the RCRA and are designated with the EPA hazardous waste number F024: *Process wastes, including but not limited to, distillation residues, heavy ends, tars, and reactor clean-out wastes, from the production of certain chlorinated aliphatic hydrocarbons by free radical catalyzed processes. These chlorinated aliphatic hydrocarbons are those having carbon chain lengths ranging from one to and including five, with varying amounts and positions of chlorine substitution. (This listing does not include wastewaters, wastewater treatment sludges, spent catalysts, and wastes listed in §261.31 or §261.32.).*

If a waste is not a listed waste we must then check to determine of the solid waste is a characteristic hazardous waste. A characteristic hazardous waste is a waste that possesses at least one of the four following characteristics: Ignitable, Corrosive, Reactive and Toxic. Each of these characteristics has a specific definition and the RCRA also set forth testing procedures for determining the characteristics of the waste. The specific definitions can be found in 40 C.F.R. §261.21 - §261.24. If a waste has not been identified as listed or characteristic, then it is exempt from regulation under RCRA. If a waste has been determined a solid hazardous waste as outlined in the first three steps above, we can still be exempt from regulation if an exemption exists for the waste. For reasons undoubtedly motivated by political wrangling rather than by scientific assessment, some industries and waste categories have received an exemption from regulation, even though the wastes do meet the RCRA definition.

From the above description, it should be clear that the format of the RCRA lends itself to a rule-based implementation within the MAT. The two keys to the implementation of the RCRA into a computer-based tool are (i) access to the required data upon which the rules would act and (ii) the expression of the rules within a computer system. Access to data can be established through relational databases of the properties required by the rules. Of course, not all the required data will be available and some of the data is empirical, thus the final system will require interaction with the user. Through the categorization of the rules by industry and process operation, a system could be developed which applies the appropriate set of rules to assess the materials of the process. Illustration 7.3 gives an example of the type of rules constructed for the prototype MAT system. In the prototype system we have focused on the batch processing industry and operations which would be used in those industries.

> *Illustration 7.3*
>
> In this illustration we will provide the pseudo code for the rules used to represent an F024 waste. The text of the RCRA rule was given earlier in this section.
>
> The main rule is given as follows,
>
> if Material (Designation = Process Waste) and Reaction (Type = Free Radical Catalyzed) and Product Material (Structural Type = Chlorinated Aliphatic Hydrocarbon) then Material (Hazard Designation = RCRA Hazardous).
>
> From the above rule, we see that the rule requires information about the Material, the Process, the Reaction and the Product Material. We can use a combination of rules an procedures with access to databases to retrieve the required information. For example, we can develop a rule with access to the process flowsheet to determine if a material is a process waste.

if Material(Destination = Waste) then Material (Designation = Process Waste)

It should be clear that a procedure could be written that accesses the material property database to get the molecular formula for the process material and determine easily if it is a chlorinated aliphatic hydrocarbon. The only information that can not be readily assessed from the available process information is whether the reaction type for the production of the process is a Free Radical Catalyzed reaction. In this case we would need to prompt the developer to provide this information.

The Toxic Substances Control Act

The *Toxic Substances Control Act* (TSCA), 15 U.S.C. § 2601, was enacted to regulate substances before they create an environmental hazard. [11] It authorizes the EPA to regulate any substance which poses an "unreasonable risk of injury" to health or the environment. The TSCA allows the EPA to regulate chemical usage and manufacture by requiring users and manufacturers of chemicals to report any "significant new use" of chemicals listed in the TSCA Chemical Inventory and all new chemicals (except those in exempted categories, for example drugs and pesticides). It requires chemical manufacturers to test new substances and provide health and safety data for the substances in the form of a PreManufacture Notification (PMN). The PMN allows the EPA to review the data and regulate any substance that poses an unreasonable risk of injury to health or the environment. The TSCA gives the EPA broad authority to regulate substances that are deemed to create an unreasonable risk. The range of options extends from banning the use or manufacture of the substance, to regulating methods of manufacture, use and disposal, to imposing quality control requirements.

To ensure compliance with the TSCA, the substance producer or user must screen the substance against the TSCA Chemical Inventory list. If the substance does not appear on the list and the producer is not exempt, the producer must file a PMN with the EPA. There are not only exemptions for categories of substances but also for small quantity producers. If the substance does appear on the TSCA Chemical Inventory list, the producer must determine if the substance is regulated by searching the regulated substances list. Finally, if the substance is not regulated, the producer or user may proceed with the production or use of the substance. We should note, however, that because a substance is not regulated under the TSCA, it does not remove the burden of determining if the substance is regulated under other federal regulations.

With respect to the development of the MAT, we see that the TSCA is basically a list search with a few rules to determine exemptions based on substance type and volume of material used or produced. The MAT can be valuable for determining whether the process will require action under the TSCA. The MAT can simply check the materials used in the process against the 70,000 substances in the inventory and report whether or not the substance is listed. If listed the MAT can search to determine if the substance is regulated or exempt. If not, the MAT could indicate that a PMN is required.

The Comprehensive Environmental Response, Compensation and Liability Act and The Superfund Amendments and Reauthorization Act

The *Comprehensive Environmental Response, Compensation and Liability Act* (CERCLA), 42 U.S.C. § 9601, and The *Superfund Amendments and Reauthorization Act*, 42 U.S.C. §11001, are regulations enacted to regulate releases of hazardous chemicals to the environment and the planning for possible releases. CERCLA regulations require the reporting of a release or a threatened release of specific quantities of hazardous materials. SARA extends CERCLA and requires emergency planning and inventory reporting when specified amounts of hazardous materials are located within a facility. SARA is sometimes referred to as the Emergency Planning and Community Right-to-Know Act. CERCLA and SARA both contain lists of substances and corresponding threshold planning quantities above which a facility is subject to CERCLA and SARA regulations. CERCLA and SARA include materials which are deemed hazardous in the RCRA.

The MAT can assist in the inventory reporting and emergency planning by screening the chemicals used in a process to determine if they are regulated under CERCLA and SARA. The information collected can be used to determine if the process will cause the facility to exceed the threshold planning quantities for listed materials and will trigger planning requirements.

The Occupational Safety and Health Act – Hazard Communication Standard

The *Occupational Safety and Health Act* (OSHA), 29 C.F.R. § 1910 – *Hazard Communication Standard* (HCS) was enacted in order to ensure uniform workplace health and safety standards. The HCS requires manufacturers and importers to evaluate the chemicals that they produce, to determine whether the chemicals are hazardous and to provide employees with information about such hazardous chemicals. It also requires employers to provide information about the safe handling procedures for hazardous chemicals. Under the HCS chemicals are either a *physical hazard* or *health hazard* or both. Drugs and hazardous wastes regulated by the EPA are exempt. The HCS establishes a number or rules for the determination of a hazardous chemical. The HCS rules use both empirical and physical property data to perform the hazard determination. For example, the following rule is used to determine if a chemical is pyrophoric thus, hazardous: if the chemical ignites spontaneously in air at a temperature ≤ 54.4 C then it is a physical hazard. An example of a health hazard is a chemical which is highly toxic. A highly toxic chemical is defined as a chemical that meets on of the following Lethal Dose or Lethal Concentration tests.

(a) LD_{50}(oral) ≤ 50 mg/(kg body weight) of albino rats of weight 200 – 300g.

(b) LD_{50}(dermal) ≤ 200 mg/(kg body weight) of bare skin albino rabbits of weight 2 – 3kg.

(c) LC_{50}(inhalation) ≤ 200ppm by volume of gas or vapor of albino rats of weight 200 – 300g when administered by continuous inhalation for ≤ 1 hr.

(d) LC_{50}(inhalation) ≤ 2 mg/L of mist, fume or dust of albino rats of weight 200 – 300g when administered by continuous inhalation for ≤ 1 hr.

In addition to the hazard determination using the rules contained within the HCS, the HCS also has lists of compounds considered hazardous. The HCS also recognizes lists of chemicals by other organizations as hazardous. These lists include the latest edition of the American Conference of Governmental Industrial Hygienists publication *Threshold Limit Values for Chemical Substances and Physical Agents in the Work Environment*, the latest edition of the *Annual Report on Carcinogens of the National Toxicology Program* and the latest edition of the *Monographs* published by the

International Agency for Research on Cancer. Finally, the HCS also contains rules for determining whether a mixture is a hazard.

The implementation of the OSHA – HCS within the MAT is similar to the previous regulations. It requires a combination of database search for health and safety data and physical property data for the compounds to be assessed. The information from the database searches feeds the rules derived from the regulations. Additionally, the MAT need to have simple list search capabilities and access to databases containing the lists specified above. Finally, if the data required to assess a compound is unavailable the MAT may suggest the user perform tests of the compound in order to obtain the required data.

In the preceding subsections we have discussed the seven major federal health, safety and environmental regulations. It should be obvious from regulation summaries above that the regulation based assessment of process chemicals and mixtures is a complex task. We have tried to incorporate the portions of the regulations relevant to the batch processing industry into the prototype of the MAT. By assessing a process we can determine if the process is going to require the facility to comply with any additional regulations. Ideally, by performing an assessment on the process, the process developer will be able to get a feel for the regulator implications of the various chemicals used in the process. The developer can then go back an work with the chemist to rework the process to remove the most regulatory challenged chemicals. In the following two sections we will provide an overview of how the MAT performs the assessment of pure chemicals and the assessment of mixtures.

Assessment of Pure Chemicals

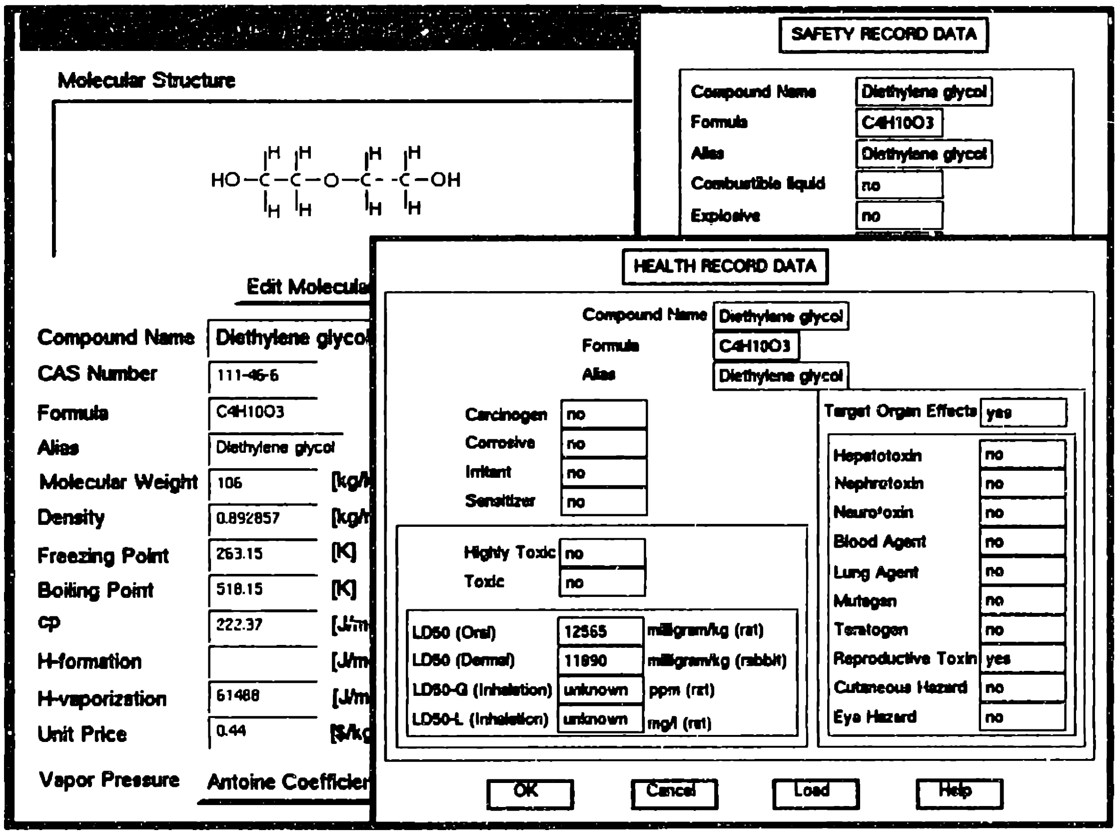
The foundation of compound assessment is rooted in three governing principles: (i) screening of dedicated regulatory lists, (ii) categorized hazardous compounds and (iii) hazardous physical properties. Specific regulations like the SARA 313 Toxic Chemical List, explicitly list offending compounds. A simple linear search maps the compounds with the respective regulations. Categorized hazardous compounds, on the other hand, can be identified based on their membership to a specific chemical group. For the analysis of the existence of such offensive molecular structures within a compound, BDK makes use of the *Molecular_Editor,* which allows the generation of graphical representations of compounds. Graphical interpretation of the molecular structure in terms of its building blocks can be exploited for the purpose of assembling the group contributions for physical property estimation techniques and also allows the extraction of structural information of chemical groups and their bond-specific connectivity.

> *Illustration 7.4*
>
> Figure 7.25 illustrates how diethylene glycol can be identified as hazardous according to the SARA 313 Toxic Chemical Categories, because it contains the functional group glycol ether. Analysis of the molecular structure begins with a search of the groups used to construct the molecule in the *Molecular_Editor*, to determine if the group matches any listed hazardous functional groups. In this example, the search is performed based on the -O-CH2-CH2- group. The number of these groups contained in the structure is counted. The next step lies in analyzing the connections of these groups e.g. if the α-oxygen of group1 is connected to the δ-carbon of group2 and there exist two or more instances of these groups then the compound is designated as an ether. Next, the structure is searched again to see if the δ-carbon of group2 is connected to a hydroxyl group. This being true would indicate a glycol ether. Finally, the α-oxygen of group1 is checked whether it is connected to a alkyl group, an aryl group or a hydrogen atom. Thus, we have found the compound to be a glycol ether and under SARA 313 the compound is designated as hazardous.

