DEVELOPMENT OF TECHNOLOGIES AND ANALYTICAL CAPABILITIES FOR VISION 21 ENERGY PLANTS

FINAL TECHNICAL REPORT

COOPERATIVE AGREEMENT DE-FC26-00NT40954
OCTOBER 2000 – DECEMBER 2004

Madhava Syamlal, Maxwell Osawe, Stephen Zitney, Lewis Collins
Fluent Inc.

David Sloan and Woodrow Fiveland
ALSTOM Power

Frank Joop and Philip Simon
Intergraph Corporation

K. Joseph Cleetus
West Virginia University

April 2005

Fluent Inc.
10 Cavendish Court, Lebanon, NH 03766

ALSTOM Power
US Power Plant Laboratories, 2000 Day Hill Road, Windsor, CT 06095

Aspen Technology, Inc.
Ten Canal Park, Cambridge, MA 02141

Intergraph Corporation
One Madison Industrial Estate, Huntsville, AL 35894

Concurrent Engineering Research Center, West Virginia University
886 Chestnut Ridge Rd., Morgantown, WV 26506

DISCLAIMER

ACKNOWLEDGEMENTS

Financial support for this project was provided by DOE National Energy Technology Laboratory (Cooperative Agreement No. DE-FC26-00NT46954) and by the project participants Fluent Inc., ALSTOM Power Inc., Aspen Technology, Inc., Intergraph Corporation, and West Virginia University. In addition to the authors, the contributions of many advisors and reviewers at the sponsoring and participating organizations are gratefully acknowledged. Special thanks are due to the individuals from the following organizations who attended one or more of the project Advisory Board meetings and provided valuable technical guidance to the team:

- Air Products
- Clean Energy Systems
- DOE/NETL
- Dow Chemical
- Dupont
- Fuel Cell Energy
- Hatch Associates
- Intelligent Light
- Intergraph
- K-S Tech Ltd
- Proctor & Gamble
- UOP

1 ABSTRACT

To accelerate the development of advanced power plants, DOE's Vision 21 program identified the need for an integrated suite of software tools that could be used to simulate and visualize new plant concepts. Existing process simulation software did not meet this objective of *virtual-plant simulation*. Sophisticated models of many individual equipment items are available; however, a seamless coupling capability that would integrate the advanced equipment (component) models to the process (system) simulation software remained to be developed. The inability to use models in an integrated manner causes *knowledge loss* (e.g., knowledge captured in detailed equipment models is usually not available in process simulation) and modeling inconsistencies (e.g., physical properties and reaction kinetics data in different models are not the same). A team consisting of Fluent Inc., ALSTOM Power Inc., Aspen Technology Inc., Intergraph Corporation, and West Virginia University, in collaboration with the National Energy Technology Laboratory (NETL), addressed this challenge in a project performed over the period from October 2000 through December 2004.

In this project the integration of the cycle analysis software was based on widely used commercial software: Aspen Plus® for process simulation and FLUENT® for computational fluid dynamics (CFD) modeling of equipment items. The integration software was designed to also include custom (in-house, proprietary, legacy) equipment models that often encapsulate the experience from the many years of designing and operating the equipment. The team adopted CAPE-OPEN (CO) interfaces, the *de facto* international standard for communication among process models, for exchanging information between software. The software developed in this project is the first demonstration of the use of CO interfaces to link CFD and custom equipment models with process simulators. New interface requirements identified during this project were communicated to the CO standard developers.

The new software capability was designed to make the construction of integrated models fast and integrated simulations robust and user-friendly. Configuration wizards were developed to make CFD and custom models CO-compliant. An Integration Controller and CFD Model Database were developed to facilitate the exchange of information between equipment and process models. A reduced order model (ROM) framework and a solution strategy capability were incorporated in the Integration Controller to enable a flexible trade-off between simulation speed and complexity. A CFD viewer was developed so that process engineers can view CFD results from the process simulator interface.

For demonstrating the capability of the integrated software suite, we first conducted simulations of (1) the conventional steam cycle at a 30 MWe coal-fired power plant for municipal electricity generation, and (2) a 270 MWe, natural gas-fired, combined cycle power plant. Although these are not advanced power plants, they embody features that a virtual-plant simulator would be required to represent. Three runs were completed for each demonstration case: (1) an initial baseline run using the existing component libraries in Aspen Plus, (2) a second run where one of the library components was replaced with an ALSTOM Power proprietary code, and (3) a third run where a cycle component was replaced with a FLUENT CFD model. Both sets of the three runs were successfully

completed over a range of loads. Subsequently, we completed a third demonstration case -- a simulation of a 250MWe FutureGen IGCC power plant, in which a cycle component was replaced with a FLUENT CFD model. Both Demonstration Cases 2 and 3 (coupled with FLUENT) were run over a Local Area Network to demonstrate the distributed-computing capability that was developed.

The software development effort was continually guided by the end-user requirements. The primary means of ensuring this guidance was by having the team member ALSTOM Power participate in software design review meetings and conduct demonstration simulations during the development phase. In addition, the project team sought the advice of engineers from NETL, other Vision 21 program participants, and representatives from other power and chemical/process companies. At the start of the project, a survey was conducted to identify user requirements. During the course of the project, semiannual Advisory Board meetings were conducted to review the progress, demonstrate the current prototype version of the software, and collect feedback from the industrial participants. The comments and suggestions from the potential end-users was used to adjust the software design and development plans.

By basing the development on a foundation of existing commercial software Aspen Plus and FLUENT, we have ensured that the technology is immediately available for use by US industry and will remain supported in the future. The use of the open standard CO will allow other companies to plug their CO-compliant models into the integrated software suite. We have already demonstrated that the integration software works with another commercial simulation executive, HYSYS®. The integration software was made available to industry in November 2003 and won the 2004 R&D100 award as " … one of the 100 most technologically significant products introduced into the marketplace over the past year". The software developed through this project enables designers to conduct process simulations with a simultaneous scope and accuracy never before possible. The deep insight from such simulations will help designers to identify opportunities for achieving unprecedented high efficiency and near-zero emissions in advanced power plants.

TABLE OF CONTENTS

1 ABSTRACT.. ii
2 Introduction.. 1
2.1 Motivation.. 1
2.2 Model reuse.. 2
2.3 Statement of project goals.. 4
2.4 Structure of the report 4
3 Executive Summary 6
4 Experimental 10
5 Software Development and Deployment 11
5.1 Software development 11
5.1.1 Development process 11
5.1.2 CAPE-OPEN standard 11
5.1.3 Architecture/Components 13
5.1.3.1 FLUENT Configuration Wizard 15
5.1.3.2 Custom Model Configuration Wizard 16
5.1.3.3 Controller COM-CORBA Bridge 16
5.1.3.4 EM Database 20
5.1.3.5 Solution Strategy.. 22
5.1.3.6 Reduced-Order Model 23
5.1.3.7 Transfer of physical properties 27
5.1.3.8 Remote execution... 32
5.1.3.9 CFD Viewer 34
5.1.3.10 Proprietary Model Wrapper 35
5.2 Excel spreadsheet integration 36
5.3 Error detection, reporting, and recovery 37
5.4 Software deployment 40
6 Integrated Model Development and Usage............................ 41
6.1 Example 1: Reaction-separation-recycle flowsheet.............. 44
6.1.1 System-level model development 44
6.1.2 Equipment-level model development 45
6.1.3 Equipment-level model configuration 46
6.1.4 Model integration.. 52
6.1.4.1 CFD model selection and edit... 52
6.1.4.2 Running the integrated model... 55
6.1.4.3 CFD Viewer 55
6.1.4.4 Aspen Plus analysis tools.. 56
6.1.4.4.1 Sensitivity analysis.. 56
6.1.4.4.2 Optimization 58
6.2 Example 2: Fuel cell flowsheet with reformer CFD model.......... 59
6.2.1 System-level model development 59
6.2.2 Equipment-level model development 61
6.2.3 Equipment-level model configuration 62
6.2.4 Model integration.. 63
6.2.4.1 Solution strategy 63

6.2.4.2 Running the integrated simulation 65
6.3 Vision of organizational usage 67
7 Application of Integrated Modeling to Power Plant Simulations. 69
7.1 Demonstration Case 1: Conventional Steam Cycle 70
7.1.1 Case 1: Run 2 – Aspen Plus Coupled with BPS 71
7.1.2 Case 1 : Run 3 – Aspen Plus Coupled with FLUENT 74
7.2 Demonstration Case 2: Natural Gas Combined Cycle 77
7.2.1 Case 2: Run 2 – Aspen Plus Coupled With HRSGPS 78
7.2.2 Case 2 : Run 3 – Aspen Plus Coupled with FLUENT 80
7.3 Demonstration Case 3: Integrated Gasification Combined Cycle 82
8 Results and Discussion 85
8.1 Validation of project goals 85
8.2 Benefits to DOE and Industry 85
9 Technology Outreach 87
9.1 Integrating physical plant knowledge 87
9.1.1 Scope 87
9.1.2 Concept studies 87
9.1.3 Conclusions and recommendations 89
9.2 Advisory Board 89
10 Conclusions 92
10.1 Program relevancy 92
10.2 Future work 92
11 References 95
12 Bibliography 96
13 List of Acronyms and Abbreviations 99
14 Appendices 101
14.1 Example of a Model XML File 101
14.2 Example of a pre-computed results file 103

LIST OF FIGURES

Figure 5.1: Three-tier architecture system of the integrated software 14
Figure 5.2: Diagrams showing the COM components implemented in the Integration Controller subsystem 18
Figure 5.3: Illustration of the Bridge design pattern implementation in the integrated simulation environment 19
Figure 5.4: Organization of Models in the Database 21
Figure 5.5: Solution Strategy tab in the Model Edit GUI 22
Figure 5.6: Comparison of the variation of product purity with shaft speed using FLUENT and regression model... 24
Figure 5.7: Comparison of the variation of product yield with shaft speed using FLUENT and Regression models.. 25
Figure 5.8: Dialog showing the option to save pre-computed FLUENT results to the database 26
Figure 5.9: Dialog showing the use of the wizard to configure a Regression model 27
Figure 5.10: Comparison of the calculated specific heat of hydrogen versus temperature, using Aspen Plus and FLUENT CO wrapper.. 28
Figure 5.11: Comparison of the calculated thermal conductivity of hydrogen versus temperature, using Aspen Plus and FLUENT CO wrapper.................................... 28
Figure 5.12: Comparison of the calculated viscosity of hydrogen versus temperature, using Aspen Plus and FLUENT CO wrapper.. 29
Figure 5.13: Comparison of the calculated density of hydrogen versus temperature, using Aspen Plus and FLUENT CO wrapper 29
Figure 5.14: Comparison of the calculated specific heat of liquid water versus temperature, using Aspen Plus and FLUENT CO wrapper.................................... 30
Figure 5.15: Comparison of the calculated thermal conductivity of liquid water versus temperature, using Aspen Plus and FLUENT CO wrapper.................................... 30
Figure 5.16: Comparison of the calculated viscosity of liquid water versus temperature, using Aspen Plus and FLUENT CO wrapper.. 31
Figure 5.17: Comparison of the calculated density of liquid water versus temperature, using Aspen Plus and FLUENT CO wrapper.. 31
Figure 5.18: Dialog showing the specification of server address in the Solution Strategy tab of the Model Edit GUI.. 33
Figure 5.19: Excel spreadsheet showing the inputs and results of running the FLUENT 2-D CSTR test problem 37
Figure 5.20: Sample error dialog reporting the non-availability of the FLUENT case/data files as specified in the solver XML file.. 38
Figure 5.21: Error dialog reporting unavailability of the remote solver process, which is encapsulated in the ICapeUnit CORBA object reference 38
Figure 5.22: Error dialog reporting a bootstrapping problem due to unavailability of the FLUENT launcher, which is registered with the Implementation Repository 39
Figure 6.1: The workflow in integrating CFD and process models................................ 43
Figure 6.2: Aspen Plus flowsheet for reaction-separation-recycle example.................... 44
Figure 6.3: Grid display for FLUENT CSTR model 45
Figure 6.4: Navigator panel for the FLUENT Configuration Wizard 46

