8-15-2003

Development of a Systems Engineering Model of the Chemical Separations Process

Yitung Chen
University of Nevada, Las Vegas, yitung.chen@unlv.edu

Sean Hsieh
University of Nevada, Las Vegas, hsiehht@nscee.edu

Randy Clarksean
University of Nevada, Las Vegas

Darrell Pepper
University of Nevada Las Vegas, pepperu@nye.nscee.edu

ANNUAL REPORT

Development of a Systems Engineering Model of the Chemical Separations Process

Submitted to

Transmutation Research Project Program
Technical Focus Area
Fuel Development Research
ATTN: Dr. Anthony Hechanova
Harry Reid Center
University of Nevada Las Vegas

Submitted by

Dr. Yitung Chen, Principal Investigator, uuchen@clark.nscee.edu
Dr. Sean Hsieh, Co-Principal Investigator, hsiehht@clark.nscee.edu
Dr. Randy Clarksean, Co-Principal Investigator, rclark@lakesplus.com
Dr. Darrell W. Pepper, Co-Principal Investigator, pepperu@clark.nscee.edu
Department of Mechanical Engineering
University of Nevada Las Vegas
4505 Maryland Parkway, Box 454027
Las Vegas, NV 89154-4027
Phone: (702) 895-1202
Fax: (702) 895-3936

Collaborators

Dr. James J. Laidler, Senior Scientist
Dr. George F. Vandergrift, III, Senior Scientist
Chemical Technology Division
Argonne National Laboratory, Argonne, IL

Project Dates: 9/1/02 – 8/31/03

August 15, 2003

Table of Contents

1. Introduction

The United States is embarking on a national program to develop accelerator transmutation of high-level radioactive waste as part of the Advanced Fuel Cycle Initiative (AFCI) at its national laboratories. Through the program, the U.S. joins international efforts to evaluate the potential of partitioning and transmutation along with advanced nuclear fuel cycles. Transmutation means nuclear transformation that changes the contents of the nucleus (protons and/or neutrons). The research and development efforts will consider a coupled accelerator and sub-critical multiplying assembly, explore the transmutation of waste from used nuclear fuel, and test advanced nuclear fuels.

The AFCI program is developing technology for the transmutation of nuclear waste to address many of the long-term disposal issues. An integral part of this program is the proposed chemical separations scheme. This process, as envisioned by Argonne National Laboratory (ANL) researchers, will be outlined later in this report.

Nearly all issues related to risks to future generations arising from long-term disposal of such spent nuclear fuel is attributable to about 2% of its content. Such 2% is made up primarily of plutonium, neptunium, americium, and curium (the transuranic elements) and long-lived isotopes of iodine and technetium created as products from the fission process in power reactors. When transuranics are removed from discharged fuel destined for disposal, the toxic nature of the spent fuel drops below that of natural uranium ore (that was originally mined for nuclear fuel) within a period of several hundred years.

Removal of plutonium and other transuranics from material destined for geologic disposal also eliminates long-term (centuries) heat management issues within such environments. The removal of neptunium, technetium, and iodine render negligible the possibility of radioactive material penetration into the biosphere in the future. Finally, removal of plutonium negates any incentive for intrusion into repositories driven by intentional recovery of material for nuclear proliferation.

The complete process considers existing LWR spent fuel, separation processes, fuel fabrication, transmutation, low-level waste disposal (LLWD), and the reprocessing of fuel after transmutation. This is an involved process that can be varied in a number of ways. Any proposed change to one of the processes can have impacts on the others. In a nuclear growth scenario, the introduction of advanced thermal reactor designs will almost certainly result in changes in separations system requirements that must be met with optimized systems.

Developing a systems engineering model of the overall process would be beneficial to analyzing complex interactions between proposed process changes. The model will evolve to incorporate all process steps and to improve process modules as more knowledge is gained. The improvements will be based on empirical data or from numerical models as appropriate.

An integral part of the overall chemical process is a UREX (Uranium Extraction) process. This portion of the process is currently modeled by AMUSE code, developed by ANL. A brief description of this portion of the project is given later in this report.

2. Project Overview

The following objectives remained the same from the first year of the project. Meeting these objectives will lead to a concrete and useful systems engineering model.

- Develop a framework for a systems engineering analysis of chemical separations process.
- Establish a baseline systems engineering model from which modifications can be made.
- Refine the existing AMUSE program that gives a detailed examination of the UREX process, a critical component of the overall separation scheme.

2.1 Activity 1: System Engineering Model

2.1.1. Define Process Models

Discussions will be held with ANL researchers to clearly define key components from overall chemical separation process. Each step of the process needs a "model" to define its functionality as well as its integration with the overall process. Key issues include:

- Identification of overall chemical separation process,
- Identification of targets for the designed system engineering model,
- Determination of software development environment,
- Designation of model modules, input and output for process optimization.

2.1.2. Demonstrate Modeling Concept

Since the model modules are still under development. We will demonstrate some capability and possible tools for the system analyses/simulations, based on the previously defined models. Key issues includes:

- Demonstration of techniques for design of experiment (DOE),
- Demonstration of graphical user interface and model modules,
- Summarization of optimization strategies.

2.2 Activity 2: Improvement/Automation/Modernization of AMUSE

2.2.1. Review/Analyze Code Structure

This work will insure that all input data is properly stored within the existing AMUSE environment. Files from ANL are used for interface layout and code development. Workflow is analyzed to determine how the process and calculations are performed.

2.2.2. Develop Verification Plan

All changes to the new interface need to be verified numerically. A set of test problems will be developed to demonstrate the numerical accuracy. Due to the Export Controlled issue on the AMUSE code, verification task continues from year one to year two.

2.2.3. Modify/Improve Software

Make changes to the software that allows it to be used as a part of the overall systems engineering model and that allows it to be used as a stand-alone process model. Determine what modifications would be necessary to allow the process optimization.

3. Progress - Systems Engineering Model

Since the chemical separation process is complicated in nature and understanding those calculations within each process require significant amount of chemistry knowledge, email communication and on-site discussion with ANL researchers are the major mechanism for clarifying integration and programming issues. A brief overview of the work completed for the systems engineering modeling is discussed in the following subsections.

3.1 Define Process Models

3.1.1. Identification of Overall Chemical Separation Process

Figure 1 shows the flowchart for the overall Chemical Separation System. This figure depicts the fuel cycle scheme in which the transuranic elements and long-lived fission products from spent LWR fuel are sent directly to an accelerator-driven subcritical reactor for transmutation. The model is provided by Argonne National Laboratory Researchers. The focus has currently been on the Uranium Extraction Process (UREX), ultimately we will extend the system engineering model to all processes.

3.1.2. Identification of Targets for the System Engineering Model

A general software tool that allows us to build the system simulation model and do the system analysis will be defined. We will apply it the complex processes of the AFCI Chemical Separation Process. Major goals of the work are:

- Developing a friendly graphical interface with system model installed,
- Implementing optimization tools,
- Implementing parameter sensibility analysis module,
- Calculating mass balance for all process streams,
- Evaluating various assumptions on process performance

3.1.3. Determination of Software Development Environment

Three possible approaches for implementing the system model were considered: developing a completely new package from the bottom up, modifying existing software, and applying an existing product directly without modification (commercial software). During year one of the research project numerous commercial packages and development environments were identified, such as iSight™, MATLAB™, LBNL – SPARK (systems tool), ASPEN™ – process modeling tool, Easy 5™ (Boeing), Visual Basic™ / Visual C/C++™ and Microsoft™.NET.

As we discovered, no one package can be applied to our research objectives and solve the problems. While developing a new software product from scratch requires a great deal of

planning and implementation time, it can provide a fully integrated and optimized research environment with high interoperability and scalability. On the other hand, using existing software provides for a shorter development time at the cost of less flexibility. For the purpose of a quick jump-start, we used iSight™ to model the system in year one. To achieve a highly flexible and complete system engineering model design, from the second year on,decided to develop new interface from scratch with the inclusion of commercial optimization toolbox modules, such as Matlab Optimization toolbox. The system will be able to:

- Build the simulation model for general process as well as the whole AFCI Chemical Separation Process,
- Plug in appropriate optimization schemes from existing commercial object libraries as needed for various chemical separation processes,
- Apply multi-objective optimization technology,
- Analyze the system using Design of Experiments (DOE) studies.