The final means by which compounds are assessed is based on their physical properties. These properties include properties such as phase at standard conditions, as well as empirical properties such as flashpoint and lethal dose concentration. These properties are linked into the BDK environment through the database manager of the material model, which supplies the system with data from external databases.

Since BDK will be used to develop processes containing new compounds for which many of these properties may not be known, the user can enter estimates of these properties to complete the assessment or rely on property group-based estimation techniques as eluded to in the previous paragraph. The assessment is performed based on rules extracted from the US Federal Regulations. The rule contained in Illustration 7.5 is based on OSHA's Hazard Communication Standard and represents a typical example of the knowledge involved in compound assessment.

> *Illustration 7.5*
>
> Rule: if the LD50 dermal concentration of compound X is less than or equal to 200 mg/kg body weight then compound X is "highly toxic" under OSHA's HCS. The relevant compound information for activating this rule is retrieved from the health record, as depicted for diethylene gycol in Figure 7.25.



Compound Name	Diethylene glyco	
CAS Number	111-46-6	
Formula	C4H10O3	
Alias	Diethylene glycol	
Molecular Weight	106	[kg/
Density	0.892857	[kg/
Freezing Point	263.15	[K]
Boiling Point	518.15	[K]
cp	222.37	[J/m
H-formation		[J/m
H-vaporization	61488	[J/m
Unit Price	0.44	[$/kg
Vapor Pressure	Antoine Coefficie	

SAFETY RECORD DATA

Compound Name	Diethylene glycol
Formula	C4H10O3
Alias	Diethylene glycol
Combustible liquid	no
Explosive	no

HEALTH RECORD DATA

Compound Name	Diethylene glycol
Formula	C4H10O3
Alias	Diethylene glycol
Carcinogen	no
Corrosive	no
Irritant	no
Sensitizer	no
Highly Toxic	no
Toxic	no

LD50 (Oral)	12565	milligram/kg (rat)
LD50 (Dermal)	11890	milligram/kg (rabbit)
LD50-G (Inhalation)	unknown	ppm (rat)
LD50-L (Inhalation)	unknown	mg/l (rat)

Target Organ Effects	yes
Hepatotoxin	no
Nephrotoxin	no
Neurotoxin	no
Blood Agent	no
Lung Agent	no
Mutagen	no
Teratogen	no
Reproductive Toxin	yes
Cutaneous Hazard	no
Eye Hazard	no

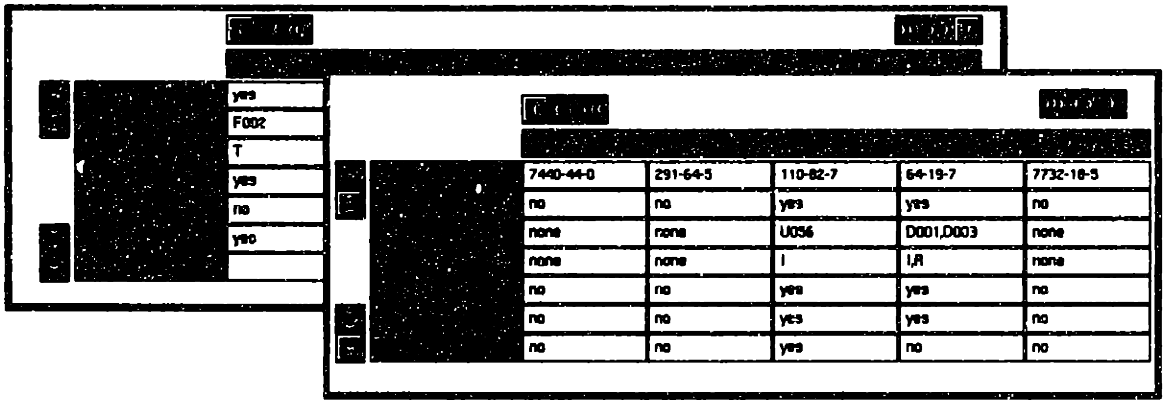
Figure 7.25 The Property Records of Diethylene Glycol

Assessment of Process Mixtures

The assessment of process mixtures builds on the qualities of its component compounds. Each regulation contains different criteria by which a mixture is considered hazardous. Generally, the assessment of mixtures is based on the concentrations of hazardous compounds contained in the mixture. If the concentration of a hazardous component or the sum of the hazardous components exceeds a given threshold, the mixture is considered hazardous. A second criterion for assessing a mixture resembles the case for the pure chemicals, using characteristic properties for the mixture. The four characteristic hazards are *ignitability*, *corrosivity*, *reactivity* and *toxicity*. If a mixture is found to show any of the four characteristics it is considered hazardous.

The results of the assessment are displayed to the user in tabular format as depicted in Figure 7.26. The name of the regulation is displayed in the row headers while the stream names are given as column headers (Figure 7.26, behind). The stream names are those corresponding to those given on the flowsheet. The elements of the table values are produced by the assessment for each stream. By clicking on the stream name another table is displayed listing the compounds contained in that stream and their assessment values (Figure 7.26, overlapped).

The conclusions of the regulation based materials assessment are summarized in a regulation report. It contains two sections, which list the results from the assessment of compounds and for mixtures, respectively. The tables include the name of the hazardous compounds, the CAS-number, e.g. 111-46-6 Diethylene glycol, followed by the citation of the violated regulations e.g. SARA 313 and specification of the reason for the offense e.g. compound contains a SARA 313 Toxic Chemical Category group. A similar summary is produced for the process mixtures. In a third section, all relevant environmental, health and safety data are listed for the respective compounds to further document the process performance in the respective disciplines.



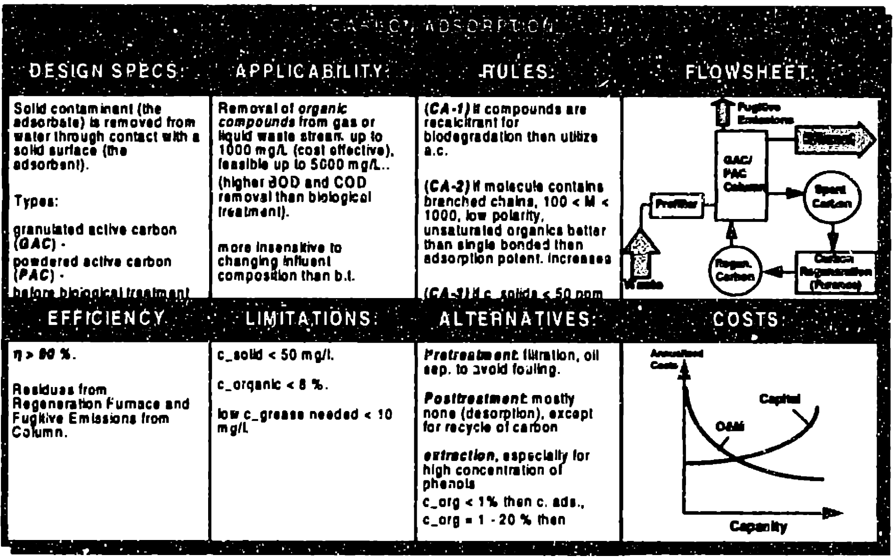
yes
F002
T
yes
no
yes

7440-44-0	291-64-5	110-82-7	64-19-7	7732-18-5
no	no	yes	yes	no
none	none	U056	D001,D003	none
none	none	I	I,R	none
no	no	yes	yes	no
no	no	yes	yes	no
no	no	yes	no	no

Figure 7.26 Material Mixture (behind) and Compound Assessment Table (overlapped)

The Waste Treatment Selection

The waste treatment selection is performed using a tool called the *Treatment_Selector* and was developed and constructed by Andreas A. Linninger [12]. The following description is from our joint paper [12]. The objective of the *Treatment_Selector* is to identify the final fate for all waste streams produced by the conceptual process design and to include the associated treatment cost for the offending substances into the quality validation of different process alternatives. In the BDK environment, all final process streams with the exception of a single product stream, are considered wastes and enter consideration for waste treatment. Matching generated wastes streams with state-of-the-art treatment technologies requires a broad knowledge base. For finding the most appropriate treatment option, information from specialized literature [13, 14], professional publications [15, 16] and industrial vendors has been condensed into a database of waste treatment technologies. This database called the *Treatment_Map*, possesses an entry for each technology, which is itself a frame-based object of type *Treatment_Object* [17]. Its key attributes Rules, Limitations and Efficiency, as they are displayed for the Carbon Adsorption based techniques in Figure 7.27, are used to determine the applicability and the efficiency for a certain treatment option in the three-step reasoning process explained in the next section.



CARBON ADSORPTION

DESIGN SPECS:	APPLICABILITY:	RULES:	FLOWSHEET:
Solid contaminent (the adsorbate) is removed from water through contact with a solid surface (the adsorbent). Types: granulated active carbon (*GAC*) - powdered active carbon (*PAC*) - before biological treatment	Removal of *organic compounds* from gas or liquid waste stream up to 1000 mg/L (cost effective), feasible up to 5000 mg/L.. (higher BOD and COD removal than biological treatment). more insensitive to changing influent composition than b.t.	(***CA-1***) If compounds are recalcitrant for biodegradation then utilize a.c. (***CA-2***) If molecule contains branched chains, 100 < M < 1000, low polarity, unsaturated organics better than single bonded then adsorption potent. increases (***CA-3***) If c_solids < 50 ppm	
EFFICIENCY:	**LIMITATIONS:**	**ALTERNATIVES:**	**COSTS:**
η > 90 %. Residues from Regeneration Furnace and Fugitive Emissions from Column.	c_solid < 50 mg/l. c_organic < 8 %. low c_grease needed < 10 mg/L	***Pretreatment*** filtration, oil sep. to avoid fouling. ***Posttreatment*** mostly none (desorption), except for recycle of carbon ***extraction***, especially for high concentration of phenols c_org < 1% then c. ads., c_org = 1 - 20 % then	

Figure 7.27 Partial view of the Carbon Adsorption *Treatment_Object* of the *Treatment_Map* database[12].

In the *Diagnosis* step, the expert system emulates the screening process as devised by a human expert categorizing the properties of each waste streams in terms of (i) high-level generalizations, i.e. *physico-chemical interpretation*, and (ii) the generation of treatment tasks, i.e. *identification of treatment tasks*, based on the analysis of the offensive character of a mixture, such as toxicity, heavy metal contamination or presence of biocidic or odorous compounds.

The *physico-chemical interpretation* of process mixtures, their chemical substances and their atomic structure, allows for the specification of each waste stream according to the global categories solid/liquid waste or air emission, as well as their type, i.e. inorganic or organic, and their membership to a chemical group such as alkane, alkine, armoate, acid, aldehyde, etc. This interpretative view on groups of chemicals, which share common behavior and properties is essential for the assessment of their treatability. Earlier approaches deduced applicable treatment options from the compounds itself, where a treatment option was simply allocated based on the pair of *compound-i treatable-through treatment-j* stored for each compound, e.g. benzene treatable-through distillation. Such a strategy, however, does not recognize the context-specific nature of the treatment selection problem, where the state of a mixture, i.e.

concentration of the respective chemicals or group of chemicals as well as temperature of a mixture are relevant for its applicability. The result of the diagnosis is summarized through a statement in natural language, which mimics the language utilized by chemists and environmental engineers to describe the nature of a problematic waste, e.g. *95% aqueous liquid waste stream, recalcitrant due to 1000 ppm heavy metals, 4 % organics, otherwise biodegradable.*

Once the waste stream has been categorized and its possible offensive compounds recognized, *identification of the treatment tasks* applies knowledge about environmental regulations and consideration of the health, safety and environmental impact of the mixture, as it was described in the previous section, to generate goals, which specify the target for recovery of valuable compounds or destruction/separation of offensive substances. These goals like *Recycle(compound-i), Destroy(toxic(compound-j), Destroy(biocide(compound-k)* are implemented as global facts in working memory. According to the priorities of the waste treatment policy, these goals are ranked to generate a hierarchy of differently weighed treatment objectives. In the current version, recycle goals are evaluated highest, followed by destruction of toxic compounds, separation/destruction of biocides, odorous compounds and elimination of organic compounds.

In the second step, the *Treatment_Selector* adopts the goals identified by the Diagnosis and tries to identify feasible treatment options to satisfy these goals. This search is driven by *Connecting Rules*, which guide the reasoning process in the direction of the most probable candidate technologies, e.g. chemical treatment for destruction of toxic compounds. The actual decision as to whether a treatment is applicable is performed through *Agents*, responsible for a single treatment option, e.g. Agent for Wet Air Oxidation. At first, the agent checks the limitations of a treatment with respect to the mixture. If no limitation, which would prohibit the applicability of the technological concept, is found e.g. no heavy metal contamination for biological treatment, the applicability rules are screened for their ability to meet one of the currently unsatisfied goals, e.g. *Destroy (biocide(compound-i))*, etc. After completion of the agent's work, its postconditions state the impossibility of a certain treatment, e.g. *Not_WAO_ Treatable(stream-i)*, or its applicability plus the performable task, *WAO_Treatable(Stream-j) & Can-Destroy(toxic(compound-i)) & Can-Destroy(biocide(compound-j)).*

Finally, at the *Recommendation* level, the candidate treatments are compared against each other in order to identify the best treatment step. The most appropriate option is found based on the following criteria: (i) capability of meeting more than one goals (ii) technical efficiency based on the destructive potential or separation efficiency as stored in the efficiency slot of the *Treatment_Objects* and (iii) cost considerations. Once the final treatment is selected, possible alternatives treatment options are stored for later user's access and reference. The selected technology is applied to the mixture under consideration and the composition of the treated mixture, i.e. the residue and the destroyed or separated fraction are evaluated. The residue, which has lost at least one offensive quality through the satisfaction of at least one goal, is assessed again as it is the case for all any other waste streams. The *Treatment_Selector* allocates additional treatment steps until no offending character can be identified. The inoffensive final treated wastes can be discharged into the surface water, for treated waste water, deposited in a landfill, for inoffensive solid residues, or released to the atmosphere for cleaned off-gases (air-emissions).