Figure 6.5: Model Files panel in the FLUENT Configuration Wizard 47
Figure 6.6: Model Information panel in the FLUENT Configuration Wizard 47
Figure 6.7: Port Selection panel in the FLUENT Configuration Wizard 48
Figure 6.8: Specification of a heat exchanger coolant fluid using the physical model ports functionality 49
Figure 6.9 Port-to-domain assignment panel in the FLUENT Configuration Wizard 49
Figure 6.10: Basic Parameters for physical property transfer from Aspen Plus to FLUENT 50
Figure 6.11: Basic Parameters for shaft speed in FLUENT Configuration Wizard 51
Figure 6.12: Advanced Parameters panel in FLUENT Configuration Wizard.............. 51
Figure 6.13: Finish panel in FLUENT Configuration Wizard....................................... 52
Figure 6.14: Load and place CAPE-OPEN CFD block on process flowsheet 53
Figure 6.15: Model Selection GUI.. 54
Figure 6.16: Model Edit GUI.. 54
Figure 6.17: CFD Viewer showing contours of mass fraction of an impurity (left) and velocity magnitude (right) in CSTR 56
Figure 6.18: Selection of CAPE-OPEN CFD parameter (shaft speed) as a manipulated variable in an Aspen Plus sensitivity analysis 57
Figure 6.19: Sensitivity analysis for product yield vs. shaft speed................................ 57
Figure 6.20: Sensitivity analysis for product purity vs. shaft speed 58
Figure 6.21: Selection of CAPE-OPEN CFD parameter (shaft speed) as a manipulated variable in an Aspen Plus optimization 58
Figure 6.22: Definition of product yield as the variable to be maximized in an Aspen Plus optimization 59
Figure 6.23: Aspen Plus flowsheet for fuel cell system example 60
Figure 6.24: Port Selection for reformer with two inlets and two outlets....................... 62
Figure 6.25: Domain 1 representing the hot gas (shell side) of the reformer 63
Figure 6.26: Domain 2 representing the process (tube side) of the reformer 63
Figure 6.27: Default Solution Strategy form on the Model Edit GUI 64
Figure 6.28: Hybrid solution strategy using a ROM and FLUENT 64
Figure 6.29: Temperature distribution in the reformer calculated with CFD 65
Figure 7.1: Aspen Plus Steam Cycle Model for Demonstration Case 1 72
Figure 7.2: Software Controller Configuration for the Proprietary BPS Code............... 72
Figure 7.3: Control Parameters and Results for Case 1 with BPS-Aspen Coupling 74
Figure 7.4: Case 1: Run 2 Results – Steam, Fuel, and Air Flow Rates 74
Figure 7.5: Coupled FLUENT and Aspen Plus Cycle for Demonstration Case 1........... 75
Figure 7.6: Convergence of Superheat Outlet Temperature for Case 1 77
Figure 7.7: Portion of Cycle Replaced by the HRSGPS Code 78
Figure 7.8: Parameter Exchanges Between HRSGPS Code and the Wrapper 79
Figure 7.9: Case 2 : Run 2 Results – Steam, Fuel, and Air Flow Rates. 80
Figure 7.10: Case 2 : Run 3 Solution Strategy... 81
Figure 7.11: Case 2: Run 3 Convergence Characteristics... 82
Figure 7.12: Case 3 control schematic 84
Figure 10.1: Virtual-plant simulator roadmap 93

LIST OF TABLES

Table 2.1: Comparison of Process Simulation and CFD models 3
Table 6.1: Information transferred between Aspen Plus and FLUENT models 41
Table 6.2: Key tasks for integrated model development 42
Table 6.3: Integrated workflow using Integration Controller 43
Table 6.4: Reformer outlet conditions 66
Table 6.5: Shell outlet conditions 66
Table 7.1: Summary of Demonstration Case Runs 70
Table 8.1: Features of the integration toolkit and the benefit to end-user 85
Table 9.1: Summary of Advisory Board comments and responses 90
Table 10.1: Fulfillment of Vision 21 Program Plan Objectives 92

2 Introduction

U.S. Department of Energy's (DOE) Vision 21 program is designed to allow integration of advanced power generation and chemical production technologies into systems that achieve the stringent performance, efficiency, and pollutant minimization goals for the future. These individual power, chemical, and fuel-conversion technologies are called *technology modules,* implying the ability to interchange and combine them to form the complete Vision 21 plant that achieves the needed level of performance at affordable costs. To design a Vision 21 plant, DOE envisions an integrated software capability called *virtual-plant simulation* that includes sub-models for components and subsystems, dynamic response and process control, and visualization. With this toolkit, the knowledge about the process and the physical plant captured in computer models of the modules can be linked together to simulate the entire Vision 21 power plant to illustrate equipment configuration and orientation and to predict plant operational performance. As a first step toward realizing this goal, Aspen Technology, ALSTOM Power, Fluent, Intergraph, and West Virginia University developed the software tools necessary for integrating process and equipment models. This report describes the activities and results of the project.

2.1 Motivation

Power plants of this century will need to produce power efficiently while reducing pollutant emissions to near zero. To preserve fossil fuels as a viable feedstock for future power plants, DOE's Vision 21 program aims to develop technologies that permit fuel and output flexibility, provide unprecedented increases in efficiency, have nearly zero emissions, and allow plant designs to be tailored to local conditions (Vision 21 Program Plan 1999 [1]). The approach taken to accomplish this goal is to develop a proven suite of technology modules that can be interconnected in suitable configurations to meet the different needs of specific markets.

Although Vision 21 plants are expected to be very complex, the number of actual demonstration plants will be limited by high construction costs and will be unable to test every unique local aspect of the design. In the absence of plant-level demonstrations, a key challenge will be to effectively use information from the testing and demonstration of Vision 21 technology modules to design and integrate complete Vision 21 plants. Fortunately, the means to address this challenge has been provided by the dramatic increase in the capabilities and prevalence of information and visualization technologies during the last two decades. There has been a concomitant rise in the usage of computers and engineering software for modeling power plants.

A concept for harnessing the various simulation tools is called the *virtual-plant simulator*, which will enable virtual demonstration of Vision 21 concepts. Although such a simulator will not make demonstration plants unnecessary, it can help reduce the number of demonstration plants and the overall technical risk level. The knowledge gained from the virtual-plant simulator is expected to decrease the development time for Vision 21 plants, reduce costly mistakes in their design, and improve their operating performance.

The virtual-plant simulator will ultimately consist of an integrated suite of software tools that can be used to design, simulate, visualize, and operate Vision 21 plants. The software suite will facilitate these tasks by accessing the repository of models and data that will result from Vision 21 technology module development projects.

Information from technology modules can be broadly categorized into two knowledge domains. One is the *process domain*, which includes information about the material and energy streams that flow between equipment items and transform within each equipment item. Much of the data captured and models created during technology module development fall into this category, and the conceptual design of Vision 21 plants will be largely based on this information. Another knowledge domain is the *physical domain*, which includes information about the piping that contains the material streams, the instrumentation, and the three-dimensional layout of the equipment items. Information about this domain will be required during the detailed design phase of Vision 21 plants.

This project addressed the question of how to use the process domain information derived from the technology modules to conduct integrated simulations and the subsequent conceptual design. In particular, simulation tools were integrated for two different scales: plant (system, process) scale and equipment (device) scale. Prior to this project, flowsheet simulation models, which are the standard basis for plant-scale studies, have typically not captured the knowledge represented by detailed CFD models or other design codes used by manufacturers. Thus, for example, the overall plant design based on flowsheet simulation would often omit important environmental or operational factors reflected in the equipment models. A principal focus of this project was to overcome this limitation.

2.2 Model reuse

Two types of models are widely used to represent process domain information: system models and equipment models. A system model is used to study the *integrated* performance of multiple pieces of process equipment, while an equipment model is used for more detailed analysis of the *individual* performance of a single piece of equipment. See Table 2.1 for a comparison of the two types of models.

System models are commonly constructed using process flowsheet simulation software, such as Aspen Plus®, to perform global mass and energy balances. They rely on a combination of simple, but fast-running, component models that are either zero-dimensional (0D), e.g., specified conversion, or one-dimensional (1D), e.g., plug flow reactor. Such models are useful for addressing system-level design questions (e.g., recycle stream options, thermal integration, overall system efficiency).

Equipment models are typically created using proprietary, "in-house" codes or with commercial, computational fluid dynamics (CFD) software (e.g., FLUENT®). They range from spreadsheet models based on engineering correlations to complex models that consider the detailed hydrodynamics, heat transfer, multi-component transport, and chemical reactions. Often they are 2-D or 3-D models that are suitable for optimizing the design of a piece of equipment (e.g., vessel geometry, burner sizing and placement, heat exchanger configuration).

Table 2.1: Comparison of Process Simulation and CFD models

	Process Simulation	CFD
Scope	Entire plant or system - typically hundreds of equipment items	A few (usually one) equipment item(s)
Resolution	lumped-parameter or one-dimensional	Detailed 2-D or 3-D geometry
Physics	Overall mass/energy balances	Distributed mass/energy/ momentum balances
Knowledgebase	Extensive physical properties and reaction kinetics database. Models for most unit operations.	Many physical sub-models for turbulence, combustion, mixing, radiation, multiphase, etc.
Computational time	Minutes to hours	Hours to days
Design questions answered	Determine recycle stream options, heat integration, overall efficiency	Perform equipment optimization and flow field visualization.

When designing a system, it is often desirable to share information iteratively between these two types of models or, better yet, run them simultaneously in an integrated fashion. This allows the overall system design to be based, as needed, on the more accurate and detailed information offered by equipment models. Likewise, an equipment model can make use of the extensive physical properties databases available in process simulation software; ensuring the consistency of the models is essential to avoid sub-optimal designs.