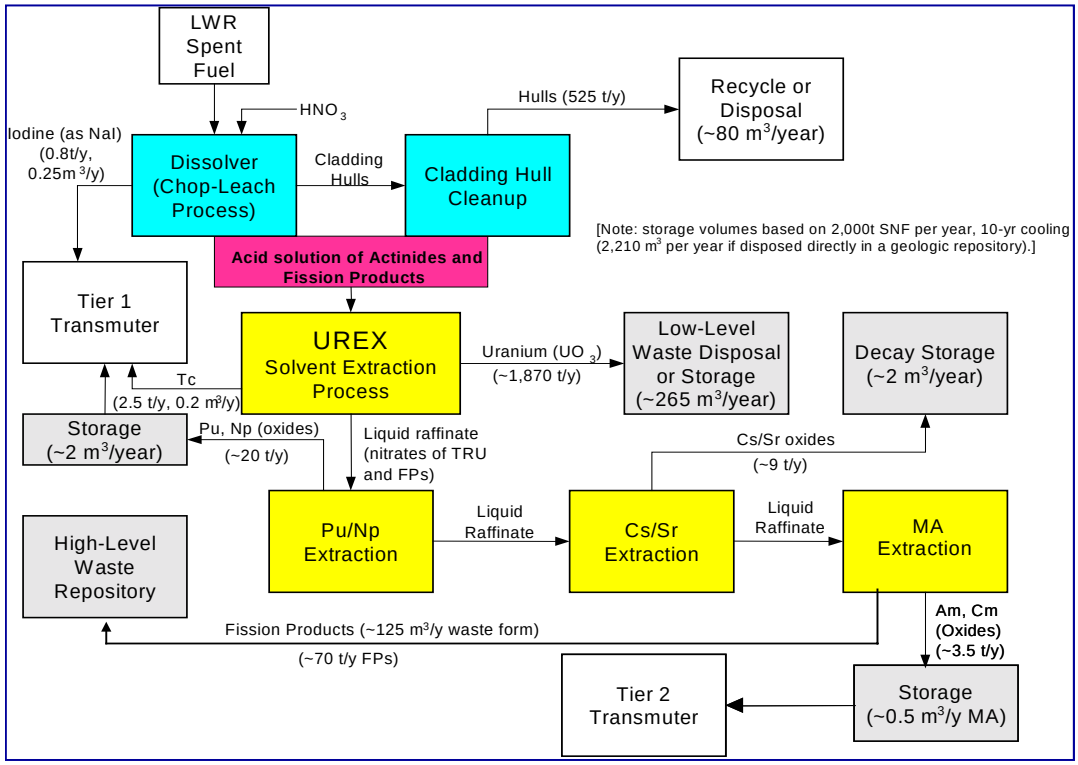


Figure 1 - Overall Chemical Separations System for AFCI Program

3.1.4. Designation of Model Components and Parameters

From a system engineering viewpoint, a system is any process that converts inputs to outputs. A system creates outputs based on inputs, over which it has no direct control, and the system's present state. The current system state and a sequence of inputs allow

computation of the future states of the system. Figure 2 illustrates the general system. The designed system model includes three main parts:

- Flowsheet Simulator – a module for building input simulation parameters,
- Task Manager – a module for computing, managing and arranging all the possible tasks, such as single/multiple run scenarios, parameter definition, DOE, optimization tools, et al.
- Solution Monitor – a module for timely updating and monitoring output data fed by model simulation.

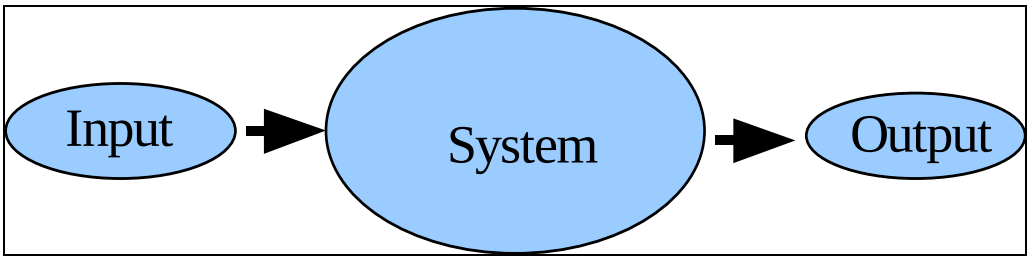


Figure 2 - General System

A framework design for the new engineering system is shown in Figure 3. The input parameters and simulated results are passed to the database that serves as a depository for later report compilation and data retrieval. We use MS SQL Server database to store all information regarding the run. The technique, ADO.Net, is used to connect the database with other interfaces. Figure 4 shows the detailed dataflow between the database, flowsheet simulator and AMUSE package.

Parameters can be defined by specifying variable, objective, and constraint information.

- Variable – adjustable design variables from input interface.
- Objective – need-to-be-determined variables generated from simulation output.
- Constraints – limitations posed on those design variables.

Due to the overwhelming number of variables used by AMUSE macros, it is nearly impossible to have all variables set to be adjustable. Therefore, based on any specific research objective, the parameters of interest can be listed in the flowsheet parameter table with their attributes assigned to variable or objective. Certain constraints imposed for each variable. Figure 5 shows the scenario for testing the impact of aqueous flow rate on the concentration of H_2. The aqueous flow rate is set as a design variable and the effluent concentration of H_2 is set as an objective variable.

To optimize the data communication and query flexibility, our new system uses Extensible Markup Language (XML) file format. XML provides a method for describing structured data. XML is a subset of SGML that is optimized for delivery over the Web. The World Wide Web Consortium (W3C) defines XML standards so that structured data will be uniform and independent of applications. Visual Studio .NET fully supports XML, providing the XML Designer the ability to easily edit XML and create XML schemas. XML can be treated as a set of rules for defining semantic tags that break a

document into parts and identify the different parts of the document. It is a meta-markup language that defines a syntax in which other field-specific markup languages can be written. Its major advantage includes:

- Data is self-describing.
- Data can be manipulated with standard tools.
- Data can be viewed with standard tools.
- Different views of the same data are easy to create with style sheets.