Figure 7.28 illustrates the abstract representation of the results obtained by the *Treatment_Selector*. The operations for the production of the product streams are symbolically represented by a block entitled production, while for each waste stream an array of necessary treatment options is displayed. Each "best" treatment step is indicated by an instance of a treatment icon, which allows access to the respective *Treatment_Object* in the *Treatment_Map* database. Clicking these icons allows the user to check on the technological information of the treatment option and the criteria utilized by the expert system. The results of this reasoning activity for this waste stream can also be printed out on an external file.

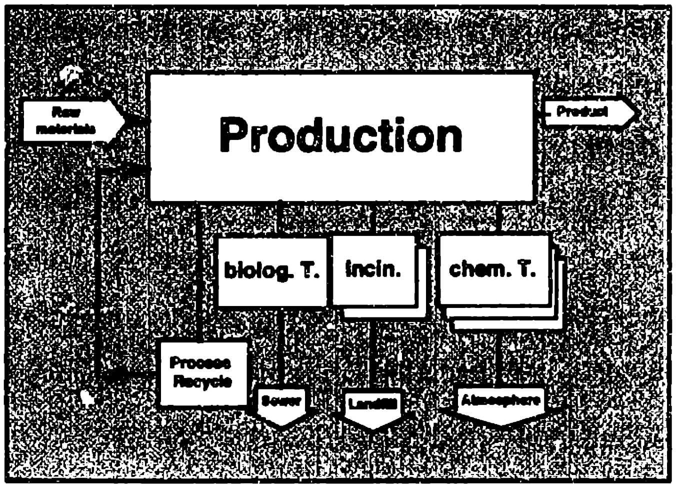


Figure 7.28 The Waste Treatment Flowsheet [12].

Possible treatment alternatives can be viewed through selection of the rectangles located behind the principal solution. Bringing forward and clicking the respective icon similarly allows one to access the data of their associated *Treatment_Objects*. The final fate of each waste stream is indicated though an arrow as depicted in Figure 7.28, where each final destination contains the associated ultimate residue which represents the final environmental impact of the production including waste treatment.

7.2.4.5 BDK Report Generation

A feature of BDK that at first glance may seem trivial, but is central to the usefulness of BDK is its report generation and information presentation facilities. A great deal of thought went into developing the method for displaying the information that BDK generates. The information must be in a form useful to the developer as well as to those who will be reviewing this information and making decisions about the process, such as Can the process be implemented? or Should the process be approved?

Currently there is a big push on information technology within the pharmaceutical industry. One of the biggest dilemmas is how to condense and display large amounts of information in a manner such that it is useful to the intended audience. We have also responded to this push by first listening to the people for whom the system was being designed and taking their needs and desires into account when creating the report generation facility. As can be expected, when asking many different companies and different groups within each of these companies what they need and how they would like to see the output from such a system, we received many different suggestions. From these we extracted a number of common features. While we realize that no one solution will be acceptable to all users, we have developed the current format around the chemists and process developers. A key concept here is the idea of standardization. To ensure efficient and effective communication between different groups it is important to develop a standard platform for communication. The reports that are generated by BDK are useful to both chemists and process developers. The features (not all of which have been implemented) that are included in a report are 1) a list of chemical reactions, 2) a list of compounds used in the process, 3) the process flowsheet, 4) the batch sheet, 5) the process stream tables, 6) waste stream tables, 7) utilities required, 8) an economic analysis, 9) a waste treatment analysis, 10) a materials assessment based on the appropriate federal regulations, 11) the equipment required, 12) a Gantt chart for the operations and 13) a process sequence diagram.

Another important aspect of the standardization is that it allows people across the company to communicate better. Also, since all the information is centralized it becomes trivial to generated alternate views of the information depending upon the intended audience. For example, the format for electronic FDA filing for a process could be added as an alternative view for the information generated by BDK (not implemented). This would greatly reduce the time needed to reproduce the information into the required FDA format. A feature that has not yet been added to the report is the decision tracing for the *Process_Synthesizer.*

7.2.5 Process Synthesis

As we mentioned previously, in order for BDK to fully capture all aspects of process development it must have the capability to assist the developer with process synthesis. The process synthesis facility of BDK uses the MEA-NMP methodology in order to assist the developer in the development of a base case design. The framework for the MEA-NMP approach has been implemented and integrated with BDK in a conceptual design tool is called the *Process_Synthesizer*. The *Process_Synthesizer* is intended to assist the process developer with the conceptual design process and helps the user determine the feasibility of operations to perform specific tasks as described in Chapter 4.

The *Process_Synthesizer* is an interactive system based on both qualitative and quantitative analysis. It exploits the features of BDK in order to assist the developer with the process synthesis task. Fundamental to the implementation of the MEA-NMP methodology is the operations-based language. It is BDK's use of the operations-based language that allows the *Process_Synthesizer* to be layered upon BDK i.e. they use the same fundamental language for describing a process.

Behind the operations-based language lie the operation and material models. The procedure we propose for process synthesis using the MEA-NMP methodology requires that we be able to apply the operation models and generate the postconditions for the selected tasks. In essence we must simulate the behavior of the operations. BDK's operations allow us to model the behavior of the materials after the operation has been applied and at the selected operating conditions, thus generating the postconditions.

Before we can even apply the operations we must check the current material state against the preconditions of the operation we wish to perform. The material model contains all the information about the state of the material and makes the evaluation of the preconditions possible. We have used the BDK operation models as the core for the *Process_Synthesizer* operations and linked them to the appropriate set of preconditions for each operation.

Process synthesis is a data intensive task. We have been able to support the process synthesis task with the physical property databases that are contained within BDK. The material model, operation models and the white knight identification procedures all use this data to model the system and construct the base case design. Additionally, the preliminary assessment facilities within BDK allow the developer to evaluate the process base on raw material and utility costs. These preliminary evaluations make use of the material and utility cost databases.

We see from the above discussion that the process synthesis task takes full advantage of all the support facilities contained within BDK. Now that we have shown that BDK does provide the facilities required in order to perform the process synthesis task, let us now discuss the functionality of the *Process_Synthesizer*. We begin the description of the *Process_Synthesizer* with a discussion of the important elements.

First and most important is that the system is designed to act as an assistant to the human developer, not a replacement. While the system does contain knowledge to make design decisions, it provides alternatives to the user so that the user can make the final decision. At any point in the design process the user can override the decisions made by the system with his own. The usefulness of an assistant is to help the expert to work more efficiently. As an assistant, the system attempts to perform the tedious tasks freeing up the user to employ his expertise. For example, the system automatically creates the batch sheet and flowsheet simultaneously from information provided by the expert and from databases. The tedious tasks of typing the batch sheet and drawing the flowsheet are done by the computer. The system can also perform simulations of the operations providing the user with material and energy balances along with the various states of the system. This allows the developer make better decisions without having to perform all the tedious simulation calculations independently. During the synthesis, the system provides lists of feasible operations for each task. Allowing the developer to only consider feasible alternatives, perhaps, some of which the developer has not thought of. The system can also keep track of the many possible alternatives for the developer to make the comparison of alternatives easier. The more alternatives that can be screened, the better the design that can be produced. Producing good designs can be very important, not only for the approval process but also for staving off competition from possible generic producers once the patent expires.

Another key feature is the ability to track the decision making process. This is extremely important when trying to find problems in the design, providing justification to the FDA for the approval of the process and for future process modifications. In order to modify the process it is extremely important to understand what decisions were made and why.

Finally, we would like to have the system automate the optimization problem formulation. The screening of a fixed number of alternative operations for each task to be performed lends itself to an optimization problem as discussed in Chapter 6. We feel it is possible for the assistant to use the information provided by the human expert to automatically construct the superstructure for the problem to be optimized along with the constraints on the problem. The expert can then review the constraints and make modifications. The system can be integrated with optimization tools to perform the optimization with the given information. It should be clear from the above examples that the tools provided by the *Process_Synthesizer* would not only make the process developer more efficient but allow for the development of better designs. Also, since design process is documented, this information could be provided to the FDA to increase the rate of approval for the process.

The framework for the *Process_Synthesizer* has been implemented within BDK. The *Process_Synthesizer* makes use of the operational models within BDK and the *Augmented Planning Network (APN)* developed by Linninger [18]. The APN allows the computer implementation of a reasoning structure in order to simulate a series of actions. The APN also allows for the graphical construction of these reasoning structures within an object oriented environment. The details of the APN are given in Linninger [18] and a short description of the key features of the APN is given here.

7.2.5.1 The Augmented Planning Network

The APN is a network composed of a tree of nodes. There are two types of nodes depicting an *action* or a *condition*. An action node simply performs an action. For example, an action node can perform a calculation and return a value, retrieve information from a database or perform a fixed set of tasks depending on the input conditions. A conditional node performs an action and, based on the results of that action, makes a decision about the next course of action. For example, the conditional node can perform a calculation an depending upon the value returned, can make a decision about which branch of the decision tree to follow. The decision making tools usually consist of rules or sets of rules. The rules provide a major advantage in allowing the decision making structure to reason about the objects in the system. For example, usually during the process synthesis, separation decisions are made based on the phases of the materials present. The APN allows the reasoning over the attributes of the process materials in order to make decisions about the choice of separation operations such as: *if*

gas and liquid phase present then flash. The nodes also provide a convenient method for constructing the reasoning network through a series of connections. The action nodes may take any number of input connections but only produces a single output connection, while the conditional node may take any number of input connections and produces two output connections. The conditional node allows branching in the network with one branch being the *True* branch and the other being the *False* branch, thus the results of a conditional operation must always return either a true or a false result. The combination of conditional nodes and action nodes is sufficient to develop many sophisticated networks.

The APN also allows the development of sub-networks. Thus, a node must not only have a procedure or a rule behind it, it can actually have another network underlying it and returning information to the parent node. This is crucial for the hierarchical decomposition of tasks into subtasks and greatly enhances the functionality of the APN. It also greatly enhances the readability of the APN since the subnodes are displayed in a similar network as that of the parent so the designer can dive deep into the reasoning structure of the network and only examine the portion of the network which is of interest. Illustration 7.6 provides an example of an APN for the selection of waste treatment options based on the phase of the process material. Additional examples of the APN created for the process synthesis are given in the next section.

> *Illustration 7.6*
>
> Figure 7.29 depicts an example of a simple APN which guides the selection of the appropriate treatment option for a process material based on the phase of the material. In the figure, we see that selection of the treatment is derived from the parent node called *Select Treatment Option.* The actual APN which performs the screening of the phases and the selection of the treatment option is a sub-plan of the Select Treatment Option node. We see that the sub-plan consists of both action nodes and conditional nodes. Once the execution of the sub-plan is complete and we have determined the phase of the material and selected a treatment option, we return to the parent node and continue following the parent path in the decision tree.

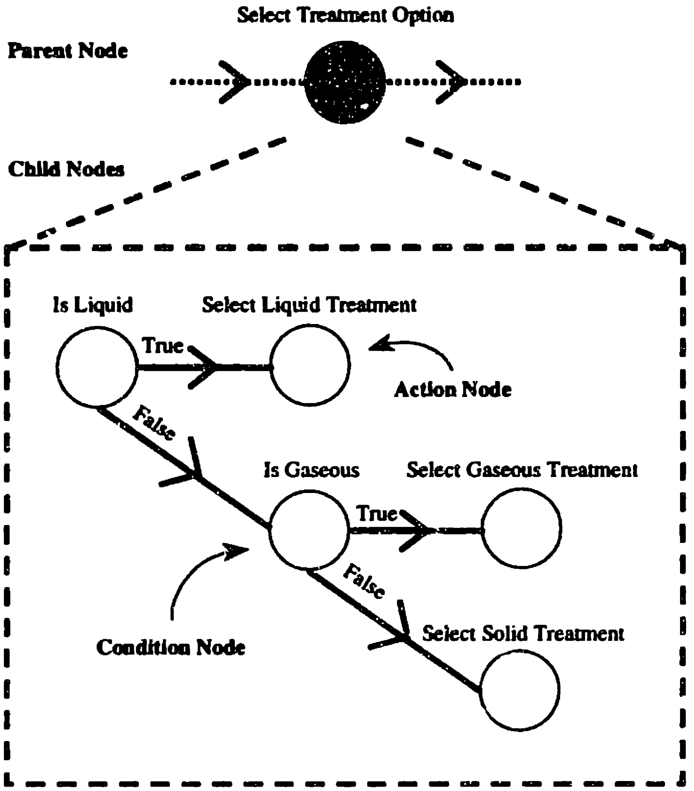


Figure 7.29 APN for Selecting a Treatment Based on Phase

Current Process_Synthesizer Implementation

The framework for the *Process_Synthesizer* has been implemented as a layered module on top BDK. This implies that the *Process_Synthesizer* makes use of elements of BDK but BDK remains a standalone application. In its current form the *Process_Synthesizer* should only be considered a prototype program in order to demonstrate the feasibility of a process synthesis system. The framework for all of the components is in place to perform the process synthesis activity but there are some areas that need further development in order to carry out the process synthesis activity from beginning to end. Currently the *Process_Synthesizer* allows the developer to begin the process by defining the product specifications, select a reaction set, charge the process materials, screen for raw material preparation tasks and perform the reactions. Also, the program identifies concentration differences and generates a separation structure for the purification of the product from a single phase. The precondition checking and white knight identification methods are operational and function within the separation system generation scheme.