Such integration also ensures that using detailed equipment models to optimize individual components achieves a system-wide improvement, rather than just a local improvement at the expense of overall system performance. When plugged into a process flowsheet simulation, equipment models can take into account the effect of other equipment items on its input parameters, e.g., the effect of a recycle loop on the inlet composition. Detailed component models can provide constraints based on critical equipment parameters (e.g., turbine blade temperatures) during the optimization of the overall system. The use of an integrated process domain model is expected to reduce the time, cost, and technical risk of developing advanced energy systems.

Despite these significant benefits, integrating system and equipment models is not commonly practiced. Today, the exchange of information between system and equipment models must be manually accomplished (e.g., parsing of files, programming of user defined functions, *ad hoc* scripts, etc.) for each combination of models – a time-consuming and costly endeavor. Manually integrated models maintained even within a single company can break down when the original authors get transferred or leave the company. The manual integration of technology modules developed by different companies is highly impractical, if not impossible. Furthermore, models at different scales have different computational resource and platform requirements (e.g., memory, operating system, serial and parallel configuration), which makes manual integration difficult. Thus at the current status of modeling technology it is not possible to construct a reliable virtual-plant simulator for Vision 21 plants.

DOE identified these shortcomings as significant hurdles to the development of the desired advanced energy systems [1]. In October 2000, NETL awarded a cost-shared, cooperative agreement to a contractor team led by Fluent to develop a computational framework to integrate system and equipment models, allowing for the automated and seamless exchange of information between the two types of models.

2.3 Statement of project goals

The original project period occurred from October 2000 through September 2003. A summary of the project goals for that period is as follows:

- Develop an integrated suite of modeling codes consisting of Aspen Plus, FLUENT, and other detailed models to improve technological design process by providing physically-based simulations of Vision 21 components, modules, and complete plants.
- Develop the capability for accessing CFD results (e.g., flow fields, temperature profiles) from the process flow diagram (PFD) front-end provided by Aspen Plus.
- Demonstrate the capabilities of this advanced Vision 21 simulation and visualization tool.
- Form an Advisory Board including Vision 21 program participants and provide a system to solicit and respond to industry needs and input from this board.

During an extension period from October 2003 through December 2004, the project team completed several software enhancements: capability to specify CFD views and view results from remote machines; species mapping/filtering; extension to new types of ports; reaction basis conversion; error detection and recovery capability; Excel® connectivity; support to NETL for a presentation at the SuperComputing 2004 conference; testing a new methodology for converging the HRSG module; and LINUX and LAN execution of Demonstration Cases 2 and 3.

During the project, all the goals stated in the original and extension proposals were met or surpassed. A detailed account of the validation of the project goals is presented in Section 8.1.

2.4 Structure of the report

Section 3 gives an Executive Summary of the project. Section 4 is a contractually-required section on Experimental narrative, not applicable to this project.

The project narrative begins in Section 5, written from the software developer's perspective. Section 5.1 describes the activities in the original period, including discussions of the development process, the CAPE-OPEN standard, and the software architecture. Sections 5.2 and 5.3 discuss the additional developments completed during the extension period: Section 5.2 describes the Excel-FLUENT interface; Section 5.3 describes the error detection and recovery capability. Section 5.4 is included for completeness to outline software deployment; i.e., software installation and administration, setting up of a CFD database, and conducting simulations. However, the reader is directed to the toolkit User's Manual [5] for the complete instructions.

Section 6 describes how the software can be used for model development. This section is written from the perspective of the software users (CFD engineers and process engineers). Two examples are used to illustrate how integrated models can be built: a reaction-separation-recycle flowsheet with a CFD model of the stirred tank reactor (Section 6.1) and a fuel cell power system flowsheet with a 3-D CFD model of the reformer (Section 6.2). Section 6.3 outlines the project team's vision of organizational usage of the software for model building, CFD database accumulation, and simulation work flow. It briefly describes the collaboration between CFD engineers and process engineers within an organization and how models can be shared between organizations.

Section 7 describes two real-world power plant applications of the integration toolkit, as well as that of a FutureGen plant concept, based on demonstration simulations conducted by ALSTOM Power. Section 7.1 describes the simulation of a 33 MWe coal-fired municipal power plant based on a conventional steam cycle. Section 7.2 describes the simulation of a 270 MWe power plant based on a natural gas combined cycle (NGCC). Section 7.3 describes the simulation of a conceptual 250 MWe FutureGen integrated gasification combined cycle (IGCC).

Section 8.1 describes the project goals, along with qualitative and quantitative assessments of how well these goals were met. To show how the software developed in this project will help end-users, a comparison is presented of how certain problems were solved before and how they would be solved now. Section 8.2 describes the benefits that DOE and Industry would derive from this development.

Section 9 describes the technology outreach efforts. Section 9.1 describes the Intergraph task, in which the feasibility of coupling physical domain information with the integrated process model was investigated. Section 9.2 describes the Advisory Board activities and findings.

Section 10 shows how the software developed in this project is relevant to the vision set forth in the Vision 21 Program Plan [1] and discusses possible future work required to develop a comprehensive virtual-plant simulator.

3 Executive Summary

DOE's Vision 21 program is designed to allow integration of advanced power generation and chemical production technologies into systems that achieve the stringent performance, efficiency, and pollutant minimization goals for the future. These individual power, chemical, and fuel-conversion technologies are called *technology modules* implying the ability to interchange and combine them to form the complete Vision 21 plant that achieves the needed level of performance at affordable costs. To design advanced power plants of the twenty-first century, the DOE Vision 21 program identified the need for an integrated suite of software tools that could be used to simulate and visualize advanced plant concepts. Then the knowledge about the process and the physical plant captured in computational models of the technology modules can be linked together to simulate the entire Vision 21 power plant to illustrate equipment configuration and orientation and to predict plant operational performance.

Existing process simulation software did not meet this objective of *virtual-plant simulation*. Advanced component models are available; however, a seamless coupling capability that would integrate the advanced equipment (component) models to the process (cycle) simulation software remained to be developed. The inability to use models in an integrated manner causes *knowledge loss* (e.g., knowledge captured in detailed equipment models is usually not available in process simulation) and modeling inconsistencies (e.g., physical properties and reaction kinetics data in different models are not the same). A team consisting of Fluent Inc., ALSTOM Power Inc., Aspen Technology Inc., Intergraph Corporation, and West Virginia University, in collaboration with the National Energy Technology Lab, took first steps toward developing a *virtual-plant simulator*.

The original project period occurred from October 2000 through September 2003. A summary of the project goals for that period is as follows:

- Develop an integrated suite of modeling codes consisting of Aspen Plus, FLUENT, and other detailed models to improve technological design process by providing physically-based simulations of Vision 21 components, modules, and complete plants.
- Develop the capability for accessing CFD results (e.g., flow fields, temperature profiles) from the process flow diagram (PFD) front-end provided by Aspen Plus.
- Demonstrate the capabilities of this advanced Vision 21 simulation and visualization tool.
- Form an Advisory Board including Vision 21 program participants and provide a system to solicit and respond to industry needs and input from this board.

During an extension period from October 2003 through December 2004, the project team completed several software enhancements: capability to specify CFD views and view results from remote machines; species mapping/filtering; generic, extension to new types of ports, and heat exchanger model ports; reaction basis conversion; error detection and recovery capability; Excel® connectivity; support to NETL for a presentation at the SuperComputing 2004 conference; testing a new methodology for converging the HRSG module; and LINUX and LAN execution of two demonstration cases.

The software development process was conducted using a hybrid of the standard *waterfall* model and the *unified development process*; i.e., progressing where necessary, through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance. Thus, the design was often modified to accommodate new requirements discovered during development and testing. Although at the start of the project a survey was conducted to identify user requirements, the bulk of user requirements was generated from brainstorming sessions conducted by the project team. The results of this activity were documented in a User Requirements Document (URD). Based on the URD, a Software Requirements Document (SRD) was created to identify programming interfaces. In addition, several *use-cases* were identified and documented. Based on this information, a software design was generated. The software design ideas were discussed in weekly development team meetings, and the developers fleshed out the details in a Software Design Document. The designs were reviewed in four quarterly design review meetings conducted from October 2001 to October 2002. Thereafter, the software implementation and testing were completed according to a Software Development Plan.

The team adopted CAPE-OPEN (CO) interfaces, the *de facto* standard for interfacing process modeling software components for use in the simulation, design, and operation of processing plants. This standard was born out of two European Union projects and the result of five years of international collaborative work involving more than thirty leading process industry companies, researchers, and software vendors in Europe, Asia, and North America. The software exploits three major classes of CO interfaces—unit operations, physical properties, and reaction kinetics. The software developed in this project is the first demonstration of the use of CO interfaces to link CFD models with process simulators. New interface requirements identified during this project were communicated to the CO standards committee.

The new software capability was designed to make the construction of integrated models fast and integrated simulations robust and user-friendly. Configuration wizards were developed to make CFD and custom models CO-compliant. With the FLUENT Configuration Wizard, a CFD engineer can specify which CFD model parameters (e.g., current and voltage for a fuel cell model) and boundary zones to make available as variables and stream ports, respectively, in Aspen Plus. The CO-compliant equipment models are stored in a model database and become readable in Aspen Plus, just as any other native models. The configuration process, which takes less than two hours, is typically two orders of magnitude faster than the manual integration approaches used previously.

An Integration Controller (IC) was developed to facilitate the exchange of information between equipment models and process models. The IC provides three graphical user interfaces (GUIs), to the process engineer:

1) Model Selection GUI to browse and select a suitable equipment model from the Model Database;
2) Model Edit GUI to modify parameters for the equipment model;
3) CFD Viewer to display, within the process simulator, the results of a CFD simulation conducted as a part of an integrated simulation.

The IC was developed to allow the remote execution of equipment models. For example, an Aspen Plus model running on a Windows machine can launch and run a FLUENT model on a LINUX cluster located on the LAN.

Because equipment models can be much more expensive to run than process models, the software toolkit was designed to accommodate a siginficant disparity in computational time. A reduced order model (ROM) framework and a *solution strategy* capability were incorporated in the IC. A ROM is a class of models of reduced fidelity and run-time derived from detailed equipment simulation results. For example, a correlation between the inputs and outputs of CFD model is a simple form of a ROM. The solution strategy capability allows a unit operation to be represented with a combination of different models. The process engineer then has the ability to choose a combination that speeds up the calculation. For example, the engineer may choose to combine a ROM and CFD model, using a fast ROM for the initial Aspen Plus iterations and the high-fidelity CFD model for the final iterations.

To demonstrate industrial power plant applications, the team defined three cases: (1) a conventional 30 MWe coal-fired steam plant for municipal electricity generation, (2) an advanced 250 MWe, natural gas-fired, combined cycle (NGCC) power plant, and (3) a 250 MWe FutureGen integrated gasification combined cycle (IGCC) power plant. Although the first two demonstration cases are not as advanced as the FutureGen cycle, they represent actual existing power plants and embody features that any virtual-plant simulator would be required to model. Three runs were completed for each of the first two demonstration cases: (a) an initial baseline run using the standard component libraries in Aspen Plus, (b) a second run where one of the library components was replaced with an ALSTOM Power proprietary code, and (c) a third run where a cycle component was replaced with a FLUENT CFD model. Both sets of the three runs were successfully completed over a range of loads on a PC. The third demonstration case was run only at the maximum load point with a FLUENT CFD module. Both Demonstration Cases 2 and 3, coupled with FLUENT, were run over a LAN.