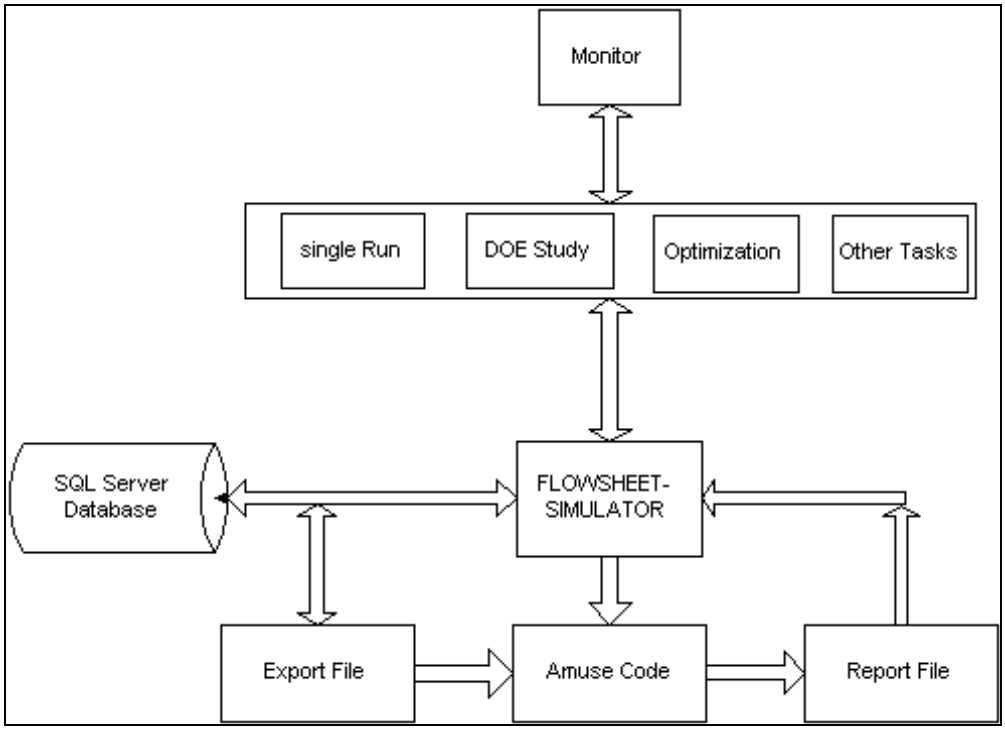


Figure 3 - System Framework for the Engineering Model

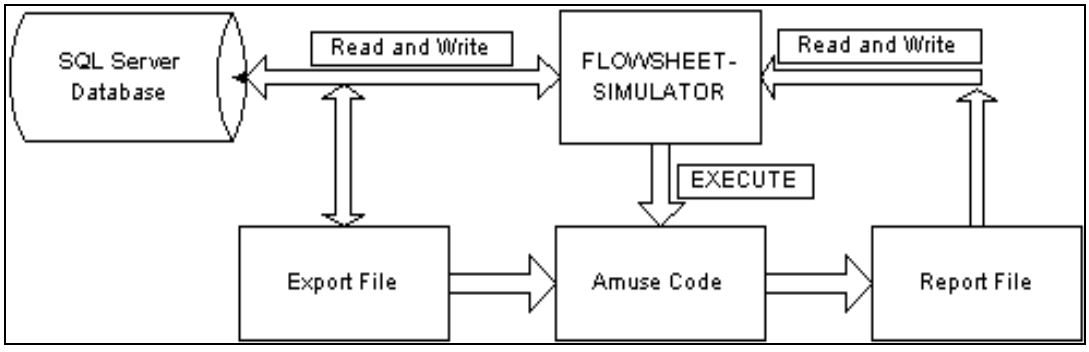


Figure 4 - Relationship between FLOWSHEET-SIMULATOR and AMUSE Code

A representative XML file for flowsheet simulations is shown in Figure 6. Such data sets can be easily manipulated and viewed with a web browser as shown in Figure 7. XML

files form can also be generating a database for reports, parameter configuration, tasks, and manager configuration.

[Note: Figure 5 has been removed from unlimited distribution for proprietary reasons of Applied Technology.]

Figure 5 - Parameter definition table within flowsheet simulator

3.2 Demonstrate Modeling Concept

3.2.1. Demonstration of Design of Experiment (DOE)

DOE refers to any of the many formal methods available for setting parameter values in a set of experiments. The software module for the DOE is under development.

The purpose of the DOE module is to:

- Assess design variable impact,
- Identify significant design variable interactions,
- Analyze a design space and provide an initial estimate for an optimal design,
- Screen broad design variable ranges for an optimal design (design space reduction)

To conduct a DOE study, we need to identify potential parameters, set the number of experiments that can be affordably run, output quantities of interest, and analyze the results to evaluate the study. The techniques for the DOE study include full-factorial, parameter variation, data file, orthogonal arrays, central composite, and Latin hypercubes. A design matrix example is shown in Figure 8. Each row of the matrix constitutes a separate design evaluation and each column represents a different variable.

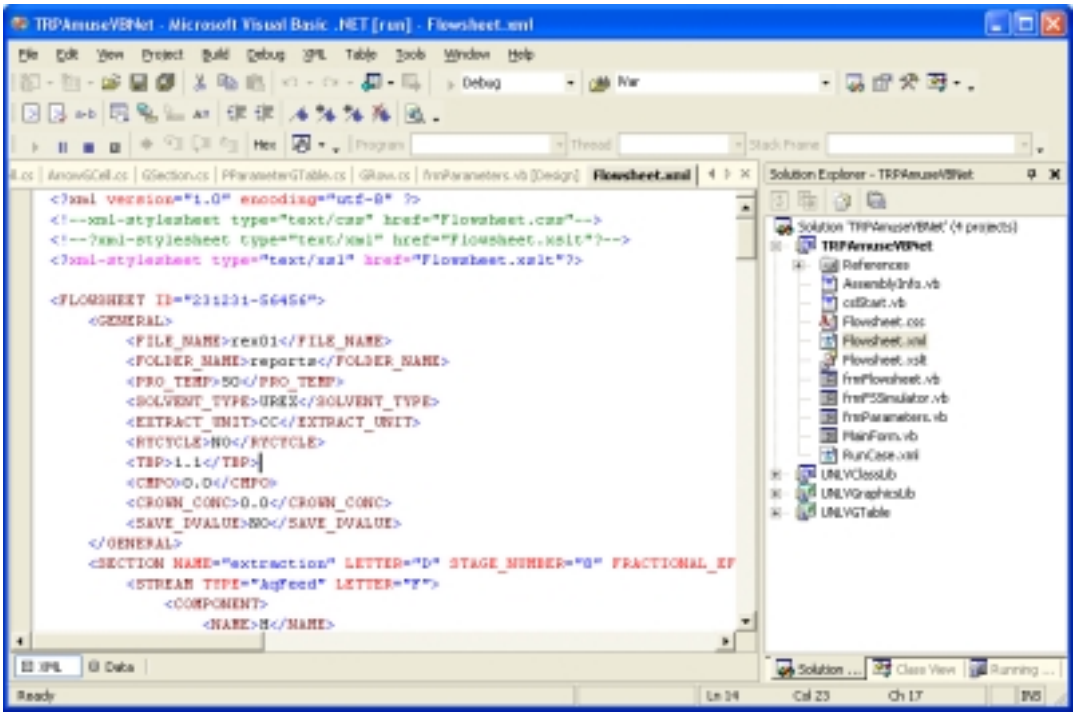
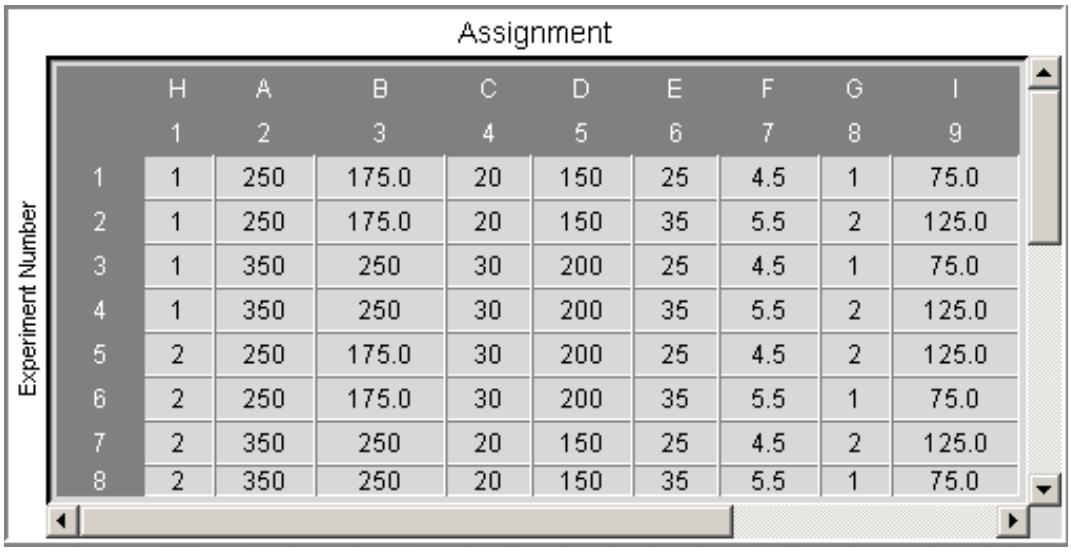


Figure 6 -XML File for flowsheet simulator

[Note: Figure 7 has been removed from unlimited distribution for proprietary reasons of Applied Technology.]