The biggest obstacle to the automated process synthesis is the lack of phase composition prediction capabilities. Without having an accurate prediction of the composition and phases that will be formed at different operating conditions we cannot develop effective processing schemes. This is especially true during the separation system synthesis. Without the modeling capabilities to predict the proper phases that are formed after an operation is performed we are reduced to using perfect split assumptions or asking the user for the composition and phases that are formed.

In addition to phase composition prediction capabilities, a number of the components with the *Process_Synthesizer* need to be expanded. First, the reaction system synthesis module needs to be expanded and refined. Currently the developer can define the product specifications, define reaction sets from the stage reactions and perform the charge operation to introduce compounds to the process. All of the interaction with the developer is in the form of predefined dialogs, which the developer uses to enter the required information. The reaction synthesis system prompts the developer to introduce solvents based on the phase of the raw materials, determines if all the reactants are present and models the reactions. It also screens to determine if the product has been formed, if so it moves on to the separation system synthesis module, if not it prompts the user to define an additional reaction set in order to produce the product. The APN for the current reaction system synthesis module is given in Figure 7.30. The gray nodes in the APN indicate that there is a sub-APN for that node. In Figure 7.30 we have two nodes that have substructures, the ScreenUsingPhase node and the GenerateSepSystem node. The ScreenUsingPhase node substructure APN is shown in Figure 7.31. This structure identifies common preprocessing operations for raw materials.

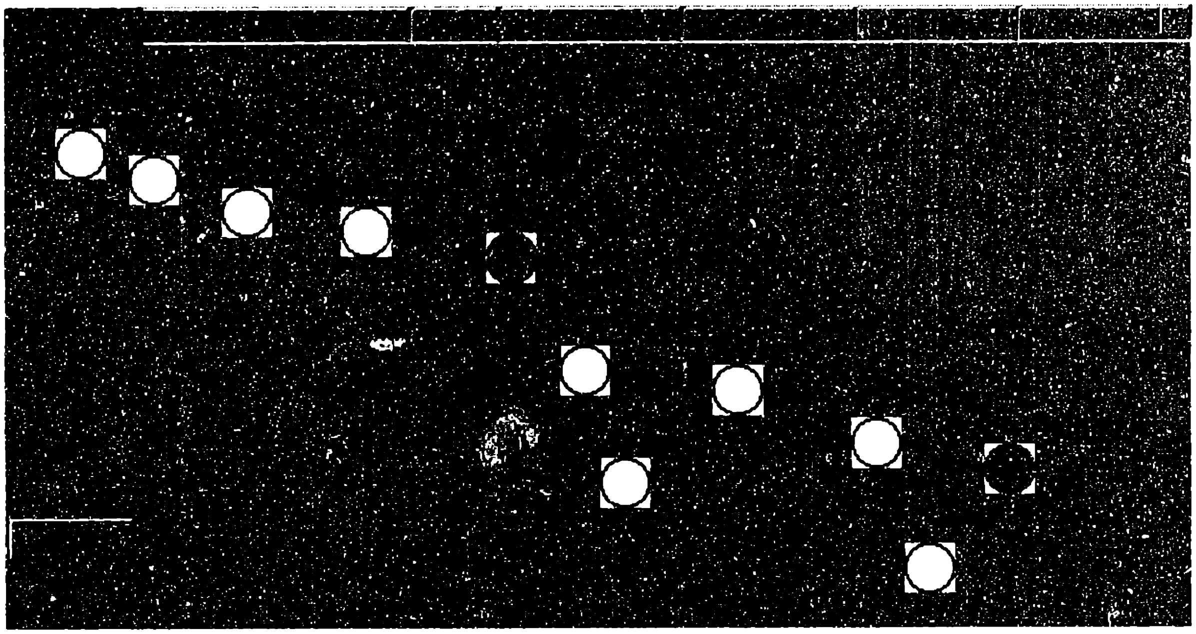


Figure 7.30 *Process_Synthesizer* Reaction System Synthesis APN

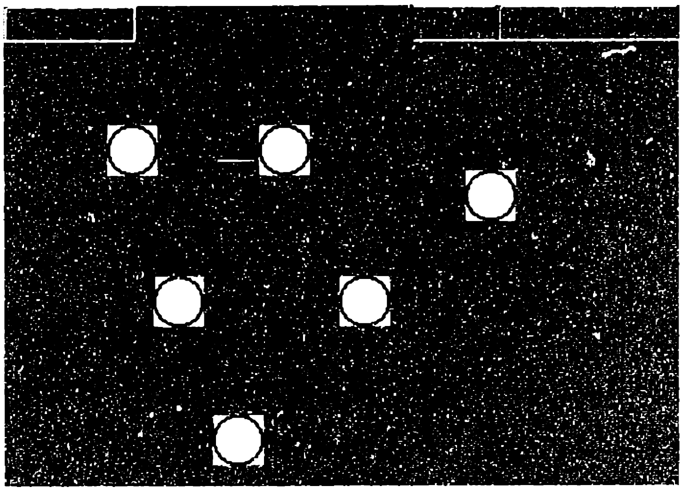


Figure 7.31 ScreenUsingPhase Sub-APN - Raw Material Preparation APN

The GenerateSepSystem substructure is shown in Figure 7.32. We see from the figure that the GenerateSepSystem substructure contains a network that first identifies the concentration difference, then if multiple phases are present it separates the phases, isolates the product from the dominate phase and finally checks to see if the concentration difference has been resolved, if not it loops over the network again. Figure 7.33 contains the APN used for isolating the product from a single phase. We see the basis for this process is the comparison of the desired phase of the product versus the current material phase. From this comparison it identifies the task category and primary alternative. Once the primary alternative is identified the precondition checking and white knight identification are initiated as described in Chapter 4. As the *Process_Synthesizer* applies operations it simultaneously commands BDK to generated the process flowsheet and batch sheet as shown in Figure 7.3.

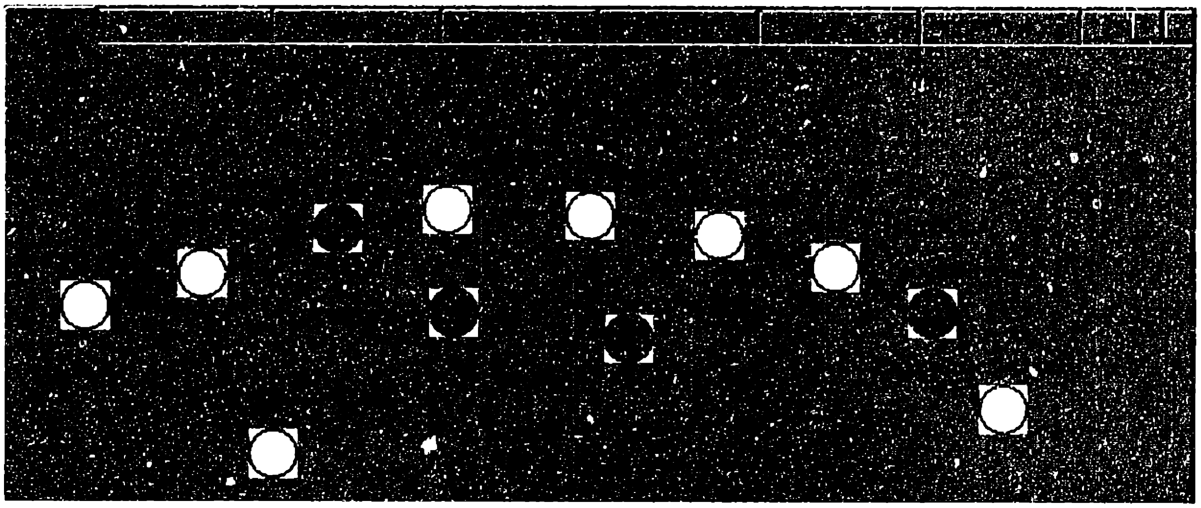


Figure 7.32 *Process_Synthesizer* Separation System Synthesis APN

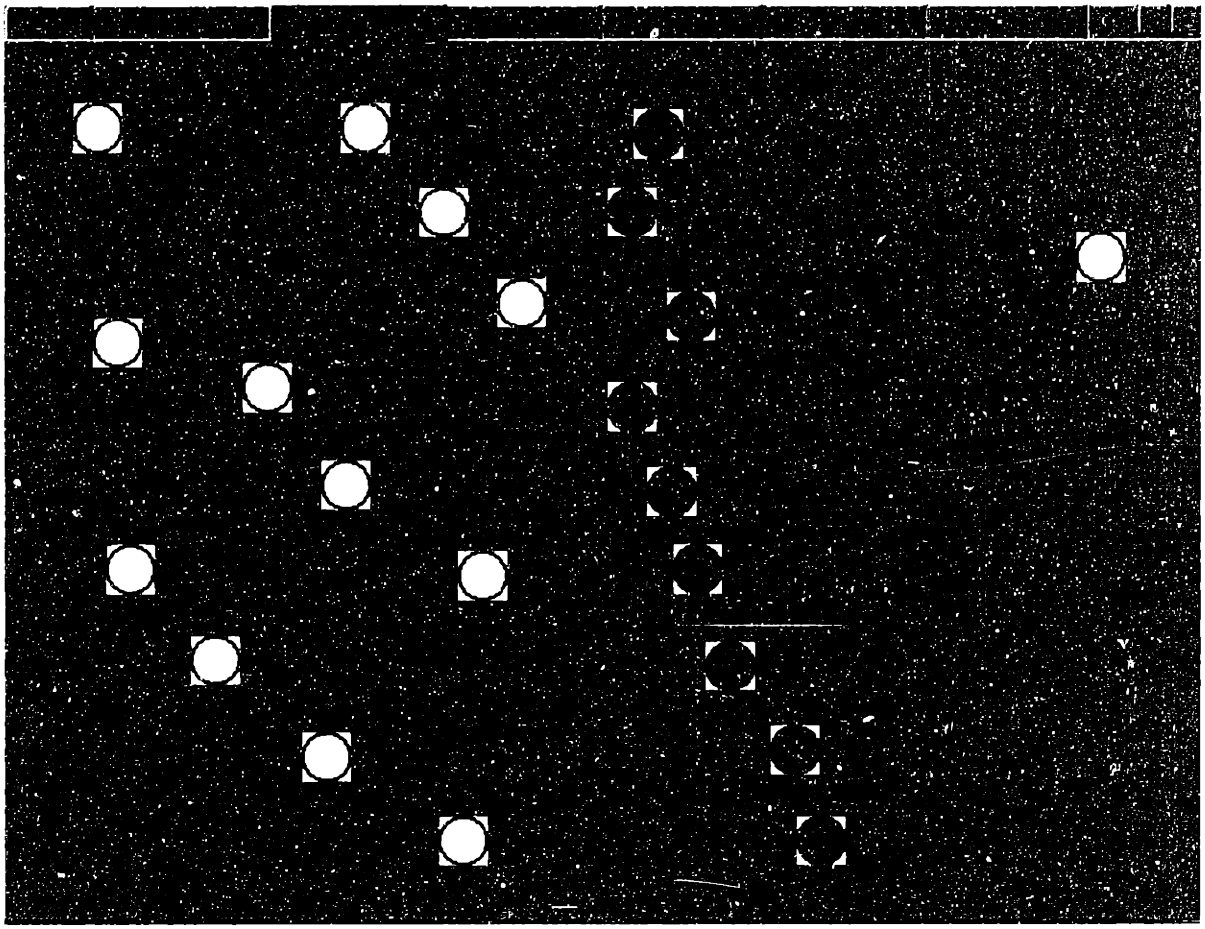


Figure 7.33 IsolateProduct Separation System Synthesis APN Subnode

The next portion of the system that needs to be expanded is the completion of the difference identification. Currently, the system only identifies the identity difference and the concentration difference. In order to complete the system we need to include the detection of the amount, phase, temperature and pressure and form differences. Along with expanding the difference detection, the separation synthesis system logic needs to be expanded and improved. In the prototype that has been developed, the logic can only handle the purification of the product from a single phase. If the product is distributed across multiple phases the system will isolate the product from the phase that contains the dominant amount of product.

The prototype for the *Process_Synthesizer* currently only allows the development of a single alternative through the construction of a base case design. Additional alternatives can be developed by running the system again and making different design decisions. We have not attempted to incorporate the superstructure construction scheme discussed in Chapter 6. This would require the completion of the task category tracking function to maintain the task categories that were selected at each node. Upon completion of the superstructure construction facility, we could then consider the implementation of optimization routines into the system for process optimization.

7.3 Implications for New Drug and Specialty Chemical Development

The above description shows BDK to be an appropriate and effective tool in the process development process but, what are the implications for new drug and specialty chemical development? And how can we measure its success?

The area of new drug discovery has experienced tremendous advancement in the science and technologies involved. Combinatorial chemistry, high throughput screening and many other processes have allowed for the development of many potential new compounds and drugs. While these have had a huge impact on the new drug development process, there is also a new focus on the process of process development. In order to remain competitive in today's business environment, companies are realizing that they must become more efficient in all areas of product development. It is this push toward greater efficiency while maintaining high quality standards that has led to the formation of the BatchDesign-Kit research project.

BDK with the implementation of the *Process_Synthesizer* is the first computer-aided system that attempts to capture the entire batch process development process in a single environment. The inclusion of process synthesis into the development process will have a large impact on the development of batch processes. For the first time we can systematically create processes and evaluate alternative design decisions for batch processes. Through the utilization of BDK's equipment allocation and process evaluation tools, developers with be able to evaluation a greater number of alternative designs in a shorter amount of time.

The implications for new drug and specialty chemical development are clear. BDK, if used as described above, has the potential to dramatically improve the quality of the processes being used within industry for the production of pharmaceuticals and specialty chemicals both from an environmental and a cost perspective. The iterative feedback loop between the process engineer and the chemist allows for better, cleaner processes to be developed. Acting as an assistant, relieving the process developers from the tedious tasks during batch process development, BDK allows them to be more productive by examining a greater number of alternatives as well as performing the development in a reduced time period. This, in turn, reduces the time to market for a given drug or chemical. The ability of BDK to keep track of the decision making process and various alternatives in a well documented and systematic manner implies that there should be fewer surprises along the development route. Any surprises that do occur can be quickly resolved through the use of the proposed decision tracking mechanism. The system should also make it easier to get approval for a drug under development. From the process perspective, the ability to submit the design as well as the decision making logic to the FDA for justification of the desired process should help the FDA to better understand the submitted process. The automatic report generation facilities should also reduce the amount of time putting the materials together for presentation.