In the conventional steam plant, a three-dimensional FLUENT CFD model represents the gas-side and steam-side of the boiler. An Aspen Plus "design specification" is used to adjust FLUENT model parameters, namely the damper position and excess air, to control the superheated steam temperature at 763 K. In the NGCC plant, a 3-D FLUENT model is used for a two-pressure, once-through heat recovery steam generator (HRSG), which consists of several nested heat exchangers and pollutant control devices. An Aspen Plus design specification is used to adjust the high-pressure economizer feed rate to achieve a superheated steam outlet temperature of 838 K. In the FutureGen IGCC plant, a 3-D FLUENT model is used for a three-pressure HRSG. Aspen Plus design specifications are used to adjust the intermediate- and high-pressure economizer feedwater flows in order to balance the circulation requirements for the drums. For the HRSG cases, which have a large number of connectivity points with the cycle, the Aspen Plus solver capabilities were also utilized to converge the CO variables transferred between FLUENT and Aspen Plus.

The software development effort was continually guided by the end-user requirements. The primary means of ensuring this guidance was by having the team member ALSTOM Power participate in software design review meetings and conduct demonstration simulations during the development phase. In addition, the project team sought the advice of engineers from NETL, other Vision 21 program participants, and representatives from other power and chemical/process companies. At the start of the project, a survey was conducted to identify user requirements. During the project, semiannual Advisory Board meetings were conducted to review the progress, demonstrate the current prototype version of the software, and collect feedback from the industrial participants. The comments and suggestions from the potential end-users was used to adjust the software design and development plans.

By using existing commercial software Aspen Plus and FLUENT we have ensured that the technology is immediately available for use by US industry and will remain supported far into the future. The use of the open standard CO will allow other companies to plug their models into the integrated software suite. We have already demonstrated that the integration software works with another commercial CO simulation executive, HYSYS®. Another demonstration will occur in a project begun in 2004, funded by the Department of Trade and Industry of the United Kingdom, that will use this software to link FLUENT with gPROMS®.

The integration software was made available to industry in November 2003 and won the 2004 R&D award as " … one of the 100 most technologically significant products introduced into the marketplace over the past year". In addition to power plants, the integration software will ultimately improve analysis, design, and optimization of plants in the chemical, petroleum, and other process industries. Nearly 100 engineers from commercial companies attended two web seminars on the software conducted in October 2003 and March 2004. The first sale of the commercialized version of the software toolkit was to a major chemical company.

The software developed through this project enables designers to conduct process simulations with a simultaneous scope and accuracy never before possible. The deep insight from such simulations will help designers to identify opportunities for achieving unprecedented high efficiency and near-zero emissions in advanced power plants.

4 Experimental

There were no physical experiments planned or conducted as part of this project. This contractually required section of the report is therefore not applicable and is intentionally left blank.

5 Software Development and Deployment

This section describes how the software was developed and describes the software architecture. An outline of software deployment as described in the integration toolkit User's Manual [5] is also provided.

5.1 Software development

5.1.1 Development process

The software development process was conducted using a hybrid of the standard waterfall model and the unified development process; i.e., progressing where necessary, through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance. Thus, the design was often modified to accommodate new requirements discovered during development and testing. Although at the start of the project a survey was conducted to identify user requirements, the bulk of user requirements was generated from brainstorming sessions conducted by the project team. The results of this activity were documented in a User Requirements Document (URD). Based on the URD, a Software Requirements Document (SRD) was prepared to identify programming interfaces. In addition, several use-cases were identified and documented. Based on this information, a software design was generated. The software design ideas were discussed in weekly development team meetings, and the developers fleshed out the details in a Software Design Document. The designs were reviewed in four quarterly design review meetings conducted from October 2001 to October 2002. Thereafter, the software implementation and testing were completed according to a Software Development Plan.

5.1.2 CAPE-OPEN standard

The Vision 21 integrated software infrastructure delivered in this project is based on the CAPE-OPEN (CO) standard. CAPE-OPEN is the *de facto* standard for interfacing computer-aided process engineering (CAPE) software. Born out of the European Union's CAPE-OPEN Project (1997-1999) and the Intelligent Manufacturing Systems (www.ims.org) Global CAPE-OPEN Project (GCO, 1999-2002), the CO standard represents five years of international collaborative work involving more than thirty of the leading process industry companies, researchers, and software vendors in Europe, Asia, and North America. A recent review of industrial applications of the CO standard, including a brief discussion of the integrated Aspen Plus and FLUENT solution, can be found in Pons (2003).

The CO standard is open, multi-platform, available free of charge, and supported by many of the leading commercial CAPE products, such as Aspen Plus and HYSYS® from Aspen Technology, PRO/II® from Simulation Sciences, and gPROMS® from Process Systems Enterprise Ltd. Other specialist suppliers have also produced CO-compatible versions of their process simulators and thermodynamics packages. In addition to these commercial implementations, many other facilities are in an advanced state of development. For example, there are software utility "wizards" that enable existing code to be wrapped into CO-compliant components for both unit operations and thermodynamics and testers to check components for CO compliance.

Today the CO Laboratories Network (CO-LaN, www.colan.org) is the internationally recognized, user-driven organization for the management, exploitation, and dissemination of the CO standard. CO-LaN members include industrial users of CAPE software, providers of CAPE software, government agencies, academic institutions carrying out research activities in the field of CAPE, and other interested parties. Full members from the process industries include Air Liquide, BASF, BP, Dow, IFP, Norsk Hydro, Shell, and Total Fina ELF. CO-LaN associate members include Vision 21 project partners, AspenTech and Fluent, as well as a dozen or more other process modeling software suppliers.

Using the CAPE-OPEN standard, our main software component, namely the Integration Controller (IC), integrates the Aspen Plus process simulator and the FLUENT CFD package. Over the course of this three-year project and the extension period, we used a phased approach to upgrade the IC to work with the latest available releases of the CO standard, Aspen Plus, and FLUENT. CO v1.0 is the current official release and final product of the GCO Project. Our IC is designed for use with steady-state process simulators that are compliant with the CO v1.0 interfaces. To date the IC has been used and tested with Aspen Plus and HYSYS.

The IC exploits three major classes of CO interfaces—unit operations, physical properties, and reaction kinetics. The CO unit operation interface enables the seamless use (e.g., create, edit, solve) of FLUENT equipment models in the Aspen Plus process flowsheet. This interface also facilitates the bi-directional exchange of stream information (flow rate, temperature, pressure and composition) between Aspen Plus and FLUENT. The project addressed the task of mapping the multi-dimensional CFD boundary conditions to the single-point Aspen Plus streams and *vice versa*. This mapping is considered sufficient for system-level simulations of Vision 21 plants where process equipment items are typically connected by pipes. More tightly coupled equipment models requiring the transfer of multi-dimensional boundary conditions can be handled in a single CFD model or combined in another problem-solving environment, which can then used in Aspen Plus via the CO interfaces.

The CO physical property interface is used to transfer constant or temperature-dependent pure-component physical properties (e.g., density, viscosity, heat capacity, thermal conductivity, and molecular weight). The CO reaction kinetics interface facilitates the automatic transfer of reaction stoichiometry and power-law parameters.

The CO interfaces are based on universally recognized interoperability standards, COM and CORBA. COM (Component Object Model) refers to both a specification and implementation developed by Microsoft Corporation that provides a framework for integrating software components running under the Windows operating system (www.microsoft.com/com). CORBA (Common Object Request Broker Architecture) is a specification of a standard architecture for object request brokers (ORBs), which allows vendors to develop ORB products that are portable and interoperabile across different programming languages, hardware platforms, and operating systems (www.corba.org).

The IC establishes a platform for middleware interoperability using a COM-CORBA Bridge. Osawe et al. (2000) presented the details of bridge implementation. The bridge allows models running under Windows to exchange information with models running under a different operating system. It translates COM objects into CORBA objects and *vice versa*. COM implementations of CO interfaces are available in Aspen Plus. CORBA implementations of CO interfaces were included in FLUENT. Thus, for example, Aspen Plus running under the Windows 2000 operating system is able to communicate with FLUENT running under the LINUX operating system via the COM-CORBA Bridge.

Throughout the course of this project, we provided AspenTech with considerable feedback on the implementation and use of the CO interfaces in Aspen Plus. In addition we proposed to CO-LaN various interface enhancements and extensions related to Vision 21 plant simulations. For example, we recommended an extension to allow CO-compliant process simulators to interrupt the execution of computationally expensive CO-compliant unit operation models (e.g., CFD). As CO-LaN associate members, AspenTech and Fluent will continue to pursue CAPE-OPEN issues relevant to the integration of process simulation and CFD.

5.1.3 Architecture/Components

The integrated simulation environment has a three-tier architecture system consisting of the process simulation executive, the Controller subsystem, and the external unit operation server(s). The implemented architecture of the Controller is presented in Figure 5.1.

The Controller subsystem de-multiplexes incoming data originating from the process simulator to the activated external solvers or unit operation models. In the reverse flow of data, updated outlet stream properties and the unit operation parameters are multiplexed by the subsystem and made seamlessly available to the Aspen Plus. In principle, a single copy of the IC executable can be loaded by multiple instances of CO-compliant process simulators on the same machine provided there are no licensing restrictions. There are no known limitations on the number of simulation or equipment models that may be coupled with a single process simulator via the integration toolkit.

The FLUENT Configuration Wizard is a tool for transforming a regular FLUENT CFD model into a CO-compliant model so that it can be readily integrated with the host process simulator. It is currently implemented as a loose extension of the FLUENT code. The Custom Configuration Wizard is a stand-alone tool for rapid integration of external unit operation models other than FLUENT equipment models (e.g., industrial proprietary codes).