Figure 7 - Visualization of Flowsheet simulation XML file using web browser



	H	A	B	C	D	E	F	G	I
	1	2	3	4	5	6	7	8	9
1	1	250	175.0	20	150	25	4.5	1	75.0
2	1	250	175.0	20	150	35	5.5	2	125.0
3	1	350	250	30	200	25	4.5	1	75.0
4	1	350	250	30	200	35	5.5	2	125.0
5	2	250	175.0	30	200	25	4.5	2	125.0
6	2	250	175.0	30	200	35	5.5	1	75.0
7	2	350	250	20	150	25	4.5	2	125.0
8	2	350	250	20	150	35	5.5	1	75.0

Figure 8 - Design matrix for various experimental scenarios

3.2.2. Demonstration of User Interface and Model Modules

Frequently, a valuable aid in the initial identification of the inputs and outputs for the various subsystems of a given system is to graphically model the overall system, including explicit designation of internal inputs and outputs. During the first year study, we carefully studied and identified various parts of the AMUSE software package. The second year tasks were to create various Microsoft™ Visual Basic components, such as export file generation, report file generation, parameter validation, and to have this software interface between Microsoft ™ Excel, Visual Basic and .Net.

- **Flowsheet Simulator**- as shown in Figure 9, the interface developed with Microsoft ™ .Net can be used for visually defining the UREX process. Within the tool, the user can actively define inputs/outputs for the process module and can be easily modified for adding/deleting sections, editing stage number. The interface also enforces mass balance of the process. Figure 10 shows the export file information and Figure 11 shows the report file information.
- **Real-time Solution Viewer -** Some design exploration packages are "black box" in nature, in which the user needs to wait for the completion of the simulation after starting the program. Typically, this means that any performance analysis or intermediate results cannot be viewed. Currently our system model is developing a "Solution Visualization Module" which allows users to monitor the process in real time. You can visualize the input/output parameters of any design task during the simulation through tabulated and/or graphical presentation (Figures 12 and 13). Meanwhile, all intermediate results will be stored in a Microsoft SQL Server database. Finally, "Crystal Report Module" is integrated into our .Net development component and will be used to create customized report formats from database.

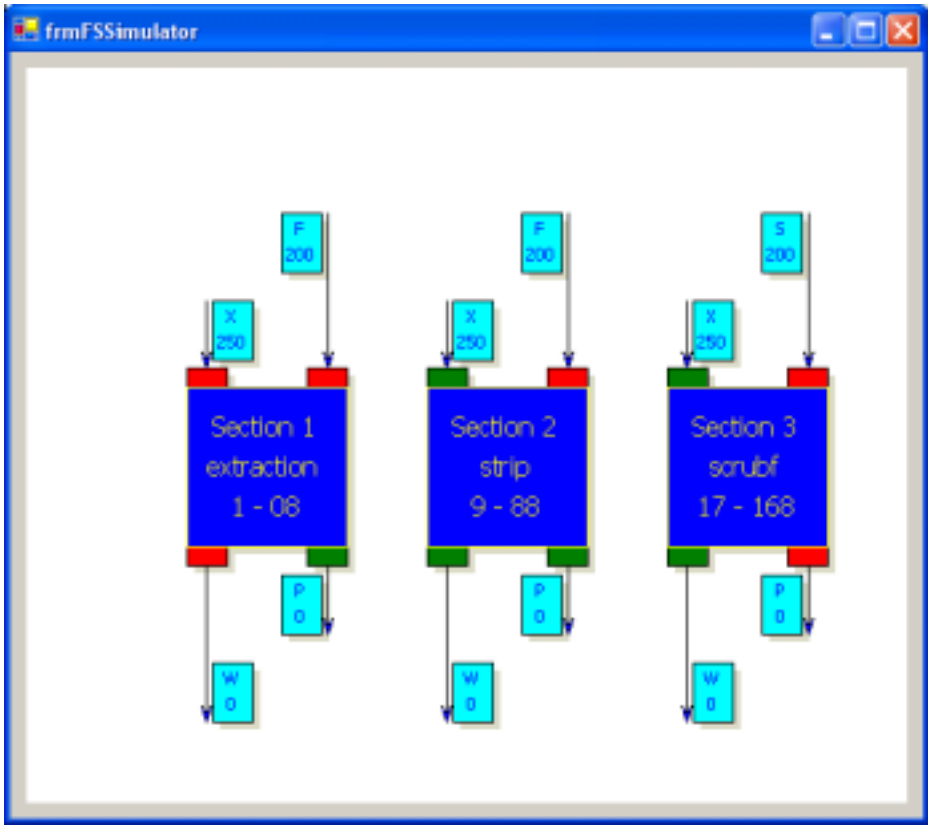


Figure 9 - FLOWSHEET-SIMULATOR for UREX Process

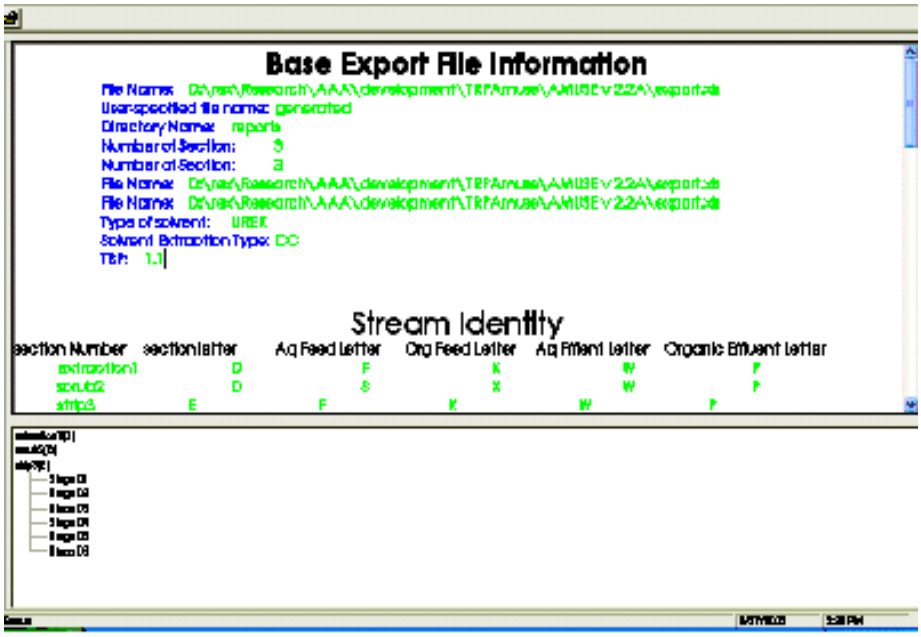


Figure 10 - Export File Information after parsing

[Note: Figure 11 has been removed from unlimited distribution for proprietary reasons of Applied Technology.]

Figure 11 - Report file information generated after AMUSE run

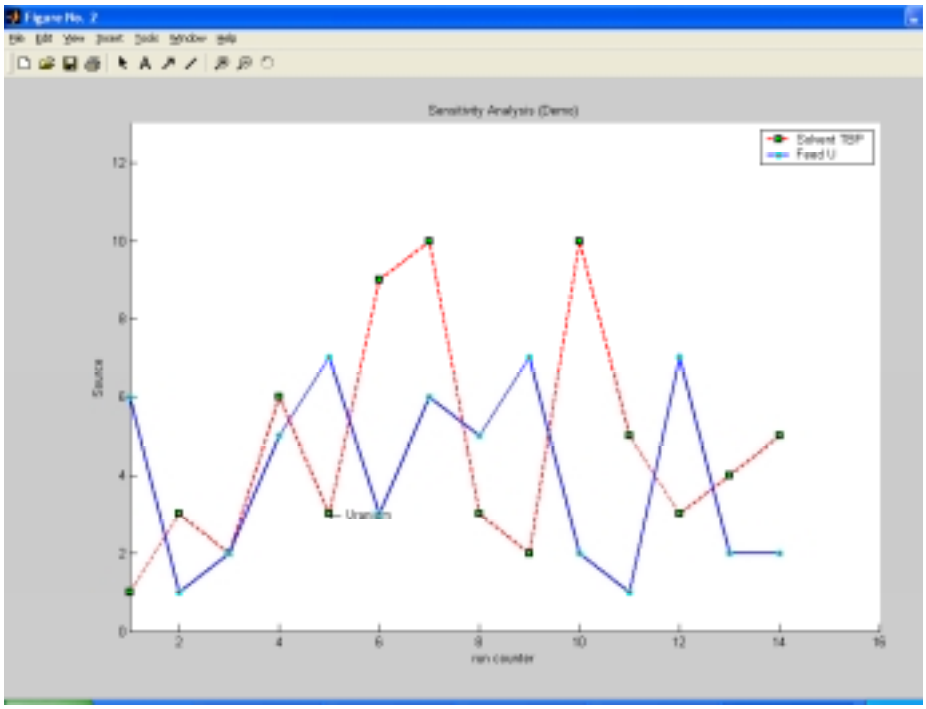


Figure 12 - System sensitivity analysis chart (Line Plot)