The development of sound processes can also provide an additional advantage. Especially with complex processes, the development of strong processes can provide some protection from generic competition. If the process generated is more economical due to the production of less waste, production of the product in a more efficient manner or is more robust; then the difficulties in producing a more economical process may help to deter generic competitors or at least will allow the brand to compete based on price against the generics.

The above discussion shows that the main economic benefit from BDK will be in the reduction of time to market for the new drug or specialty chemical. Anything that can reduce the time to market can produce millions of dollars in extra revenue for the company. In new drug development we can generally reduce on the order of 10,000 possible candidates to one or two drugs that have the potential to make it to market. We need to have sound processes that expedite the push to get these drugs to market. BDK can provide tremendous value in expediting the process of development.

How can we truly measure the success of BDK? We feel the success of a tool like BDK is measured by its adoption into industry along with whether or not it has an impact on the way industry does its process development. In order for BDK to have an impact it must get wide spread acceptance throughout the industry. For it to get acceptance the chemists and the engineers must actively want to use it. They will not use something if they are simply told it is useful by academics or even their supervisors. The chemists and engineers must find some value in the tool. The tool must either make their jobs easier or allow them to do something the previously could not. BDK meets both of these criteria. First it makes the jobs of the chemists and the engineers easier by providing a tool that acts as an assistant and performs the tedious tasks allowing them to focus their expert skills on the interesting problems. It also provides the ability for a better integration between the chemists and engineers which will ultimately lead to better more efficient design teams and better designs.

We have a preliminary indication that BDK has met the challenges to its acceptance. BDK's philosophy and design features have been broadly accepted by process chemists and process development engineers from a number of pharmaceutical and agricultural chemicals companies. In addition, BDK has been licensed by Hyprotech Ltd. which is making BDK the central piece in a strategy to completely integrate all aspects of the batch process design process. Hyprotech intends to integrate its an entire suite of process development tools with BDK as the backbone. These developments are manifested in BDK's compatibility with the needs of industrial batch process development. An additional measure of success it that BDK has spawned a similar batch process development tool by Hyprotech's chief competitor. Thus, we see that the industry clearly thinks that BDK will provide the benefits claimed here and will become an indispensable tool within the batch process industry.

7.4 References:

1. Linninger, A.A., *et al. BatchDesign-Kit: A Comprehensive Computer-Aided Design Framework with Ecological Considerations for Batch Process Development.* in *AIChE Annual Meeting.* 1996. Chicago, IL.
2. Linninger, A.A., S.A. Ali, and B. Harrison, *Batch Design-Kit Reference Manual*, . 1997, Laboratory for Intelligent Systems in Process Engineering: MIT, Cambridge, MA. p. 194.
3. Stephanopoulos, G., *et al.*, *Design-Kit: An Object-Oriented Environment for Process Engineering.* Computers and Chemical Engineering, 1987. **11**(6): p. 655-674.
4. Linninger, A.A. *M-Project - A Material-Based Approach for the Solution of Industrial Design Problems.* in *Conference of Mathematical Modeling and Computer Simulation of Industrial Problems.* 1993. Steyr, Austria.
5. Linninger, A.A., *et al.*, *Generation and Assessment of Batch Processes with Ecological Considerations.* Computers and Chemical Engineering, 1995. **19**(Suppl): p. S7-S13.
6. Salomone, H.E., *et al. Allocation of Multiputpose Production Facilities: A Computer Integrated Framework for Representing the Problem.* in *ENPROMER 97, Mercosur Conference on Process Engineering.* 1997. Bahia Blanca, Argentina.
7. Congress, U.S., *United States Code*, **42** Section 7602 (g)(h).
8. Congress, U.S., *United States Code*, **33** Sections 1251-1387.
9. Congress, U.S., *United States Code*, **33** Section 1362 (12).
10. *Book 4: Resource Conservation and Recovery Act*, in *Environmental Regulation Essentials*, M.A. McFaddin, Editor. 1993, Executive Enterprises: New York.
11. *Book 7: Toxic Substances Control Act*, in *Environmental Regulation Essentials*, J.P. Woodyard, Editor. 1993, Executive Enterprises: New York.
12. Linninger, A.A., S.A. Ali, and G. Stephanopoulos, *Knowledge-Based Validation and Waste Management of Batch Pharmaceutical Process Designs.* Computers and Chemical Engineering, 1996. **20**(Suppl): p. S1431-S1436.
13. EPA, U.S.E.P.A., *Control Technologies for Hazardous Air Pollutants, 7th Ed.*, . 1991, EPA.
14. EPA, U.S.E.P.A., *Technology Profiles, 7th Ed.*, 1994, EPA.
15. Wentz, C.A., *Hazardous Waste Management.* 1995, New York: McGraw-Hill.
16. LaGriega, M.D., P.L. Buckingham, and J.C. Evans, *Hazardous Waste Management.* 1994, New York: McGraw-Hill.

17. Linninger, A.A. and G. Stephanopoulos. *Modern Waste Management Using a Technology Assessment Tool.* in *AIChE Summer.* 1995. Boston, MA: AIChE.

18. Linninger, A.A., *Augmented Planning Networks for the Synthesis of Processing Schemes*, . 1997, MIT: Cambridge, MA.

Chapter 8

Conclusions, Contributions and Directions for Further Research

We begin this thesis with the challenge of developing a systematic approach for the synthesis of batch processing schemes for the production of pharmaceuticals and specialty chemicals. In order to meet this challenge, we have investigated the entire batch process development process and determined that a significant contribution could be made to the synthesis of batch processing schemes by systematizing the process synthesis phase of development. Until now, this area has been largely overlooked by most researchers. With this thesis we introduce the importance of the systematization of process synthesis, provide a systematic method for batch process synthesis, demonstrate how the method we developed integrates with the other phases of process development and improves the overall process development process.

The core of this thesis is the development of the Means-Ends Analysis with Nonmonotonic Planning methodology for batch process synthesis. The MEA-NMP methodology combines means-ends analysis, for the identification of differences between the current state of the system and the goal state, with the concept of nonmonotonic planning, for the generation of partial plans in a nonlinear manner. Our study of process development and process synthesis has allowed us to conclude that these tasks are, in essence, planning operations.

Batch process synthesis is the task of developing a plan for the processing of chemicals through the selection of batch process operations and process materials, which transforms the raw materials, the initial state, into the product material, the goal state. Nonmonotonic planning facilitates the efficient construction of feasible processes. We begin the process synthesis by first applying means-ends analysis to identify the differences between the initial state and the goal state and then identify operations to remove these differences. The key to the nonmonotonic planning is the assignment of preconditions to the operations. The preconditions indicate the feasibility of the operation. If the preconditions of an operation are violated, we identify another operation, called a white knight, which is capable of removing the precondition violation and place the white knight before the operation with the violated precondition, thereby removing the condition which caused the violation. In addition to white knight identification, nonmonotonic planning incorporates a concept known as promotion and demotion of operators. Promotion of operators allows us to move, or promote, an operation before a preceding operation, which causes a precondition violation of the operation being promoted. Demotion is simply the opposite of promotion. We chose to promote operations after our attempts at white knight identification have failed. This identification of operations to remove precondition violations combined with a concept known as operation promotion facilitate the construction of the process in a nonmonotonic manner. The justification for concluding process synthesis is a planning operation is given in Chapter 3 and a detailed description of the MEA-NMP methodology is given in Chapter 4.

The research in this thesis is part of the larger goal of completely systematizing the entire batch process development process. The systematization of the process synthesis phase is an important part of this goal since the process synthesis phase has a strong influence over the design phase. Also, it is at the process synthesis phase where we can begin to assess the decisions made during the chemical route synthesis phase and evaluate their impact on the development of the process. This evaluation of the decisions made at the chemical route synthesis phase allows us to provide feedback to the synthesis chemist before we have moved too far along in the development process. The feedback can be a valuable method for introducing the chemists to issues raised in the downstream phases of process development.

In order to gain the full benefit from the systematization of the batch process development process we have integrated the MEA-NMP methodology with the BatchDesign-Kit computer-aided process development tool. The MEA-NMP methodology has been implemented as a layered application called the *Process_Synthesizer*. This system acts as an assistant to the process developers allowing them to begin with the chemical reaction route and interactively develop the process through the selection of operations suggested by the system. The current implementation is a prototype of the full process synthesis system and only provides limited identification of operations. The system does however demonstrate that the implementation of the process synthesis system, described in this thesis, is feasible.

8.1 Conclusions and Contributions

Now that we have developed a systematic methodology for batch process synthesis, What can we conclude from this research? and what has this research contributed to the field of process development?

One of the most important conclusions that we can draw from this research is that the task of process synthesis can be systematized. As we have shown with the development of the MEA-NMP methodology, it is possible to develop methods for systematically constructing processes which use as a starting point the information provided by the synthesis chemist. While others have attempted to formally develop the synthesis problem as a mathematical programming problem, the scope and complexity of the problem has prevented those solution strategies from being applied to general practical problems. Other researchers have attempted to apply heuristic methods to the problem of process synthesis. The criticism of heuristic methods is that they are not systematic and have no way of knowing the quality of the process developed. The systematic methodology that we have developed, while not guaranteeing the development of an optimal solution, does allow us to systematically develop alternative processes that lie within the feasible region of the solution space. By keeping track of the decisions made and the possible alternatives at each decision node, we can rapidly develop and screen a large number of alternatives, increasing our ability to develop processes that meet our design criterion.

A crucial element in the systematization of batch process synthesis is the mathematical formalization of the problem. In order to accomplish this, we have employed two modeling concepts, the finite automaton and the State Task Network. The expression of the batch process synthesis problem using these concepts facilitates the mathematical formalism. The finite automaton and the STN allow us to structure the problem and then express each component mathematically. In addition to expression the operation models and the material models mathematically, we have demonstrated the use of logic to convert heuristics and constraints into mathematical expressions to be incorporated with the mathematical expression of the batch synthesis problem. Chapter 3 provides an extensive discussion of the concepts used to formalize the batch process synthesis problem.

The research we present in this thesis is the first attempt at the development of a process synthesis methodology specifically for batch processes. It is unique in adapting and combining a number of traditional approaches to process synthesis for continuous process synthesis, as well as, planning concepts from the Artificial Intelligence community. A large body of work has influenced and contributed to the methodology. Most notably Douglas's hierarchical decomposition, Grossmann's mathematical programming and logic, Siirola's means-ends analysis, Charalambides's mathematical formulation of the STN and Chapman's nonmonotonic planning. The contributions of each of these researchers and their coworkers have formed the building blocks and stimulated the ideas behind the development of the MEA-NMP methodology for batch process synthesis.

An additional contribution of this research is the development of the operations-centered approach to process synthesis. Through the development of the operations-based language, we have been able to show that process synthesis should be performed at the operations level not the equipment level. Performing process synthesis using an operation-centered approach allows us to give a more accurate description of what is taking place in the process.

The operations-based language is the natural language of process synthesis. It is the language the chemist uses when describing the chemical synthesis and it is the natural embodiment of the process that is being constructed by the engineer. Thus, the operations-based language facilitates communication between the two groups by expressing the process in terms familiar to both parties. The use of an equipment-based process development approach limits our ability to develop novel processes by adding unnecessary equipment related constraints at an early stage of process development. By focusing on the operations we are required to perform, we are released from forcing the process to be formulated in terms of specific pieces of equipment. An additional advantage of the operations-centered approach is that it facilitates the development of new operations. Through the merging of explicit operations, such as, the reaction operation and the extraction operation, we can develop new operations for example, reactive-extraction operation.

In addition to facilitating the development of alternative processes through interaction with the developer, the MEA-NMP methodology can be used to develop process superstructures. Through the maintenance of the task categories chosen at each node in the development of the base case design, we can simultaneously develop a superstructure of feasible alternatives from which a mathematical programming problem can be formulated through the use of mixed-integer nonlinear programming and logic based conversion of heuristics and constraints into their mathematical equivalent. As we have discussed in Chapter 6, the size of full scale optimization is too large to be handled with today's solution methods for optimization problems, especially in the face of the nonconvexities contained in many of the models used for batch processes. The MEA-NMP methodology generates reduced superstructures which significantly reduce the search space of alternatives but, even with the reduced superstructure generated by the MEA-NMP methodology, the number of alternatives may still be exceedingly large for solution using optimization routines, especially for complex processes.

What have we gained through the systematizing the approach to process synthesis? The systematic approach to process synthesis has provided us with a number of advantages over the traditional approaches to process synthesis. First, is that we have an explicit method for tracking the decision making process. This is the key to better process synthesis. By tracking the decisions that are made during the process synthesis along with the reasoning behind why they were made, we are in a position to learn from our previous designs thus, capturing some of the experience of the previous developers. If flaws are found in our processes, we can go back and study the decisions that were made and use this information to improve future processes. For industry, perhaps, the most important benefit of a systematic approach to process synthesis is the ability to reduce the time-to-market for high value-added products. As we mentioned in Chapter 7, this is extremely important especially for the pharmaceutical industry where each day can be worth millions of dollars. Decreasing the time-to-market increases the profit for the company by increasing the time the product is being sold under patent protection. Also, since we can develop processes faster, we can examine a greater number of alternatives allowing us to deploy better processes into manufacturing. Through a better understanding of the process, we are less likely to have processing variations which result in fewer batches not meeting product specifications. A sound record of high quality standards and consistent processing ability makes the FDA approval process go more smoothly and reduces the approval time for new drugs. Another advantage of deploying efficient processes is the possibility of providing protection against generic competitors. If the process that is deployed is the most efficient process possible, once patent protection expires, it will provide flexibility in pricing which could deter generic competition.