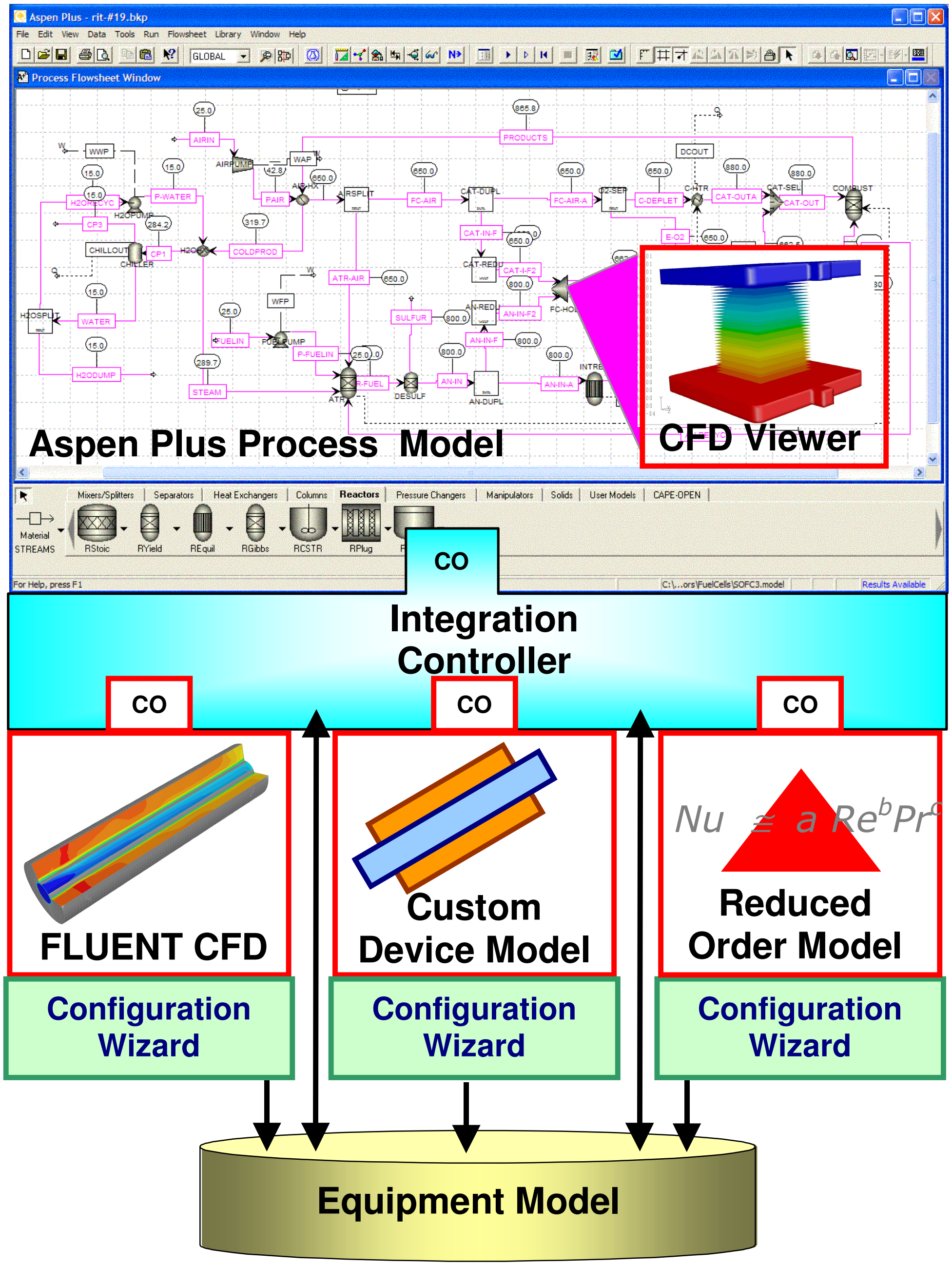


Figure 5.1: Three-tier architecture system of the integrated software

The CFD Viewer enables the analyst to view the internal distributions of species, velocities, pressures, temperature and the convergence residuals computed by FLUENT in the process simulator environment. The CFD Viewer is launched from the IC Model Edit GUI to display the graphics results that are generated by the FLUENT model. The CFD Viewer is designed to display the graphic results of external operation models other than the results of FLUENT simulations, provided the results are restricted to the options that can be displayed in the CFD Viewer.

In implementing the graphics display capability, a couple of methods were added to the CORBA IDL specifications for the ICapeUnitReport interface so as to permit an easy and efficient transfer of the graphics contents from FLUENT to the IC over a LAN, without the need of having to encode the file contents before transfer, and to decode the received graphics data in the IC. We have proposed these extensions to the CO-LaN, and they are likely to be included in a future release of the CORBA IDL specifications.

The Equipment Model (EM) database is a directory/file based repository for storing input and output files of equipment models, including pre-computed CFD results, which are employed by the Reduced Order Models for generating fairly accurate and fast-running equipment models relative to the base CFD model. Data stored as pre-computed results need not originate from the FLUENT CFD solver; conformance to the specified storage format suffices to reuse the results stored in the database to develop a ROM. It is intended to replace the existing method of storing pre-computed data with a standalone database module so as to insure data integrity, and provide for better scalability and data management. The EM database system is discussed in more detail in Section 5.1.3.4.

Although all the development was done using Aspen Plus as the simulation executive, the IC has been successfully demonstrated to work with HYSYS process simulator. Details of the components of the IC subsystem are presented in the following sub-sections.

5.1.3.1 FLUENT Configuration Wizard

The FLUENT Configuration Wizard is an easy-to-use tool that is designed to transform FLUENT case and data files into a CO-compliant external unit operation model. The Wizard generates the following files, which are placed in the *.model sub-directory in the EM database:

Master XML file
Solver XML file

For an illustration, the above files could be placed in the appropriate subdirectory in the C drive as:

C:\ModelDatabase\Reactors\CSTRs\CSTR1.model\Master.XML
C:\ModelDatabase\Reactors\CSTRs\CSTR1.model\Fluent2D.XML

The contents of both files are required by the Controller to populate the various fields and grids of the IC GUIs, as well as to display the various options to the user. The files also make it possible to integrate external unit operation models at run-time without having to recompile and re-link the IC executable for every new model that is coupled with the process simulator.

The Master XML file contains data that are generic to the equipment type, irrespective of the selected solver(s). Typical examples are cell current for a fuel cell unit; the inlet/outlet boundary names and types, i.e., whether the inlet/outlet boundary is a material, energy or informational port. The Solver XML file contains the solver specific information such as the launch command(s), the solver path and the default number of solver iterations. The solver specific data are required for launching and registering the solver reference and attributes with the IC Session Manager (see section 5.1.3.3 for more details on the Session and Solver Managers).

5.1.3.2 Custom Model Configuration Wizard

The Custom Model Configuration Wizard is very much similar to the FLUENT Configuration Wizard, with the essential difference that it does not require a FLUENT case file as input to generate the XML files that are required by the Controller GUIs. The Wizard has tabbed pages, in a manner similar to the FLUENT Configuration Wizard, and is intended for use in integrating external unit operation models other than the FLUENT model. The tool generates an additional output file: the Collections.dat file. The collections file contains the list of the unit's ports and parameters as specified by the user during the configuration steps. It is placed in the solver sub-directory in the model database upon clicking the finish button on the last tabbed page of the Wizard. For an illustration, a Collections.dat file could be placed in the Regression sub-directory after running the Wizard as specified below:

C\ModelDatabase\Reactors\CSTRs\CSTR1.model\Regression\Collections.dat

The data contained in the collections file is employed by the custom model to create the required collection of COM ports and parameters during bootstrapping. Further details on using the Configuration Wizards are available in the IC User's Manual [5].

5.1.3.3 Controller COM-CORBA Bridge

The Controller COM-CORBA Bridge design pattern [4] is implemented using the C++ programming language, COM and CORBA middleware technologies. A software Bridge pattern is a design concept, which decouples an abstraction from its implementation so that both can be varied independently as the need arises. A COM-CORBA Bridge pattern was necessitated for two main reasons. The first was to provide an effective mechanism or infrastructure that facilitates seamless inter-process communication between Aspen Plus and the external unit operation(s). In the envisaged and demonstrated scenarios, the external unit operation process need not run on the same machine as the simulation executive, and the operating systems of the two machines can also be dissimilar. In the latter case, cross-network and cross-platform issues become significant, so that the need to achieve a robust, reliable and scalable interoperable bi-directional exchange of data without significant network latency was considered of prime importance.

Another factor that was taken into account is that Aspen Plus only runs on Windows, and it relies heavily on the COM/DCOM technology that is native to the Windows platform. This naturally implied that COM/DCOM must form an integral part of any adopted enabling middleware technology, irrespective of the route taken to integrate the disparate codes via inter-process communication. Although COM/DCOM has proven to be an effective middleware technology in enterprise and in a few scientific applications, it has also been shown to be largely platform dependent. For example, a COM server can hardly be installed on a UNIX platform without an interfacing proprietary software layer. In addition, the available documented evidence of its use clearly points to the fact that the

COM/DCOM middleware technology is best employed in a Windows-only network environment. Thus, the need to overcome cross-platform barriers had to be met, given that a sizeable proportion of CFD simulations are conducted on UNIX and LINUX based systems.

A closely related consideration to the above limitation is that COM/DCOM is also not entirely language-neutral. There are no language mappings for the technology outside of C++ and Visual Basic; it is only directly interoperable with J++ or J# (non-standard Java), which clearly raises additional implementation and portability issues if heterogeneous platforms and networks are to be supported. Therefore to avoid significant future re-engineering and rewrite of wrapper implementation code, a design decision was made by the project team to employ a more interoperable middleware technology. Hence, CORBA, which is a well-established and supported mainstream middleware technology, was selected. CORBA library implementations are available on almost every combination of hardware and software, and have been successfully applied to a number of mission-critical applications such as in avionics. CORBA has most, if not all, of the desirable features of location transparency, interoperability, and language and platform independence.

We remark, however, that in spite of its ubiquitous position in the middleware arena, direct interoperability with other leading middleware technologies through the COM/CORBA inter-working specifications of OMG, for example, is not widely available. Cheap and wide availability of such direct interoperable capability, where a COM object can directly be employed in a CORBA environment as if it were a CORBA object and vice-versa, would have precluded the need to implement the COM-CORBA Bridge infrastructure. While such off-the-shelf Bridges are available, e.g., from Iona (www.iona.com), the product cost was considered unacceptable; the project team developed an equivalent capability instead.

In source code terms, the CO COM interfaces are implemented in both the process simulator and the IC subsystem. The COM interfaces implemented in the IC subsystem are shown in Figure 5.2, using the industry standard lollipop diagramming technique for interfaces. These interfaces constitute the components of the external unit operation model from the standpoint of the process simulator.

Since the CO-compliant CORBA wrappers must extract reaction kinetics data and thermo-physical properties from the simulation executive via the Controller, CORBA Servants for the parameter related interfaces (ICapeParameter, ICapeParameterSpec, etc.), ICapeIdentification, ICapeRealParameter, ICapeThermoMaterialObject and ICapeReactionChemistry interfaces were implemented as part of the Controller subsystem. A CORBA Servant is the C++ implementation of a CORBA object as defined by the IDL specifications. The thermo-material object, electrolyte, and kinetic reactions interfaces are implemented by the simulation executive, and the object references are made available to the Bridge environment via the “Connect” method of the ICapeUnitPort and the “SetReactionContext” method of the ICapeKineticReactionContext interfaces, respectively.