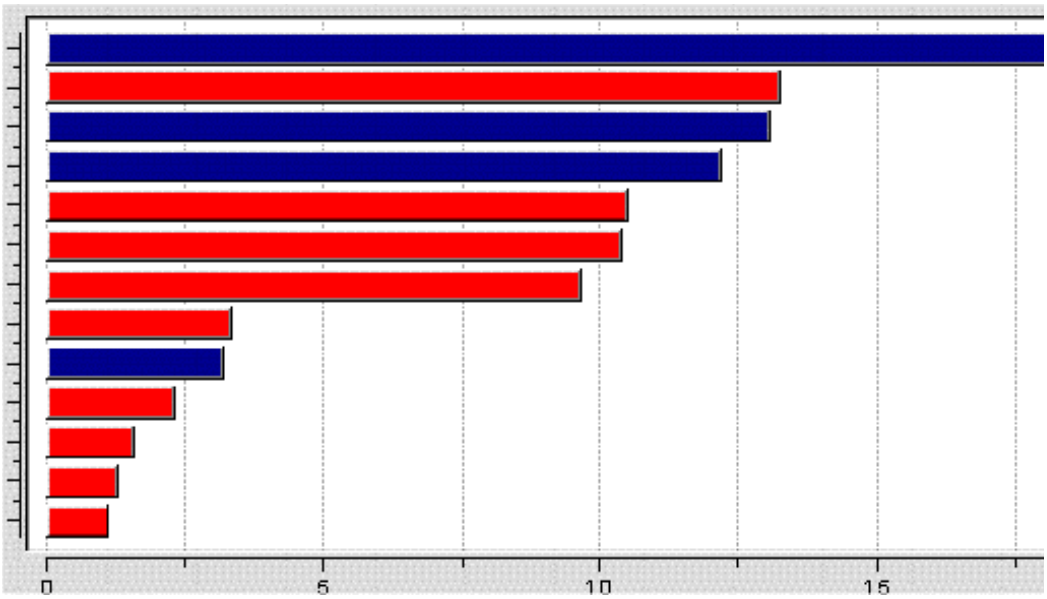


Figure 13 - System sensitivity analysis chart – Bar Plot

3.2.3. Summary of Optimization Strategies

One of the primary tasks of a systems engineer is to ensure the optimization of the design process. Optimization studies for nonlinearly constrained problems (i.e., most complex design problems) have repeatedly shown that no single optimization technique works best for all design problems, and, in most cases, a mix of techniques works better than a single technique for a single design problem. Currently, different optimization techniques have been studied:

- Graphical Optimization
- Linear Programming
- Nonlinear Programming
- Numerical Techniques –
    - Sequential Linear Programming(SLP)
    - Sequential Quadratic Programming(SQP)
    - Generalized Reduced Gradient(GRG) method
    - Sequential Gradient Restoration
- Discrete Optimization
- Multi-objective Optimization

Optimization techniques are used to find a set of design parameters, $x = \{x_1, x_2, \cdots, x_n\}$, that can in some way be defined as optimal. In a simple case this might be the minimization or maximization of some system characteristic that is dependent on x. In a more advanced formulation the objective function, f(x), to be minimized or maximized,

might be subject to constraints in the form of equality constraints, or inequality constraints; and/or parameter bounds.

An efficient and accurate solution to one design problem depends not only on the size of the problem in terms of the number of constraints and design variables, but also on characteristics of the objective function and constraints. When both the objective function and the constraints are linear functions of the design variable, the problem is known as a Linear Programming (LP) problem. Quadratic Programming (QP) concerns the minimization or maximization of a quadratic objective function that is linearly constrained. For both the LP and QP problems, reliable solution procedures are readily available. More difficult to solve is the Nonlinear Programming (NP) problem in which the objective function and constraints can be nonlinear functions of the design variables. A solution of the NP problem generally requires an iterative procedure to establish a search direction for each iteration. This is usually achieved by the solution of an LP, a QP, or an unconstrained sub problem.

Currently, the optimization code is under development and relies on Microsoft ™ .Net and the MATLAB Simulation toolbox. To demonstrate the optimization, capabilities we used MATLAB as shown in Figure 14.

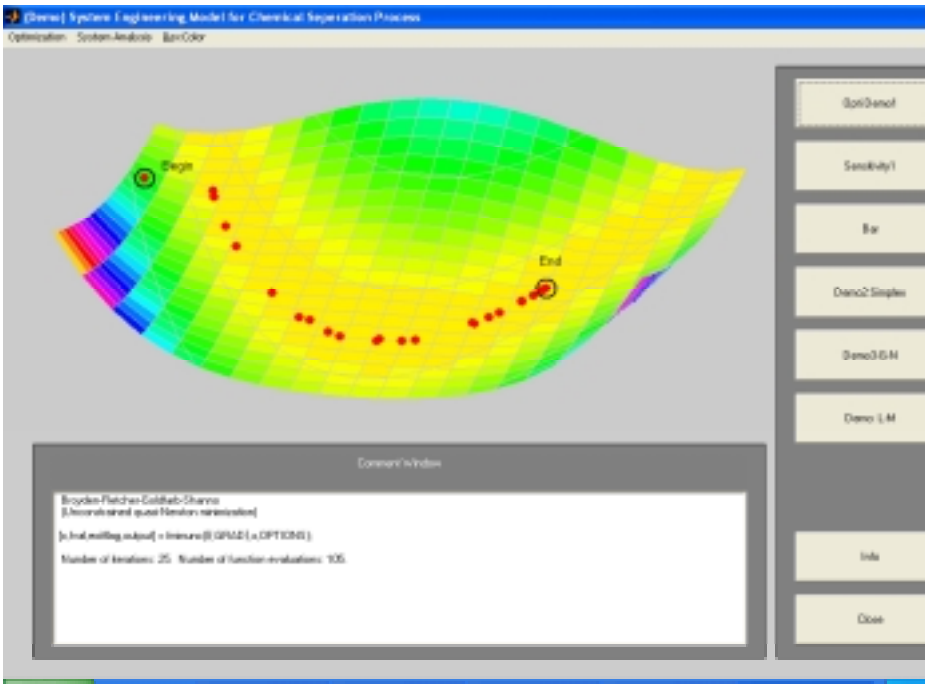


Figure 14 - MATLAB program demonstration for optimization

4. Process – Modernization of AMUSE

The AMUSE (**A**rgonne **M**odel for **U**niversal **S**olvent **E**xtraction) code is a software package developed by Argonne National Laboratory for the analysis of a Generic TRUEX process. The TRUEX process is a solvent extraction process capable of separating small quantities of transuranic elements (for example; Np, Am, Pu, and Cm) from aqueous nitrate and chloride solutions. These types of chemical streams are typically generated in reprocessing plant operations or in plutonium production and purification processes.

UNLV's role is to develop a general user interface to aid researchers in analyzing different process scenarios. In addition, UNLV researchers will incorporate the AMUSE Code into the overall systems engineering model. This will be a key component to the overall analysis of the process.

4.1 Review/Analyze Code Structure

Obtain the appropriate files from ANL and study the layout and development history of the code. Understand the code and analyze the data flow through the package to determine how the different process steps are included and how the calculations are performed. The code has been reviewed and analyzed thoroughly. Based on the working code, a new GUI has been designed to be compatible with AMUSE software. A significant amount of effort has been spent on insuring proper data storage and retrieval within the existing AMUSE environment.