Finally, the implementation of a fully integrated computer-aided process development system has the potential to significantly impact the way processes are developed and deployed in the batch chemical industry. The implementation of the MEA-NMP methodology within the *Process_Synthesizer* of BDK has the potential to help the process developers increase their efficiency in developing processes and knowledge of the process development process. The integrated system provides a central knowledge and data repository, which could be used throughout the enterprise, reducing the need for the reproduction of data and computer systems within the firm and through the development process. The computer-aided system will also relieve the developers of the burden of performing some tedious tasks that have traditionally been performed across distributed systems such as using spreadsheets for material balance calculations, drawing packages for flowsheet construction and word processing tools for generating reports. A fully integrated system such as BDK combines all these features into a single system thus, facilitating communication and information sharing throughout the firm. Additionally, we can and have implemented sophisticated process analysis and evaluation facilities, which allow the developer to evaluate alternative processes based on health, safety and environmental impact criterion. This will allow the development of processes that operate away from the bounds of regulatory compliance, which are healthier, safer and more environmentally friendly. All these features definitely have the potential to decrease time-to-market, increase the quality and reduce the over all cost of processes which will be deployed into manufacturing.

8.2 Directions for Future Research

With the increasing importance of batch processing within the chemical industry and the dramatic increase in the amount of research in the area of batch processing by academics, the work presented in this thesis should not be viewed as the final solution to the synthesis of batch processing schemes but, more of a renewed beginning. The work of this thesis provides a starting point for the investigation of new concepts in the synthesis of batch processing schemes. Hopefully, future researchers in the area of process synthesis can take advantage of what has been learned through this work and expand on the concepts we have introduced. The concept of the operations-centered approach to process synthesis will undoubtedly provide the basis for new approaches in process synthesis, as well as, promote the development of novel batch process operations. Additionally, we have broken ground on the idea of combining multiple previous attempts at developing systematic approaches to process synthesis. We encourage future researchers to continue to improve upon the ideas that have been developed in this thesis, as well as, take advantage of any new and novel approaches to process synthesis.

We would like to extend the challenge posed by Rippin [1] and advanced by Daichendt and Grossmann [2] by suggesting that we, not only look at the two main views of process synthesis (hierarchical decomposition and mathematical programming) through the eyes of each other but, look at the process development problem through the eyes of all previous research efforts, while determining the directions for future work. Fore, if we continue with the development of processes by using either one approach or the other, or even a combination of the two, we are artificially limiting ourselves to methods which may not be capable of finding the solution we seek. Since we have no way of knowing where the solution lies, we must be willing to examine the batch process development and batch process synthesis problems in different and new ways.

Along with looking at process development and synthesis in new and different ways, we must also be ready and willing to identify and exploit new advances in the fields of *information technology* and Artificial Intelligence. In our work, we have made extensive use of advancements in these areas and we are sure that additional advancements in these fields will help us to make equally important breakthroughs in the field of batch process development. Improvements in the expert system tools, which are available, will help us to develop more sophisticated expert systems for process development. We can begin to imagine systems which can "learn" and become more effective assistants through the analysis of the decision making process. Work in MIT's AI lab, in the development of systems that try to understand the engineer's decision making process and offer suggestions to the engineer during the design process indicates that these tools will someday be available to anyone who is ready to take advantage of them. These systems will only be of use to us if we stay abreast of their progress and are willing to provide the time and effort to adapt them to our specific area of interest, batch process development.

In addition to work in batch process development, our research has also generated ideas for additional areas for future research. These ideas are introduced below.

8.2.1 Selection of Process Materials

The process development problem begins with the chemist synthesizing the chemical route for the process. In this thesis we have suggested that, during the process synthesis phase of development, an assessment of the process be performed in order to determine if any materials used in the process exhibit properties that will cause problems with respect to health, safety or environmental impact. What this implies is that we need to look at the material selection problem. Is there a way we can help with the selection of process materials that will facilitate better designs? Two areas that show promise are solvent selection and the development and selection of homogeneous catalysts.

8.2.1.1 Solvent Selection

Until now there has been very little work in the area of process materials selection. Modi et al [3] have developed a system to assist with the selection of replacement solvents in a chemical process. Using the system, the developer specifies the current solvent and the offending properties, and the system returns a list of solvents that could potentially replace the original solvent based on the properties of the solvents. Aumond [4] has been working to develop a solvent selection system which will help the developer select solvents that enhance reaction rates. This system provides the developer with a list of possible solvents and leaves it up to the developer to select the solvent from the list that has the most desirable characteristics.

It should be clear that systems like these could be extremely beneficial to the development process, especially if applied during the chemical route development. We could also envision expanding systems like these, which will allow the developer to input many different design criterion and the system would return a list of most promising solvents that have the desired behavior, as well as non-offensive properties.

8.2.1.2 Homogeneous Catalysts

Along with the selection of solvents, there is an emerging interest in the development of *homogeneous catalysts.* The development of very specific catalysts that produce a specific product without any unwanted by-products would also have a dramatic impact on process development. The homogeneous catalyst has the potential for significantly simplifying the process which is developed, through the reduction of the number of purification steps that would normally be needed in order to remove the unwanted by-products. It should be obvious that this would also remove waste treatment problems by not producing the waste in the first place. Since the reduction of the number of separations generally reduces the amount of solvents that need to be used and a large part of the health, safety and environmental concerns are generated through solvent usage resulting in an increase the safety and health of the process, as well as, reducing the environmental impact. The effort in this area should be applied to developing a systematic method for the identification of these homogeneous catalysts, along with methods for their implementation with chemical processes.

8.2.2 Relaxation of Intermediate Crystallization

In all of the pharmaceutical processes and many of the specialty chemical processes we have examined, we have found that the intermediate products are produced through crystallization as a solid, only then to be re-dissolved in a solvent in later processing steps. Through discussions with our industrial contacts, we have found that this is done in order to guarantee process robustness. The isolation of intermediate products as solids guarantees process robustness through the elimination of any variation in the process by removing the possibility of disturbances in the process propagating through the process. Each time an intermediate is isolated as a high purity solid, all potential disturbances are removed. And, if the intermediate product does not meet specifications, it can be disposed of or additionally purified, before further processing. These frequent check points help ensure product quality and consistency, which is most important within the pharmaceutical industry.

The problem with this approach is that it forces the inclusion of many extra processing steps and frequently, the usage of additional solvents increasing processing time, as well as, the amount of waste produced. From a chemical engineering standpoint much of the extra processing that is now performed could be avoided by moving the intermediate material from stage to stage as a solid dissolved in a solvent that has already been used in the processing stage, in essence, do not crystallize the intermediate products. This can be accomplished through two changes which should be investigated. One of the methods is better selection of process materials as we have mentioned in the previous section. If we can select a solvent that can be used across many of the processing stages, we would eliminate the need to isolate the solid intermediate in order to perform solvent replacement. Additionally, the concept of selecting highly specific reactants and homogeneous catalysts would reduce the possibility of disturbances in the process. The highly specific reactions would only be capable of generating the desired product thus, reducing the possibility of the product being contaminated with unwanted by-products. The area of highly selective reactants and catalysts can provide a significant increase in the efficiency of the processes developed and is worth investigation.

8.2.3 Exploration of Microplants

Currently, the determination of whether or not to develop a process in a batch or continuous mode is made based, for the most part, on the amount of product to be produced, although there are additional factors that influence the decision of batch versus continuous such as, safety i.e. explosives are made exclusively in a batch mode to limit the possibility of a large explosion due to an accident, and seasonal demand i.e. many agricultural products are required at specific times of the year. Generally, it is not economical to produce small amounts of product in a continuous mode of production, since the economies of scale dictate that there is no economic advantage for developing a continuous process unless the production rate is greater than a certain amount.

The advancement of micro and nano technologies is beginning to change the criterion used for determining whether or not processes should be continuous or batch. The advancement of these technologies has allowed us to consider replacing batch processes with a large number of small scale continuous processes. Small scale continuous processing could provide a number of advantages over traditional batch processing. For example, if we could construct a microplant that is the size of a large computer chip, we could run thousands of these small continuous processes to meet our product needs. The advantage of using such a small production scale for dangerous processes, which may be difficult to control, is that if a run-away reaction occurs and the plant explodes, there is only a minimal safety risk, as well as, minimal economic loss due to the small scale of the explosion. And, if these plants could be produced cheaply enough, they could become disposable so that if we are producing a biological based product and we are concerned about cross product contamination, we could simply build a new set of plants after each production run and dispose of the old ones, eliminating the waste produced by cleaning the larger scale batch process and eliminating concerns about improper cleaning of the batch process equipment.

We have already seen one example of this technology in use. Affymetrix Corporation has developed what is known as a *gene chip*. One application of this technology is genetic screening of protein samples. In a join venture Hewlett Packard and Affymetrix (http://chem.external.hp.com/cag/products/protein.html) have developed a small scale chemical "plant," about the size of CD-ROM jewel case, for the separation and preparation of the protein sample to be screened. This plant takes as input a small protein sample and then performs chemical processing and mechanical separation in order to process the sample and perform the genetic screening. While the process is used in a batch mode for each sample, it does provide an excellent example of micro technology finding industrial application. Since what Affymetrix and HP have developed is basically a small scale chemical plant, this gives us a strong indication that small scale continuous chemical plants are not far from becoming reality.

The research in the area of microplants should continue through the study of producing and improving the technology to produce these microplants. It should also expand to cover the issues of controlling such plants in order to be able to run them in a continuous mode for long periods of time while maintaining constant product quality. As part of this effort in the area of microplants, we will need to reexamine the process development, synthesis and design activities in order to take full advantage of these new processing facilities. This should provide interesting and unique challenges for future researchers in the fields of process development, process synthesis and plant design.

8.3 References:

1. Rippin, D.W.T. *Introduction: Approaches to Chemical Process Synthesis.* in *Foundations of Computer-Aided Process Design.* 1989. Snowmass, Colorado.
2. Daichendt, M.M. and I.E. Grossmann, *Integration of hierarchical decomposition and mathematical programming for synthesis of process flowsheets.* Computers and Chemical Engineering, 1997. **22**(1-2): p. 147-175.
3. Modi, A., *et al.*, *Rapid Plant-Wide Screening of Solvent for Batch Processes.* Computers & Chemical Engineering, 1996. **20**(Suppl. A): p. S375-S380.
4. Aumond, J.P., *Solvent Selection for Reactions*, . 1998.

Appendix A

Additional Pharmaceutical Cases

This Appendix contains two additional pharmaceutical cases provided by a leading pharmaceutical company. We provide these cases as additional examples of typical pharmaceutical processes. These examples were developed from the actual process description provided by the process chemist. The two cases which follow contain the reaction chemistry and the flowsheet of operations performed by the chemist. Case Study I contains nine stages similar to the Carbinol case developed in Chapter 5. Case Study II consists of a single stage with multiple reactions. In these case studies we do not include the operating conditions. The reader can assume that all cooling and heating operations are relative to ambient conditions and that any heating and cooling taking place before a reaction are used to adjust the reaction operating conditions relative to ambient conditions. All reactions are liquid phase reactions and the yields given are those reported in the chemist's recipe. The catalyst and solvent information is indicated above the arrows in the reaction descriptions.

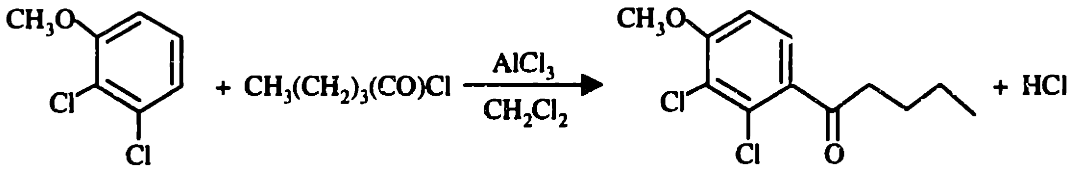
We notice that the information provided by the chemist is incomplete. In both cases and particularly in Case Study II, the reaction information is incomplete i.e. the chemist does not provide the by-products produced and also does not provide information whether any side reactions are taking place. Also, the process description does not always specify the order of addition of materials into the process, which can be important in order to inhibit side reactions. The chemist does provide the operating conditions for the operations but we were required to omit them here. These cases do show however, that information is sparse and in order to perform the design we must make assumptions as well as gather any crucial information. Additionally, the process description does not always convey the information properly. For example, within both cases the chemist simply states that a *quench* operation was performed. The process description does not include whether the operation is a *thermal quench* or a *reactive quench* used to complete the reaction. This is important when attempting to create an inventory of all the by-products produced by the process and in determining the exit points for those by-products.

These cases further illustrate the need for systematization of the process of process development by highlighting the need for communication between the chemist and the process engineers in order to ensure that all important information is transferred among the different participants in the development process.

A.1 Case Study I

Case Study I describes the nine stage process for producing a potassium salt. This process includes both liquid and solid phase intermediates transferred from stage to stage. Stage 3b is used for the production of a catalyst used in Stage 4. Note: a bold question mark indicates an assumed by-product not included in the original process description.