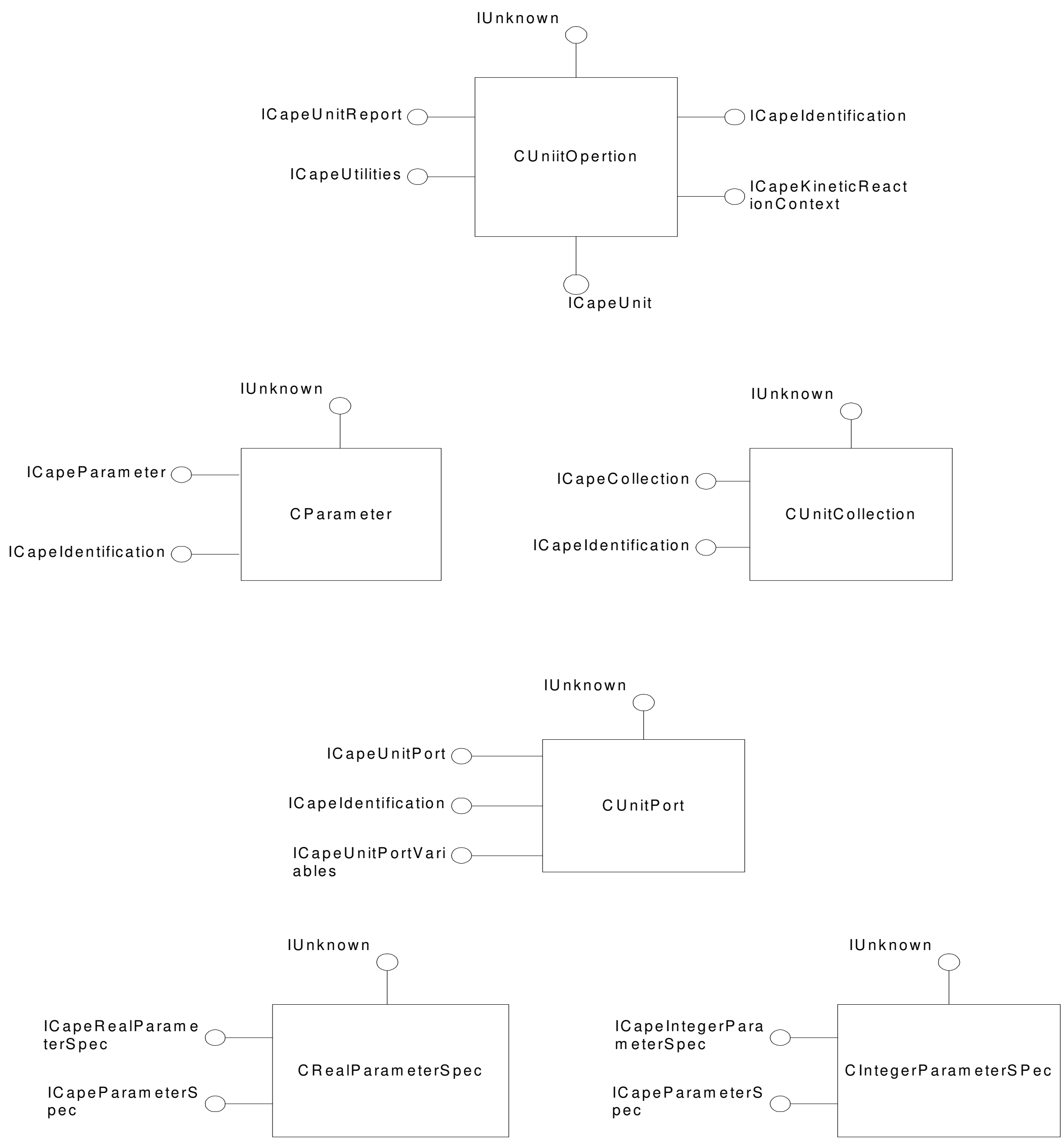


Figure 5.2: Diagrams showing the COM components implemented in the Integration Controller subsystem

In the transfer of thermo-material and reaction kinetics data, the external unit operation plays the role of a client, whilst the Controller subsystem acts as the server, which makes it imperative to also implement CORBA servants in the Controller subsystem. The implementation of the CORBA ICapeReactionChemistry interface is rather more involved because the interface makes use of the ICapeCollection and the parameter related interfaces (ICapeParameter, ICapeParameterSpec, ICapeIntegerParameterSpec, etc) to provide the needed response to clients' request for reaction kinetics data.

In the current hybrid implementation, the C++ COM classes commonly have CORBA servants and object references as attributes and vice-versa, thereby ascribing dual attributes to the COM and CORBA object implementations. For an illustration, a run-time request from the process simulator to get the collection of ports and parameters in the external unit operation model is actually made on the CORBA object reference, which the corresponding local COM object has as an attribute. In effect, the implemented COM classes mainly serve as wrapper facades to the CORBA object implementations.

The simulation executive in the above-described context is completely oblivious of which external unit operation is servicing its request; the external unit operation model that services the requests from the process simulator is determined by the IC, and it is based solely on the iteration number of the process simulator. A simple illustration of the Bridge design pattern is shown in Figure 5.3, where calls made, for example, by Aspen Plus to retrieve the pre-configured ports in the external unit operation (CUnitOperation), are marshaled to the CORBA ICapeUnit reference. The accompanying sequence diagram is added to emphasize the time sequence of calls.

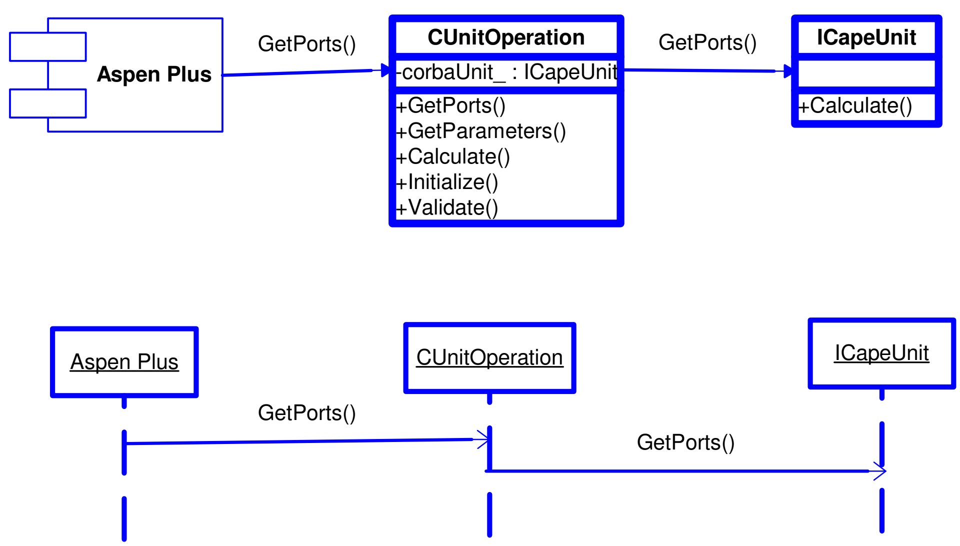


Figure 5.3: Illustration of the Bridge design pattern implementation in the integrated simulation environment

Central to the Controller are the SolverManager and SessionManager C++ classes, which have been implemented as Singletons to preclude arbitrary creation and deletion of external unit operation models. The SolverManager class is responsible for the launching of the solvers that have been selected for each CO block on the flowsheet, based on the data supplied to it by the IC Model Edit GUI; it also ensures that the activated solvers for a given CO block are correctly terminated when the user deletes the block from the process flowsheet.

The SessionManager class controls the lifetime of the SolverManager class, and by implication, the lifetimes of all the activated solver references. It also maintains a mapping of the user-specified solution strategies to the CO blocks on the flow sheet so as to ensure that solver switching is executed correctly at run-time. The SessionManager ensures that the correct FLUENT graphics results are displayed in the CFD Viewer by employing a mapping of the outer or flow sheet iterations to the activated solver references.

Each CO block is associated with an instance of a SolutionStrategy C++ class, which defines how many process flowsheet or outer iterations should be executed before switching the simulation of the external unit operation model to the next activated solver. User modifications to an existing solution strategy via the IC Model Edit GUI are updated with the SessionManager class.

5.1.3.4 EM Database

The fundamental purpose of the EM Database is facilitation of equipment model reuse. Developing a CFD model is a time-consuming process, requiring the advanced skills of a CFD analyst. But once this is done, the EM database allows the model to be stored for future use by a process analyst, who need not have CFD skills, but wishes to use the model as one Unit Operation within an Aspen Plus flowsheet process model to conduct a more accurate integrated simulation.

A second type of reuse is possible if the results of a computation made with the CFD model in the simulation can be stored along with all the distinguishing characteristics of the computer results: the species, stream values and parameters. These results, over many runs, can then be used to construct a ROM that can save future computation time by delivering the response to the process simulator.

Storing CFD models requires a file native to the CFD package (here, FLUENT), along with metadata about the model (Name, Category, Type, CFD Solver, etc). Storing the results of a computation with a particular model requires, in addition, the storage of results files, and characteristics important for interpolation purposes: the ports, species, and stream properties. The Model and the Computed Results are stored in self-describing XML files (see 14.1 for an example). There is a two-level hierarchy for storage of models, namely, category (e.g., Reactors) and type (e.g., CSTRs), which is illustrated in Figure 5.4.

The names and location of the files are defined in the storage tag of the model's XML files. All files are stored within a folder to which the pathname may be specified. The current implementation of the model hierarchy-browsing interface uses the file system to represent a hierarchy tree, and also to save the model's XML and results files. In future, a relational or object relational database could be implemented for better security, easier administration, and more refined access control.

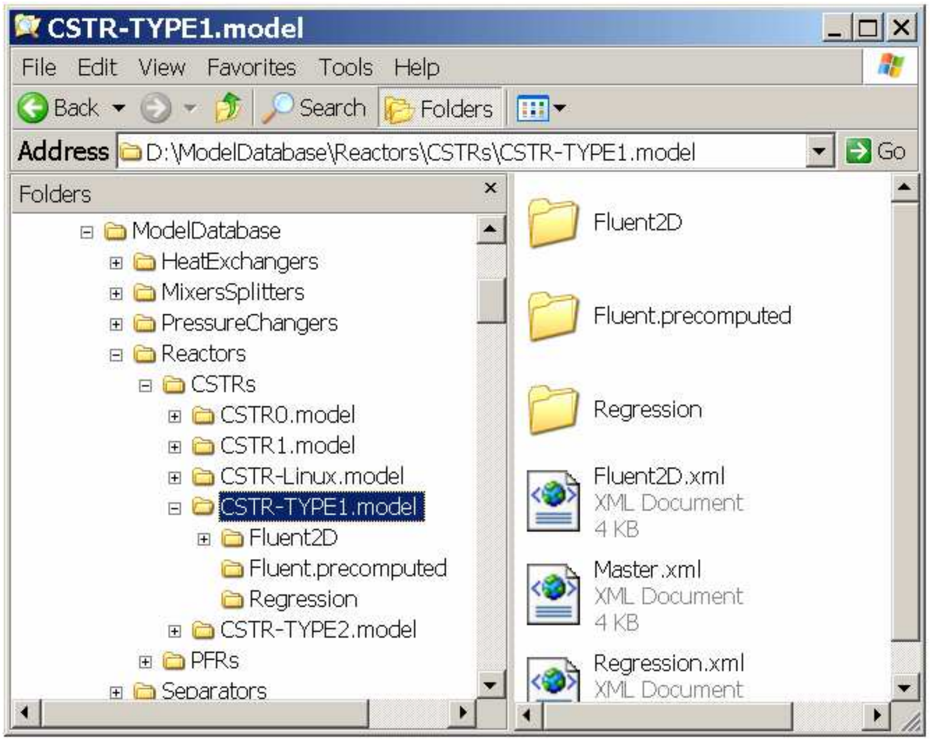


Figure 5.4: Organization of Models in the Database

Two independent sets of interfaces are available for model browsing and for data access. The former is used only for selection of a model by the Model Select GUI of the IC. The latter is used directly by the COM-CORBA Bridge to obtain the information needed to create a customized CAPE-OPEN unit.