4.2 Develop Verification Plan

The AMUSE code is being changed to include new features. Every time the changes are implemented, the code has to be verified numerically in order to check the compatibility with the newly developed GUI. Sets of test problems are chosen to demonstrate the numerical accuracy of the actual software changes. This task continues in year two because it is critical that all changes to the software be verified. The testing of both the AMUSE code and the GUI will require a significant amount of analysis by the researches to insure the results are accurate.

4.3 Modify/Improve Software

The GUI is begin developed through an object oriented programming (OOP) approach. This is critical to ANL as they can use the new GUI objects for future modifications to the code. The OOP scheme shown in Figure 15 has been selected for further software development. A drag and drop mechanism is implemented for all the objects in the GUI. The new interface is more user friendly than the one created in the beginning. A database approach is introduced in the interface for easy data transfer between AMUSE and the new GUI.

Figure 16 is the main form for creating the flowsheet. A flowsheet object is created which will appear on the left hand side of the form. A click on this object allows the user to select the type of process. Figure 17 shows the flow sheet object form. The user can select the type of process, like UREX/PUREX/TRUEX, etc., from the process option list. Process variables are also set and selected for input. A flow sheet of the process is also displayed to the right side that will be useful for novice users. The user can specify the location to where all the output should be

stored. A click on the next button takes user back to main form to design the flow sheet. Four objects are created in the tools section of the main page. This is used to create the flow sheet. Generally each flowsheet has three basic sections: First, Last and Intermediate sections. A process with a single section object can also be created.

A Drag and Drop mechanism is provided for all the tool objects. The user can drag the section objects from the tools menu and drop them on the main form as shown in Figure 18. These section objects can be resized. The user can define the input by clicking on the objects within the section object (Figure 19). In a similar way input is define to organic and/or aqueous feeds as shown in Figure 20. Stream names are also defined in the new interface. This an option does not exist in the current AMUSE interface. This allows the user to identify the streams easily. Input for Aqueous and Organic streams are defined as shown in Figure 21. The user can do stage sampling or give stage specific input by clicking on the stage object as shown in Figure 22.

Finally a database is used for easy data transfer between AMUSE and new Interface. The database is created by using SQL server. Four tables are created to support all the data (Figure 23). The data from the new Interface goes to the database. An export file is then created from the database, which is in an Excel file format. This serves as the input to AMUSE. After this, the AMUSE macro “BEGIN.GTM” is called from the new interface to perform the calculations. The results can be viewed from the user specified folder.

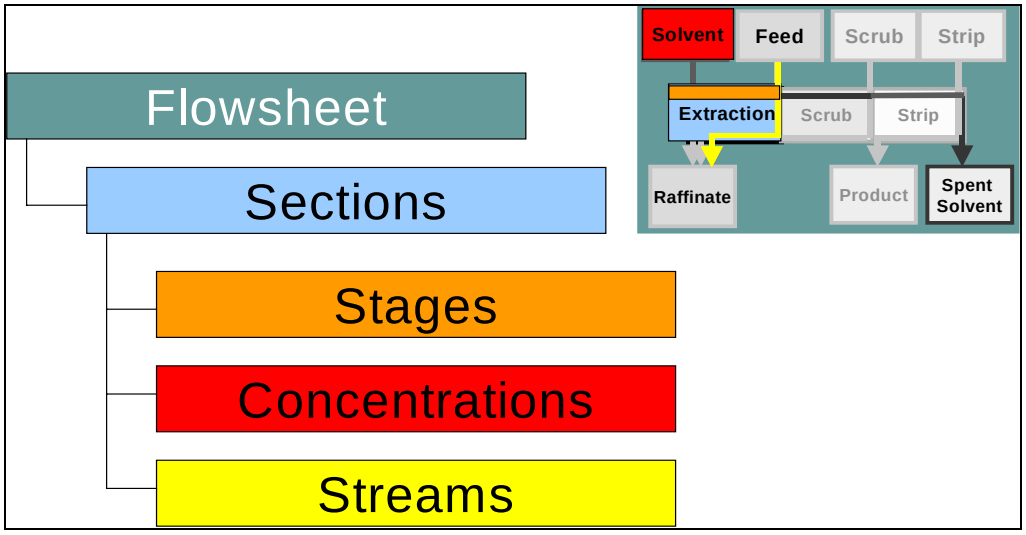


Figure 15 - Object-oriented programming approach for the new interface design

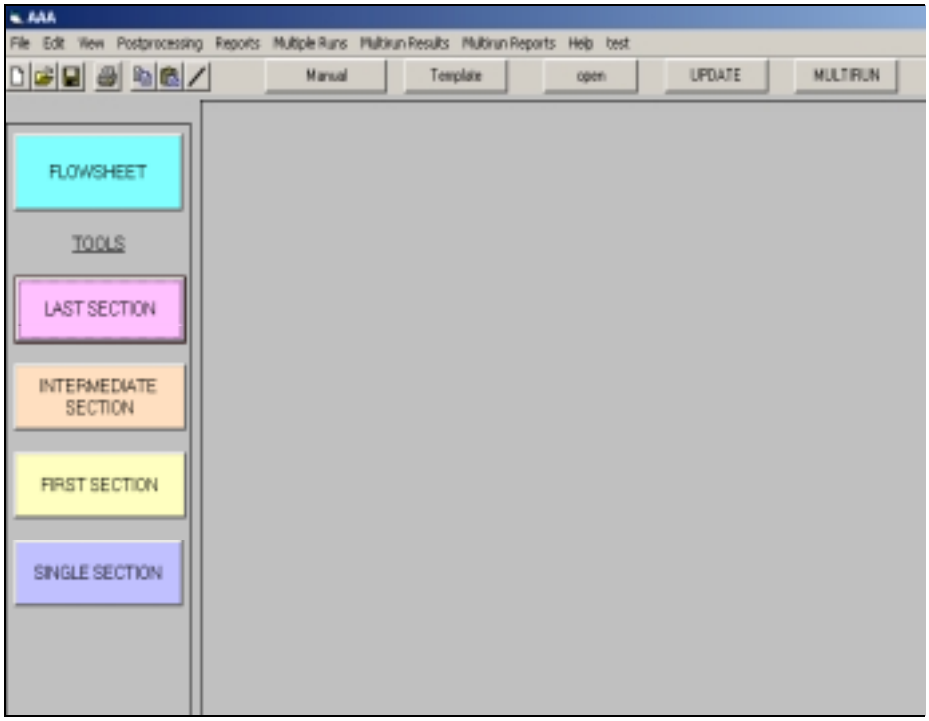


Figure 16 - Main Form for Flowsheet creation

[Note: Figure 17 has been removed from unlimited distribution for proprietary reasons of Applied Technology.]