A.1.1 Stage 1



(I) **(II)**

Figure A.1 Stage 1 Reaction Chemistry

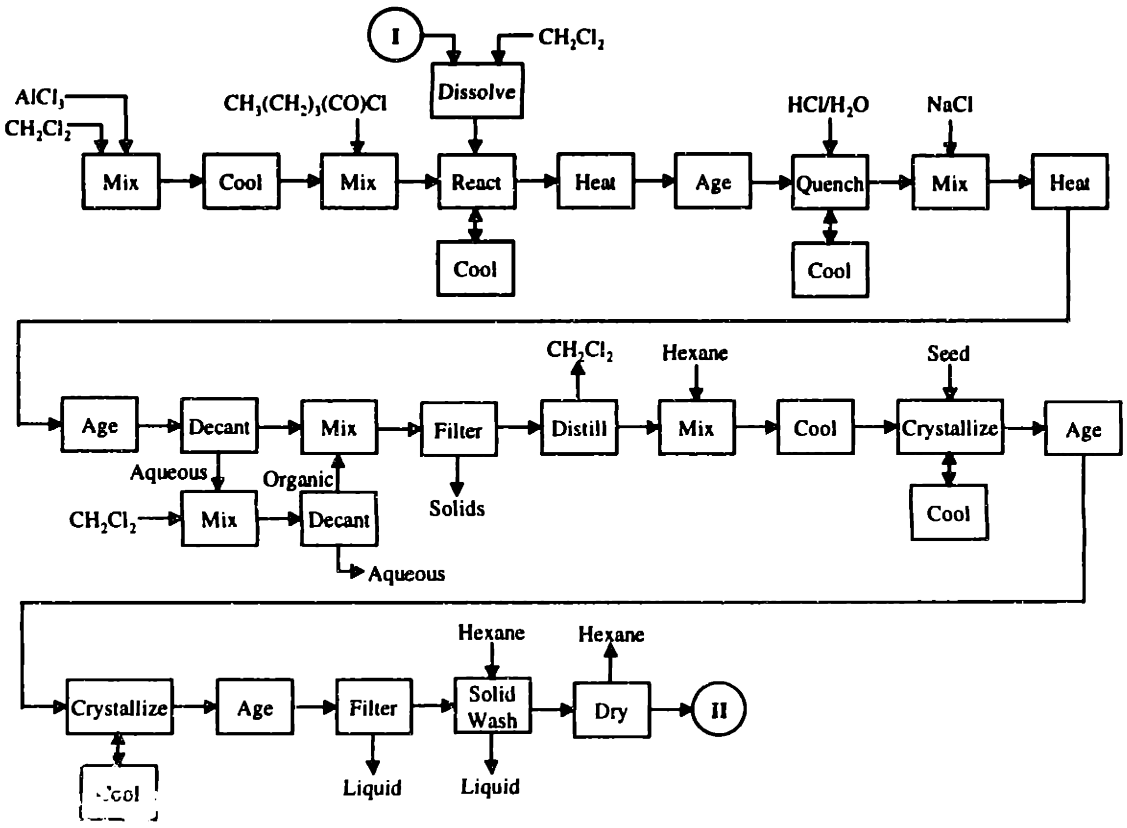


Figure A.2 Stage 1 Operation Flowsheet (Yield = 82 %)

A.1.2 Stage 2

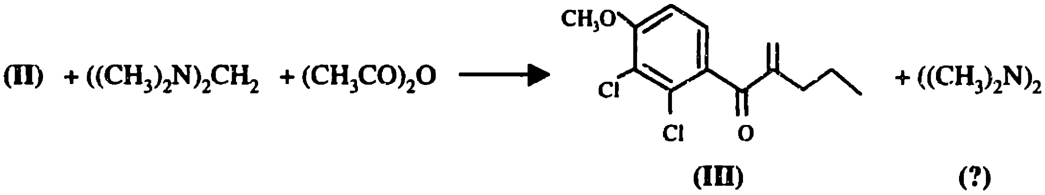


Figure A.3 Stage 2 Reaction Chemistry

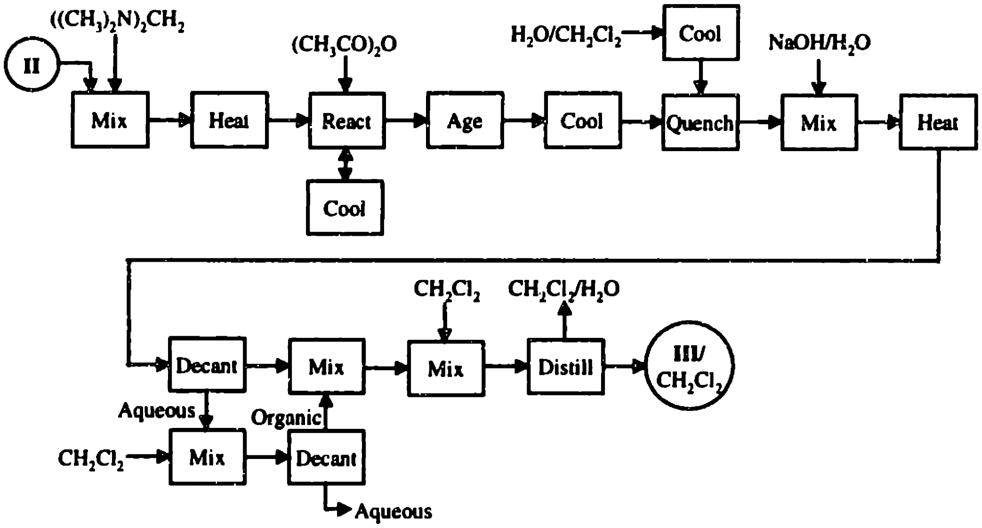
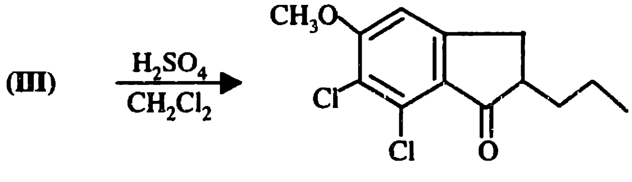


Figure A.4 Stage 2 Operation Flowsheet (Yield = 99 %)

A.1.3 Stage 3



(IV)

Figure A.5 Stage 3 Reaction Chemistry

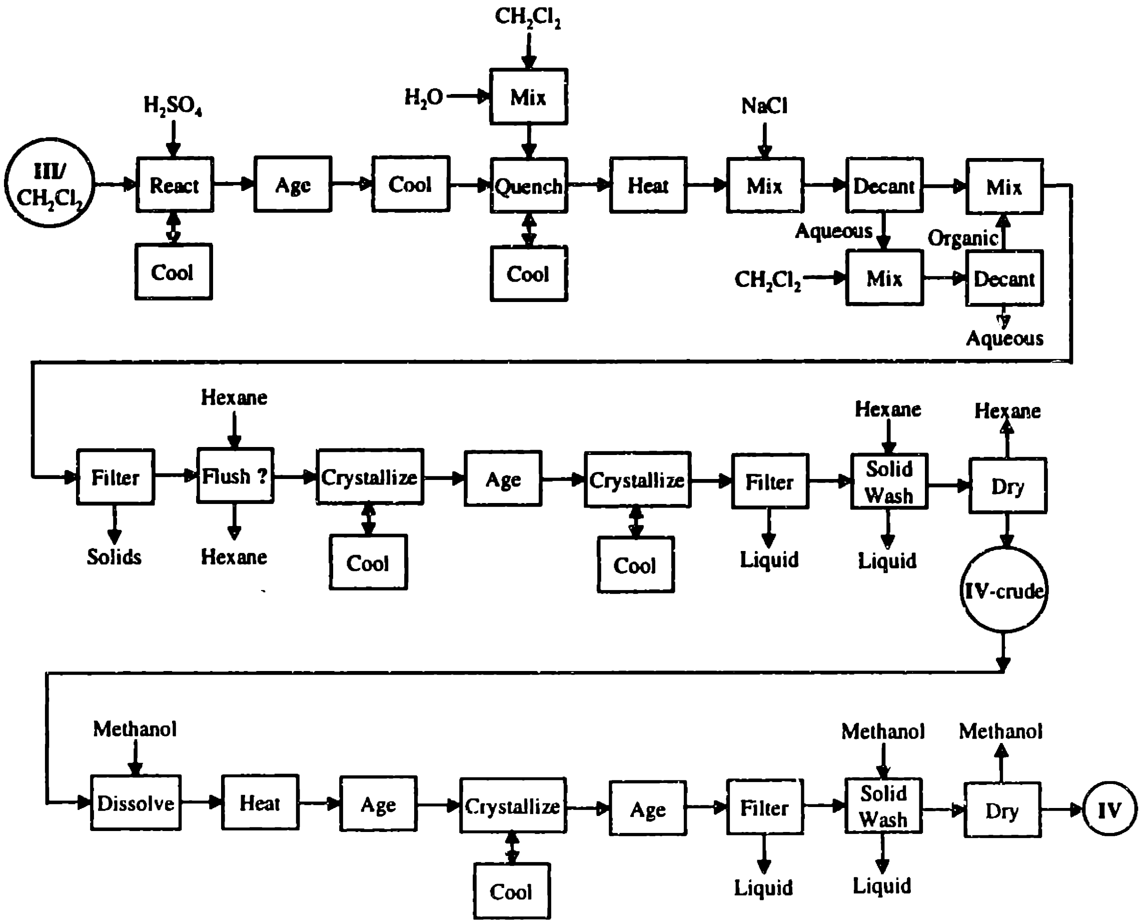
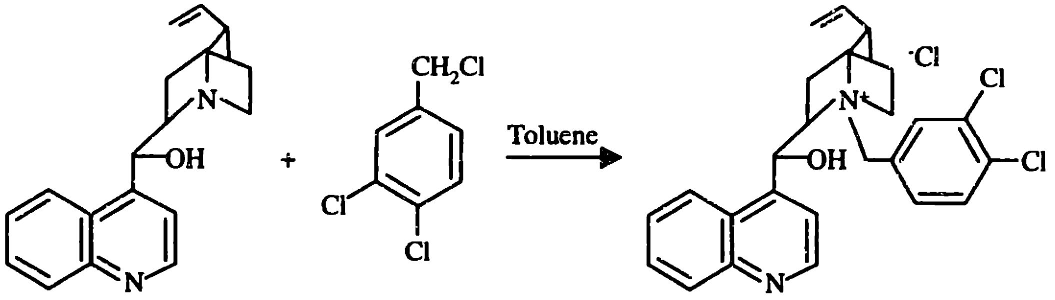


Figure A.6 Stage 3 Operation Flowsheet (Yield = 88 %)

A.1.4 Stage 3b (Catalyst Production)



(A) (B) (Cl$_2$BCDC)

Figure A.7 Stage 3b Reaction Chemistry

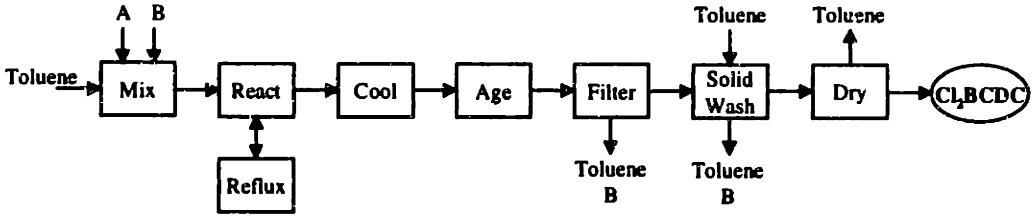
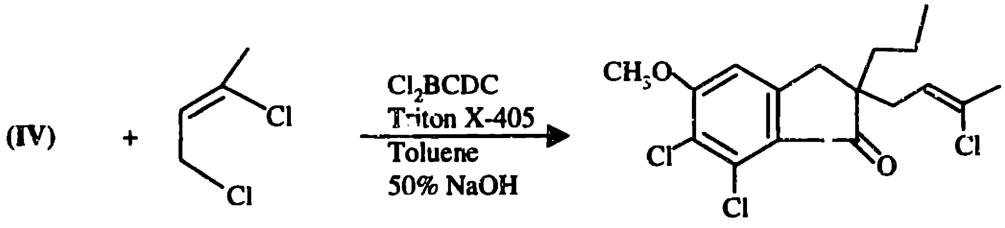


Figure A.8 Stage 3b Operation Flowsheet (Yield = 97 %)

A.1.5 Stage 4



(C)

(V) - 80/20 (+/-)

Figure A.9 Stage 4 Reaction Chemistry

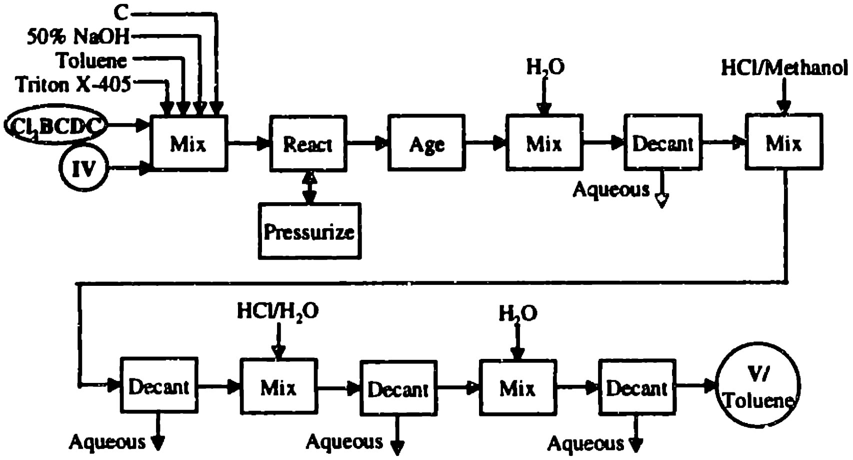


Figure A.10 Stage 4 Operation Flowsheet (Yield = 100 %)