For browsing purposes a set of application programmer interfaces (APIs) has been defined to access the EM database, and within that the category and type, in succession, until the desired model is found and selected. The defining data for the selected model is then accessed via a façade class called COModel, which provides a unified set of interfaces to the EM database. These interfaces hide the database implementation details, thus allowing a change in the underlying technology for the database in the future, without affecting the other parts of the system.

Apart from the above read interfaces, a set of classes has been implemented to enable writing the end data from an integrated simulation into a database of pre-computed results. Pre-computed results are stored along with the solver used, parameters, inlet ports, outlet ports, and solver data files. Recognizing that the same model may be run with different species, a class has been implemented to dynamically add new species into the inner structures while reading the results file. If a species were absent for a particular simulation run, then its mass fraction value is filled with zero in the result vector for that run.

5.1.3.5 Solution Strategy

The solution strategy functionality provides the flexibility to use multiple models for the same equipment item at different stages of calculation and analysis. The flexibility to "mix and match" different models in simulating a unit operation may be desirable for several reasons. Because the turn-around time in employing a CFD solver to simulate a unit operation is usually orders of magnitude higher (especially for 3-D models) than the time taken by a process simulator to simulate the same unit, one possibility to significantly reduce computational cost is to start the simulation using a ROM. For example, the available regression model may be employed in the first several iterations, and a high-fidelity FLUENT model may be used during the final stages of convergence of the process flowsheet model.

Since the iteration number of the process simulator is the sole criterion for switching equipment solvers, the user only needs to specify the number of times that an external solver should be invoked before switching to the next solver as specified in the Solution Strategy table of the Model Edit GUI. In Figure 5.5, for example, the Regression ROM is specified to execute twice before switching to the 2-D version of FLUENT solver, which will in turn execute a hundred times. Using more sophisticated switching criteria, such as the stream information, have been precluded in this initial release due to insufficient time to fully test the accuracy and robustness of some of the identified approaches.

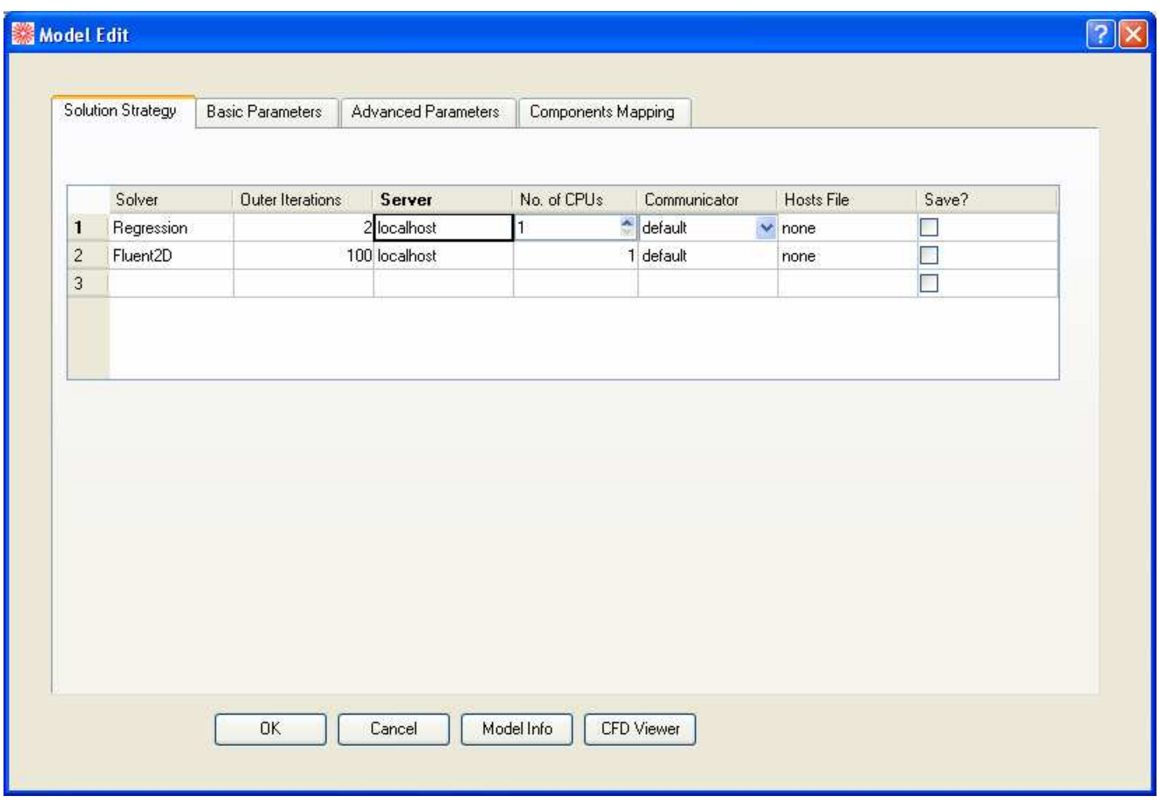


Figure 5.5: Solution Strategy tab in the Model Edit GUI

5.1.3.6 Reduced-Order Model

A Reduced-Order Model (ROM) based on pre-computed FLUENT data was developed as a first demonstration of the use of the supplied CORBA wrapper template. The ROM is based on the classic multiple linear regression technique, in which the independent or regressor variables are taken as the inlet mole fractions, the flow rates, and the model input parameters such as temperature and pressure. The stream data are predicted at the outlet ports (one at a time); the values of output parameters are updated based on the least squares error minimization technique.

For an illustration of this procedure, suppose the results of *n* precursor CFD simulations are available in the database, where *n* is an integer that is greater or equal to the total number of independent or flow variables that are pertinent to the system behavior. The linear equation model for each of the dependent variable may then be written as:

$$y_i = \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i \;, \quad \text{i = 1, 2, …, n} \tag{1}$$

In equation (1), the y_i are the dependent variables at an outlet port(s), i.e., the mole fractions, flow rate(s), temperature, pressure, etc. The key idea in the model formulation is to generate some set of β coefficients that minimizes the least squares function given by:

$$L = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} \left(y_i - \beta_o - \sum \beta_j x_{ij} \right)^2 \tag{2}$$

The function L is minimized with respect to $\beta_{0,} \beta_1, \ldots\ldots\ldots, \beta_k$. The least squares estimators of the preceding variables must, however, satisfy:

$$\left. \frac{\partial L}{\partial \beta_0} \right|_{\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k} = -2 \sum_{i=1}^{n} \left(y_i - \hat{\beta}_o - \sum_{j=1}^{k} \hat{\beta}_j x_{ij} \right) = 0 \tag{3}$$

and

$$\left. \frac{\partial L}{\partial \beta_j} \right|_{\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k} = -2 \sum_{i=1}^{n} \left(y_i - \hat{\beta}_o - \sum_{j=1}^{k} \hat{\beta}_j x_{ij} \right) x_{ij} = 0 \,, \qquad \text{j = 1, 2, 3,…, k} \tag{4}$$

Methods of simplifying the above equations to obtain the design matrix and the resulting set of linear equations can be founded in standard texts on mathematics and statistics (e.g., Spiegel and Stephens [3]). The formulations resulting from further simplification of the normal equations were coded up in a CO-complaint wrapper, using the proprietary CORBA template that is included in the installation CD-ROM. Since the focus of the current effort is to develop an infrastructure for integrated simulation rather than model building, rigorous computational procedures for selecting or de-selecting the candidate regressor variables have been precluded in the current implementation. Thus, the variables used in the example cases were selected based solely on insight and numerical experimentation.

Although model testing is still ongoing, acceptable predictions have been obtained, which indicate that the adopted default independent or regressor variables are reasonably applicable to the CFD problems that have so far been tested. In addition, accumulated experience in the use of the multiple regression technique tends to suggest that the richer the data set of pre-computed results, the better the performance of the model. Thus, successful use of the regression model is strongly dependent on both the quantity and quality of the available pre-computed CFD results stored in the database. In summary, the ROM appears to be a promising tool for knowledge reuse and time saving.

For an illustration of the model applicability, a sensitivity analysis was set up in Aspen Plus for the continuous-stirred tank reactor (CSTR) test problem to be described in Section 6.1, such that the impeller shaft speed was varied from 85 to 400 rpm. In Figure 5.6, FLUENT predictions for the variation of product purity with impeller speed is matched against the results of the regression model. The ROM correctly captures the dip in the profile at a shaft speed of about 125 rpm, where the reaction mechanism switches from finite rate to eddy-dissipation as the effect of mixing becomes more significant in the reactor. In Figure 5.7, the variation of the desired reactor yield with shaft speed for both solvers are compared, and the plots show that the peak yield is attained at a slightly lower optimum shaft speed of 175 rpm for the ROM solver.