Figure 17 - Process selection form used for creating flowsheet creation

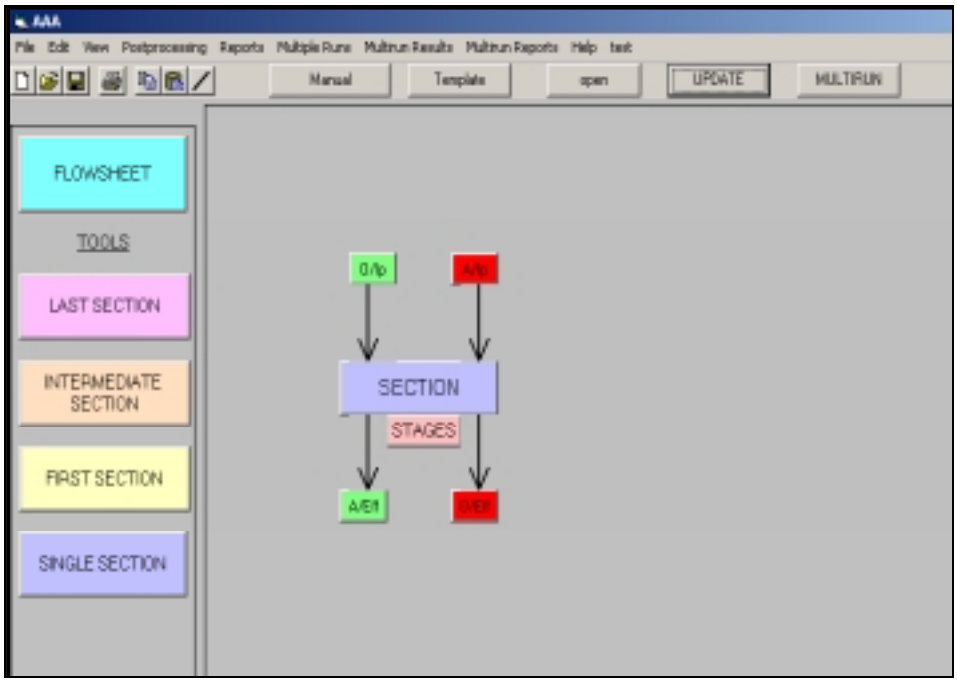


Figure 18 - Initial section object created from drag and drop

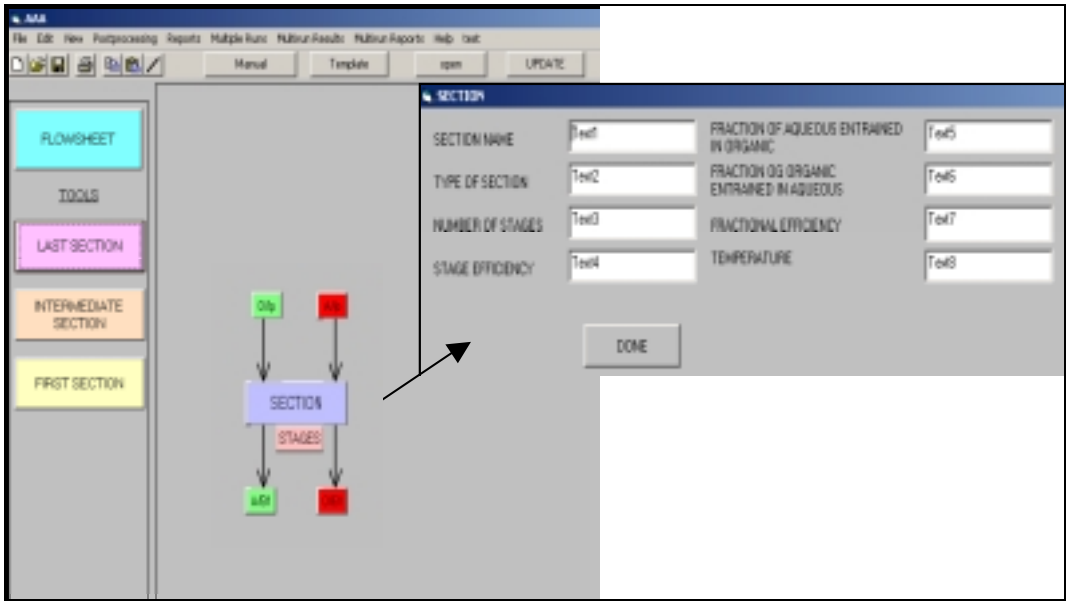


Figure 19 - Each section object with all the parameter properties

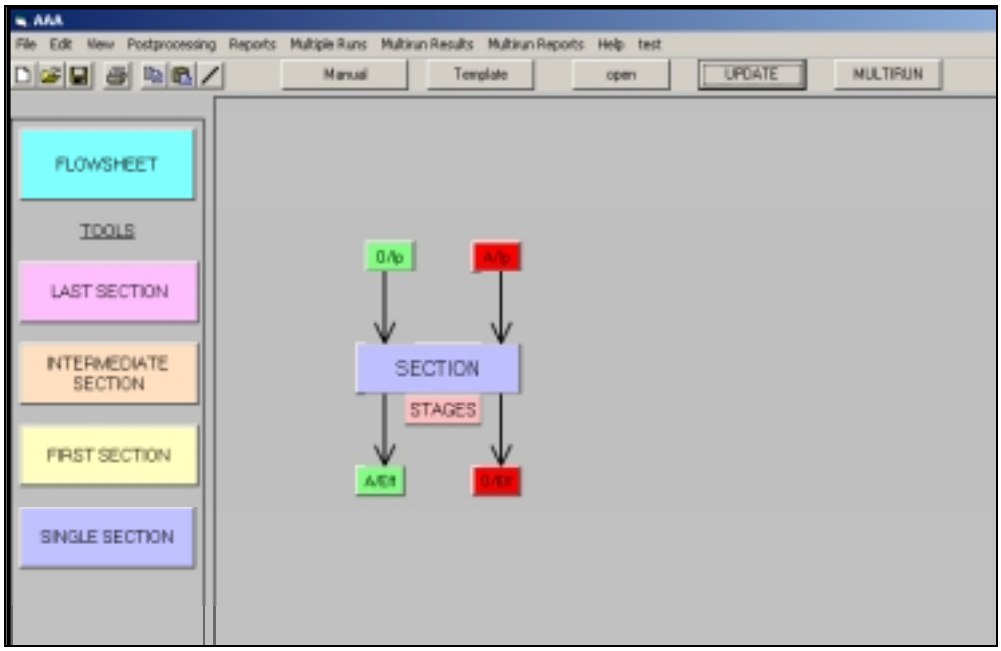


Figure 20 - Interface for the organic and aqueous feeds

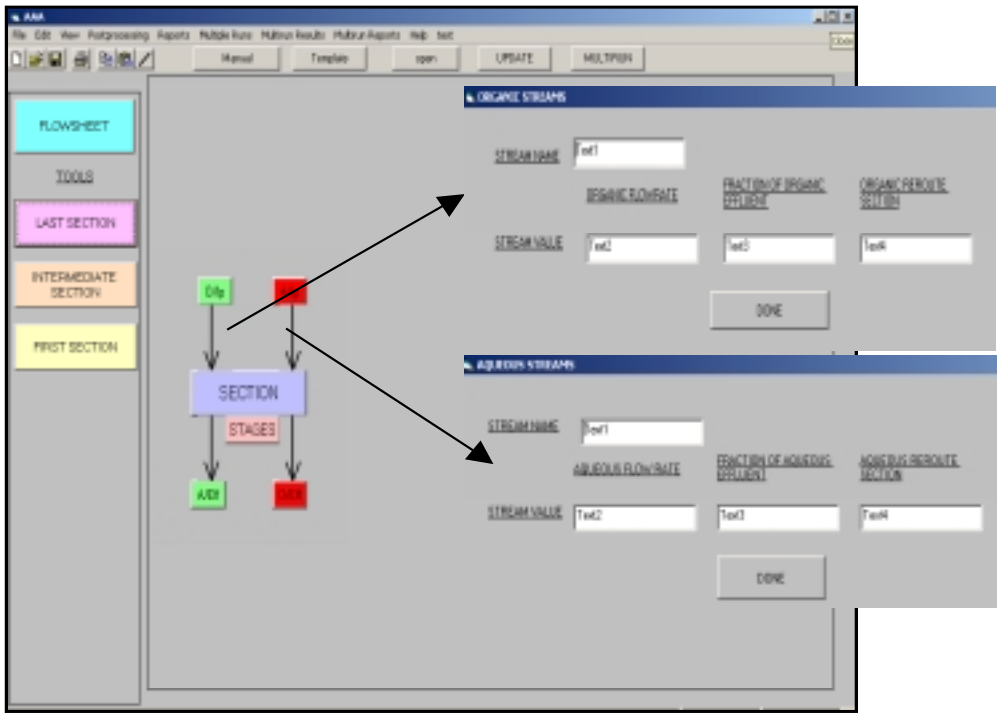


Figure 21 - Aqueous and organic stream inputs

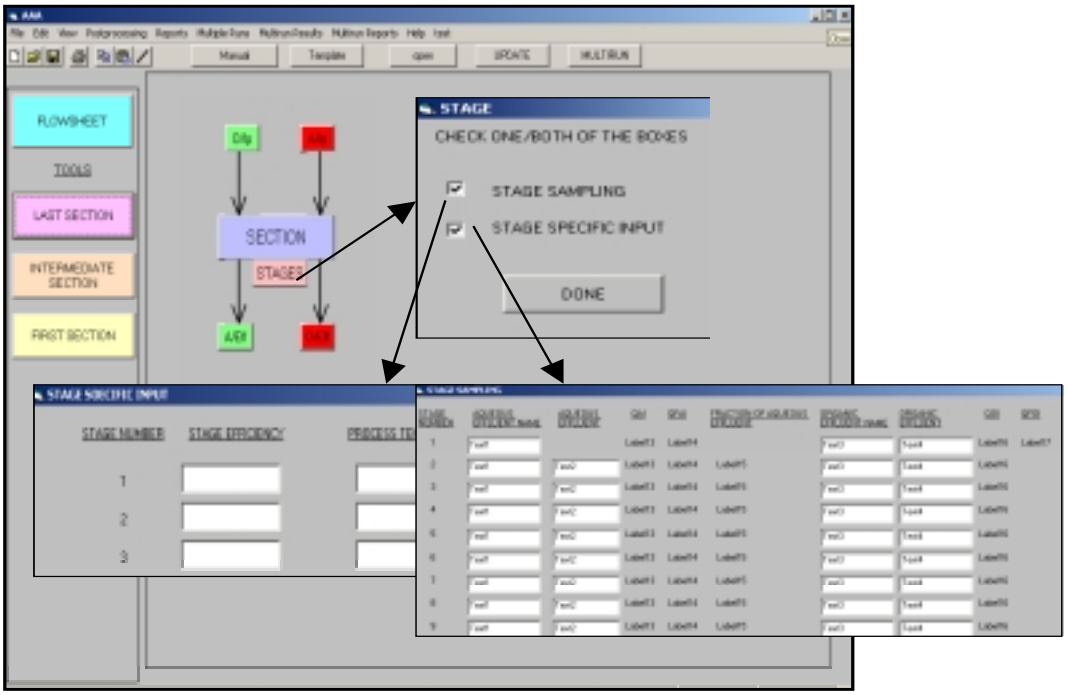


Figure 22 - Stage sampling objects according to the selections

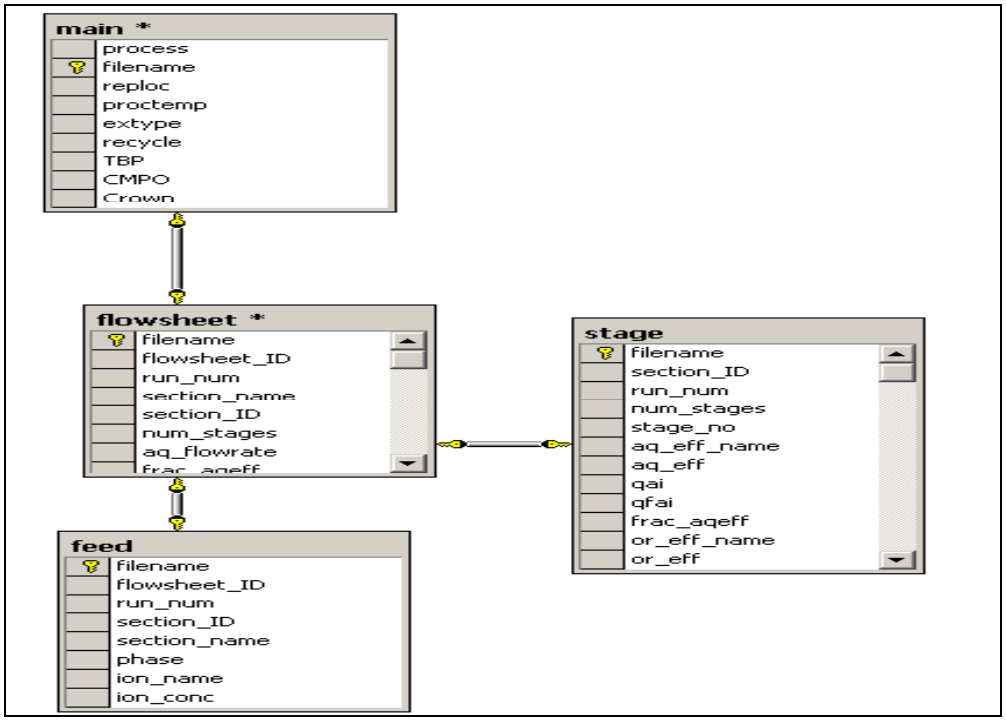


Figure 23 - Table structure for the new interface

5. Summary

A framework and environment for a systems engineering analysis of the chemical separations system has been developed. A baseline systems engineering model from which modifications and improvement can be made has also been studied. The long-term needs for a systems engineering model of the fuel processing included mass balances, system control, plant layout and design, and other features will be discussed with ANL-East engineers and scientists.

The AMUSE code has been studied and analyzed thoroughly. A new interface that takes into consideration an object oriented programming (OOP) concept is under development. A database concept has also been included in design process that provides for more efficient data manipulation and report compilation. The interface is designed in such a way that future changes in the software can be made without significant code rewriting. The new interface is user-friendly that promotes a more intuitive user interaction. All the operations that are performed in the AMUSE code are implemented in the new interface. The output from both the interfaces is checked and verified. This interface integrates with the systems engineering model to optimize the flow sheet parameters. Reports can be generated from the database based on specific task or objective.

Significant progress has been made for the development of a systems engineering model. The framework for the new software system has been constructed and the implementation is ongoing. Complete operation of the AMUSE Code has been implemented by the development of a flowsheet simulator environment. In support of this effort, a Parameter Definition module has been developed. All the parameters are shared data between the data module, simulation code and all other