A.1.6 Stage 5

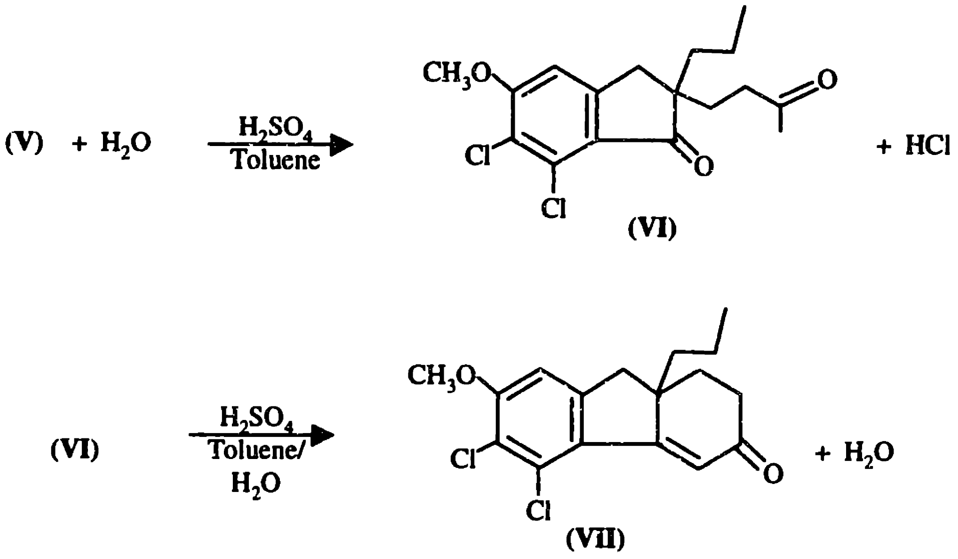


Figure A.11 Stage 5 Reaction Chemistry

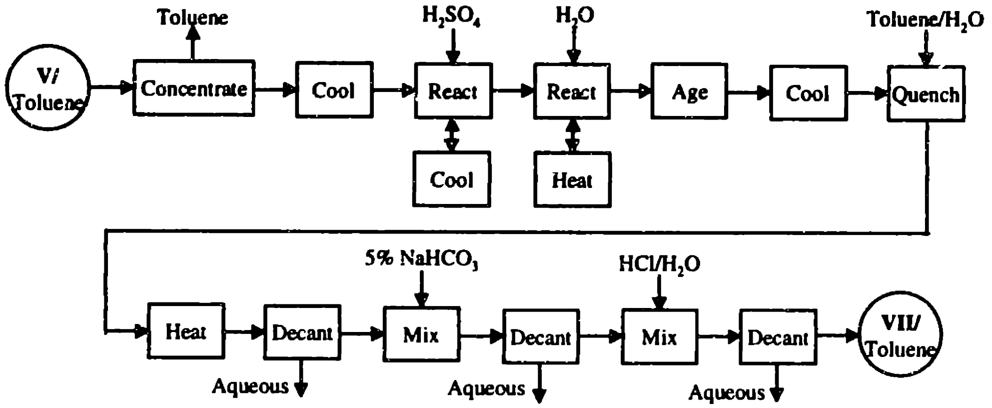
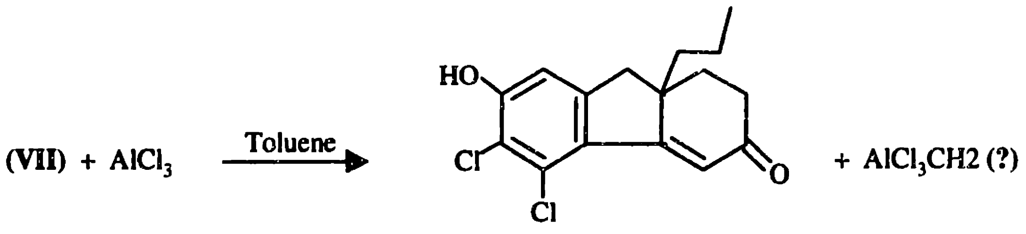


Figure A.12 Stage 5 Operation Flowsheet (Yield = 93 %)

A.1.7 Stage 6



(VIII)

Figure A.13 Stage 6 Reaction Chemistry

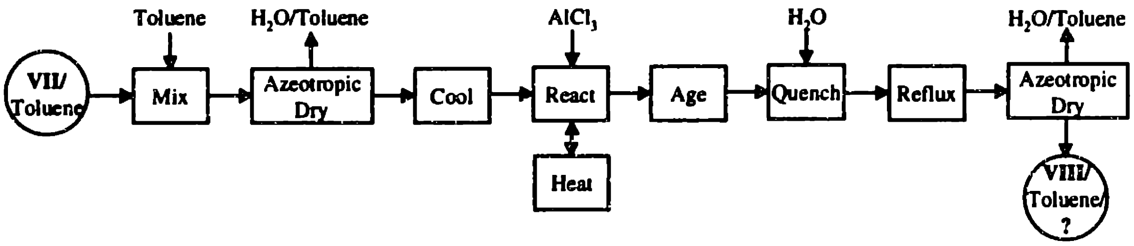
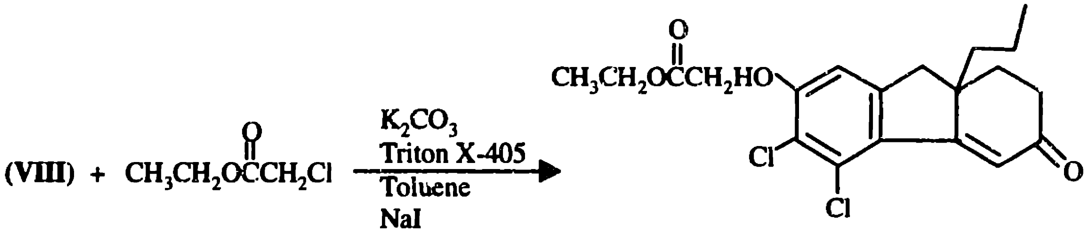


Figure A.14 Stage 6 Operation Flowsheet

A.1.8 Stage 7



(D) (IX)

Figure A.15 Stage 7 Reaction Chemistry

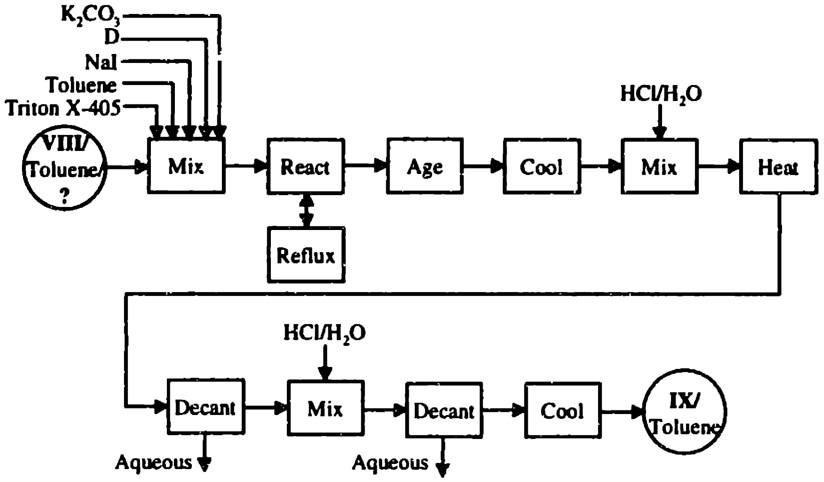
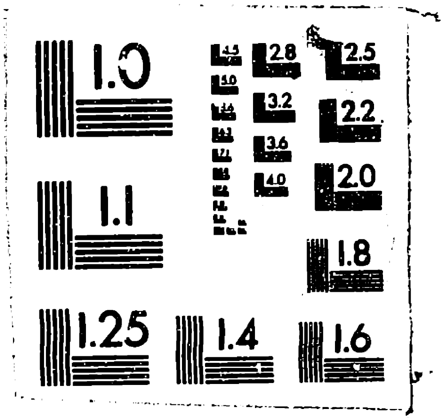


Figure A.16 Stage 7 Operation Flowsheet


1.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

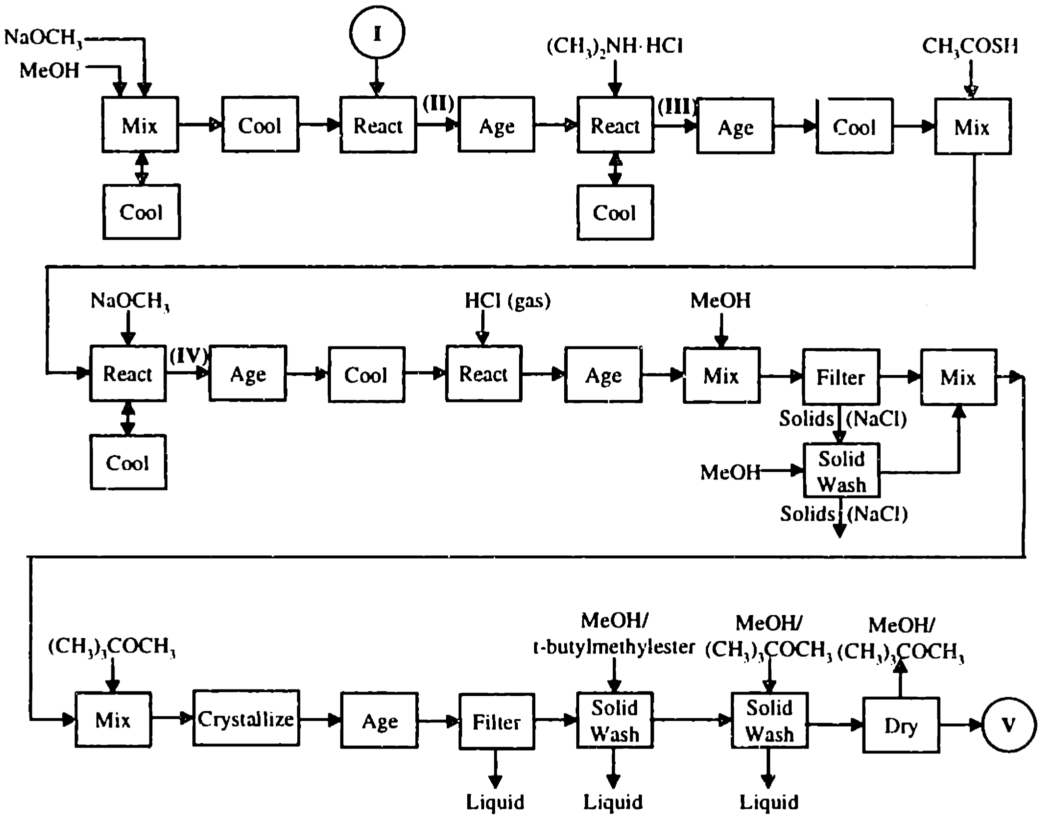
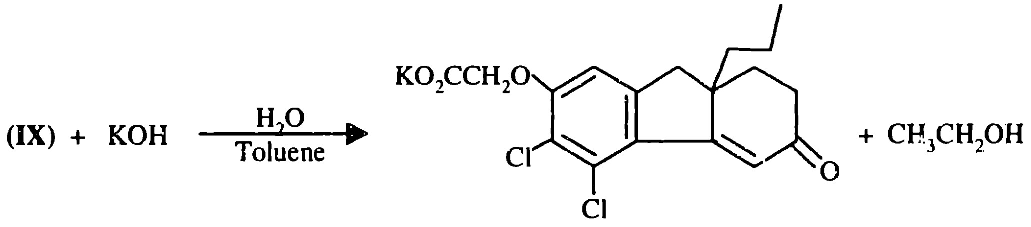


Figure A.20 Stage 8 Operation Flowsheet (Yield = 78%)

A.1.9 Stage 8



(X)

Figure A.17 Stage 8 Reaction Chemistry

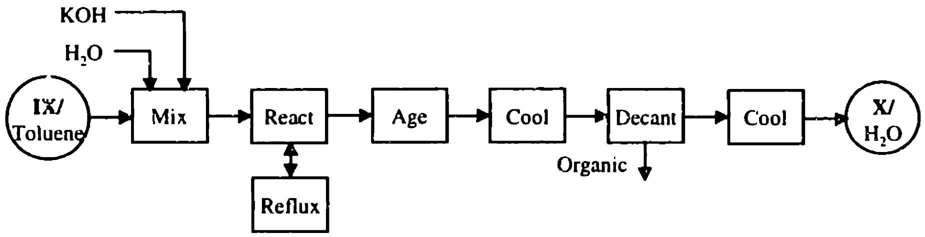


Figure A.18 Stage 8 Operation Flowsheet (Yield, Stages 6-8 = 90%)

A.2 Case Study II

Case Study II is a single stage multiple reaction process, in which the reactions are performed sequentially in the stage. The entire process is performed under a nitrogen atmosphere. Notice, in the reaction chemistry the description does not include the stoichiometric by-products. The actual process description also did not include this information.

A.2.1 Stage 1

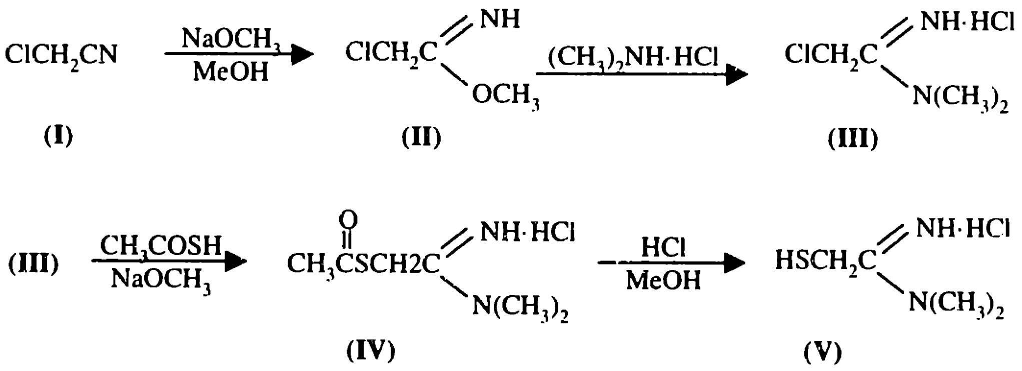


Figure A.19 Stage 1 Reaction Chemistry