tasks. Modules for parameter and problem definition are a key bridge to connect the simulation code with all tasks. The Problem Definition module has been designed and is continuously being updated. Major data manipulation and query functions were accomplished through the design and implementation of a SQL Server database approach. Various optimization technologies have been studied and algorithms for different optimization approaches are under development. An XML file format has been carefully selected and applied as the main file format for this system to accommodate heavy data communication and data transfer. An OOP design approach for this software system plays an important role to apply proper data structures and object hierarchies in order to seamlessly connect all the individual modules into an integrated system.

The final product of this research will provide engineers and scientists a friendly and useful Window's GUI package that can simulate the chemical separation process over a wide range of different input conditions for the actinides and solvents, which includes the required separation efficiencies of various equipment components.

6. References

Blanchard, B. S. and Fabrycky, W. J., "Systems Engineering and Analysis," 3rd Edition, Prentice-Hall, Inc., 1998

Chapman , William L., Bahill , A. Terry, A., and Wymore, Wayne, " Engineering Modeling and Design," CRC Press, 1992.

iSight User's Guide 6.0 and Reference Guide, Engineous Software Inc.,2001

Regalbuto, M. C., Misra, B., Chamberlain, D. B., Leonard, R. A., and Vadegrift, G. F., "The Monitoring and Control of TRUEX Processes. Volume One – The Use of Sensitivity Analysis to Determine Key Process Variables and Their Control Bound," Chemical Technology Division, ANL-92/7 Report, Argonne National Labortory.
Andrew P. Sage and James E. Armstrong Jr., "Introduction to Systems Engineering", ISBN 0-471-02766-9, 2000.

Ding-Zhu Du, Panos M. Pardalos and Weili Wu, "Mathematical Theory of Optimization", Kluwer Academic Publishers, 2001.

Arthur E. Bryson, Jr., "Dynamic Optimization", ISBN 0-201-36187-6, 1999.

Proceeding of The IEEE International Conference on Systems Engineering, August 24-26, 1989, Dayton, Ohio.

Bernhard Thome, "Systems Engineering", ISBN 0-471-93552-2, 1993.

Krishna K.Singh and Gayatri Agnihotri, "System Design through MATLAB™, Control Toolbox and SIMULINK™", ISBN 1-85233-337-5, 2001.

Willis, Thearon, Visual Basic SQL Server 7.0