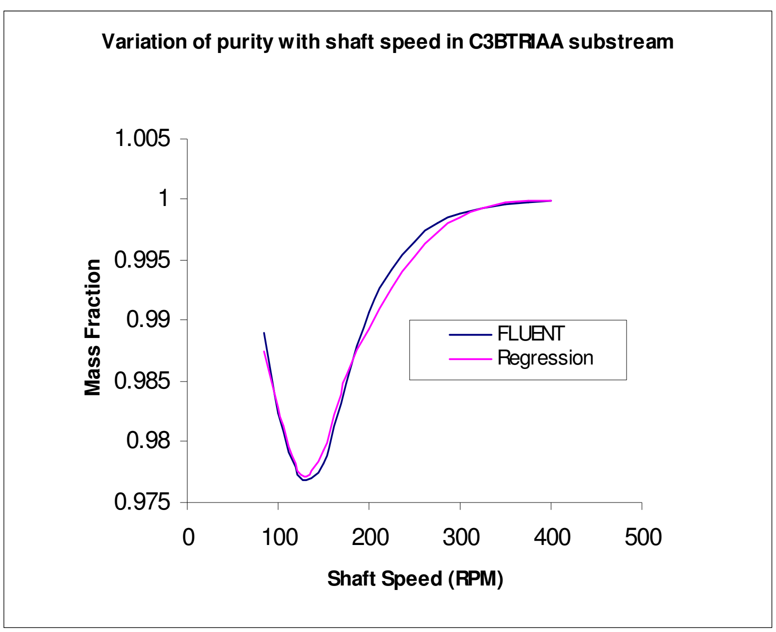


Figure 5.6: Comparison of the variation of product purity with shaft speed using FLUENT and regression model

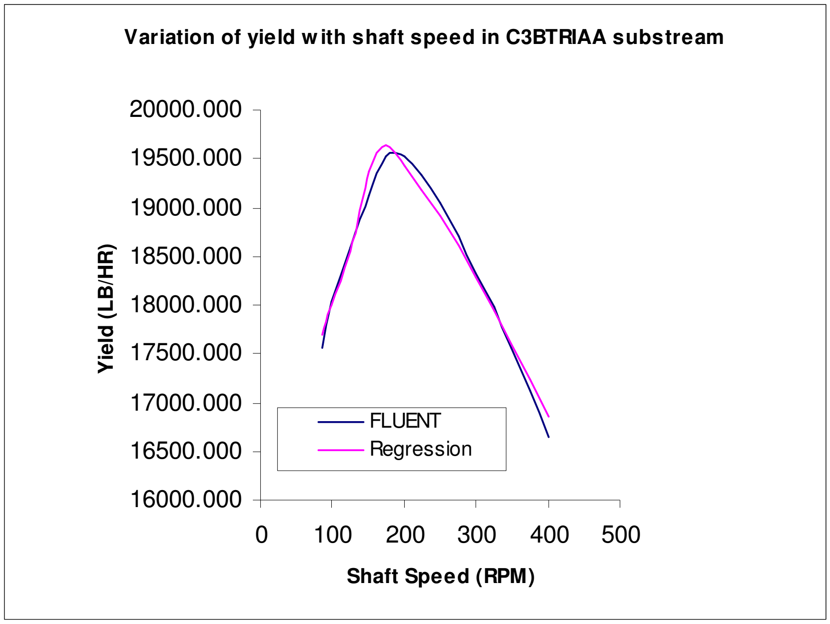


Figure 5.7: Comparison of the variation of product yield with shaft speed using FLUENT and Regression models

Viewed from a practical engineering standpoint, there is an acceptable level of agreement between the two solvers. Also worthy of note is that there is an order of magnitude difference in the time taken by the two solvers to simulate the unit. This difference will naturally increase with dimensionality and the number of cells in the CFD model.

Another feature of the current implementation is the ability to throw an “insufficient data set for interpolation” exceptional condition when the model detects that the value of an incoming variable is out of the range of the stored pre-computed results set. Thus, rather than carry out an unacceptable level of extrapolation, which generally degrades the quality of the predicted solution, the IC catches the exceptional condition, and in response, it automatically switches simulation of the current block to the next available solver.

In such cases, if the next available solver is FLUENT, then on return of control from FLUENT to the IC, a new stream record (inlet and outlet stream data set) is appended to the database before embarking on the next outer iteration. This functionality helps to enrich the database of pre-computed results, which in turn improves on the fidelity of the ROM.

The recommended procedure for using the Regression model is as follows:

1. Activate the option to “Save pre-computed results?” in the **Solvers** tab of the Controller Model Edit GUI as shown in Figure 5.8, and run some precursor simulations using FLUENT to generate the input data required by the model. Activating this option will ensure that converged CFD results are written to a

database.precomputed file in the Fluent.precomputed sub-directory, which should be created in the model database (see 14.2 for a sample).
2. Plot the results of the sensitivity analysis using Aspen Plus and the FLUENT solver.
3. Run the Custom Model Configuration Wizard to generate the required solver XML and Collections files – Regression.XML and Collections.dat respectively (Figure 5.9).
4. Add to the Collections.dat file, the collection of ports and parameters with the exact same (case sensitive) name as employed in the FLUENT simulations.

If the plots of the sensitivity analysis using FLUENT show a non-linear variation of an input variable with respect to an output variable, then the piecewise linear extension of the model should be employed by creating a piecewise interval variable named <parameter name>-pwl, i.e., udf/shaft_speed-pwl. The addition of this parameter provides a hint to the model that the piecewise linear extension should be employed in simulating the unit. In setting the value of this parameter, the sensitivity plot obtained from the FLUENT simulation should be examined to identify the commencement of a distinctive profile. For example, in Figure 5.6, a parabolic profile can be discerned at a shaft speed of 150 rpm and above. Thus, the value of the udf/shaft_speed-pwl should be set to 150, with an access mode of read-write since the parameter is not a predicted quantity.

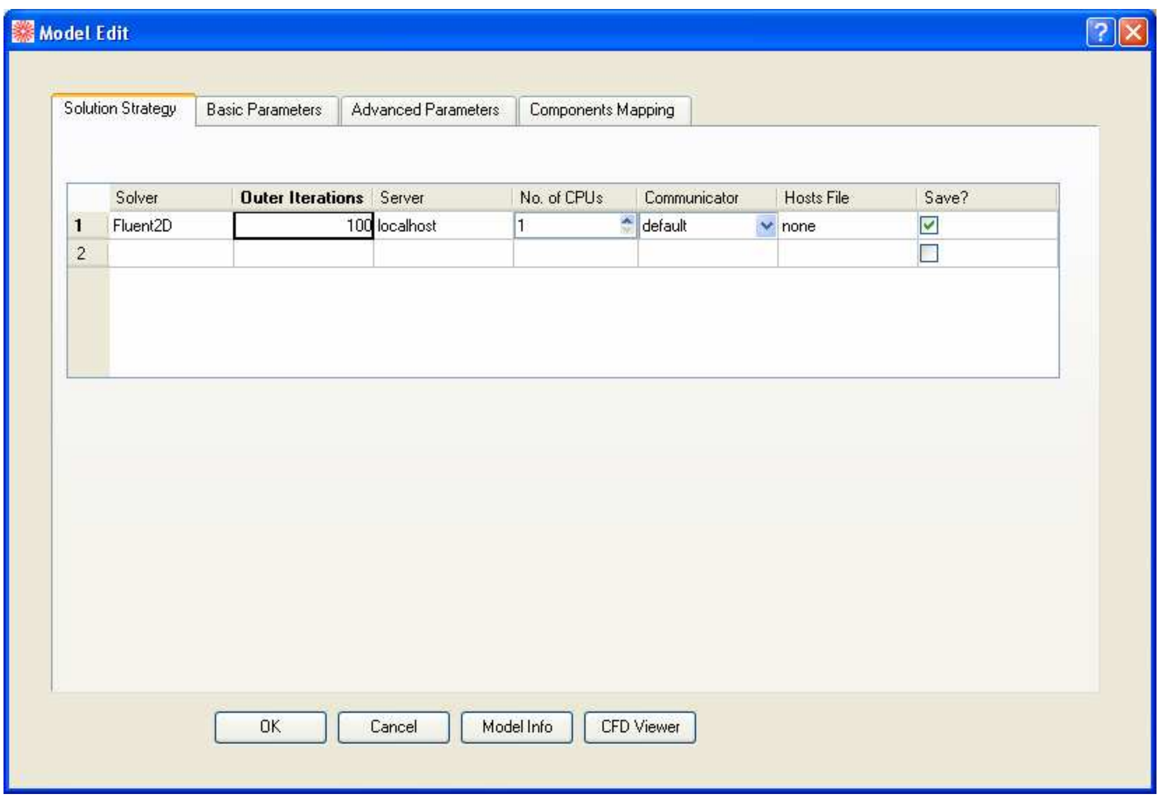


Figure 5.8: Dialog showing the option to save pre-computed FLUENT results to the database

5.1.3.7 Transfer of physical properties

To transfer pure component physical properties from Aspen Plus to FLUENT, the material properties are expressed as polynomials in temperature. The computational procedures implemented for the regression model were modified and incorporated in the FLUENT CO wrapper code to permit automatic generation of polynomial coefficients. The FLUENT solver evaluates the polynomial using the generated coefficients and cell temperature to determine the thermo-physical property of a given material in each finite-volume cell.

The addition of this capability was necessary because while Aspen Plus has a built-in capability for generating coefficients for temperature-dependent stream properties, the generated functional forms are inconsistent with those employed by the FLUENT solver. Temperature-dependent polynomial coefficients are generated for specific heat, viscosity, thermal conductivity and density, if the option to transfer temperature-dependent properties is specified by the user in the Basic Parameters tab of Model Edit GUI.

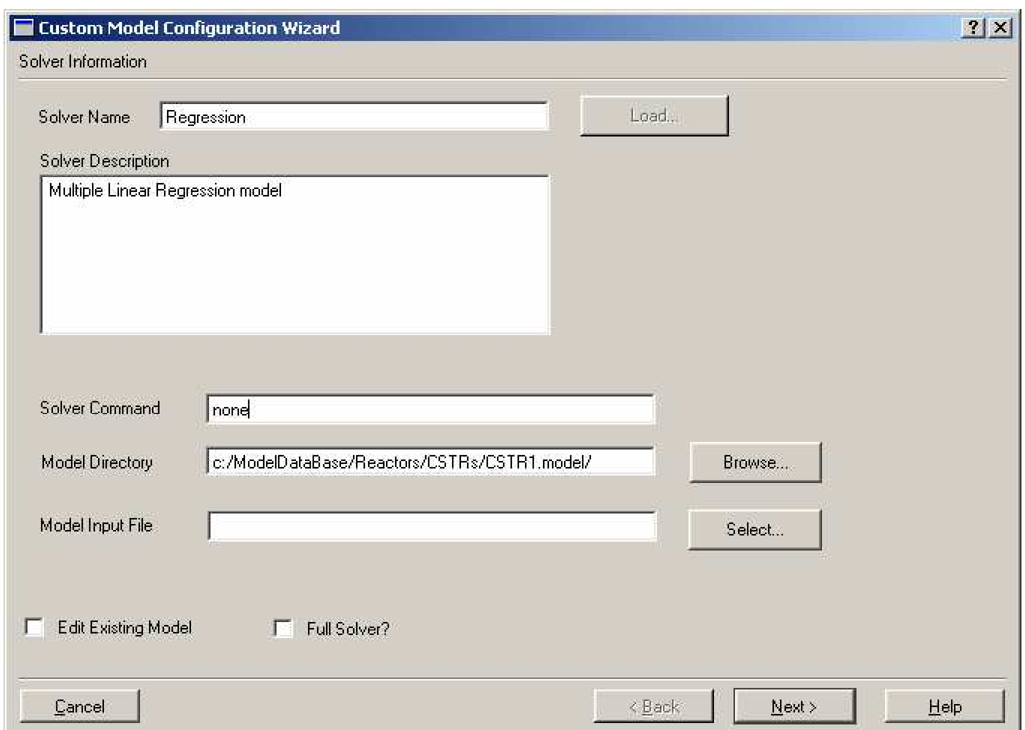


Figure 5.9: Dialog showing the use of the wizard to configure a Regression model

Whereas FLUENT accepts as inputs the linear, polynomial, piecewise linear, and piecewise polynomial coefficients for temperature-dependent properties, Aspen Plus employs wide ranging functional forms for the evaluation of material properties. Both forms should, however, evaluate to the same or closely matched values. We present sample comparisons below to demonstrate reliability of the material properties data that are transferred to the FLUENT via the IC at run-time.

In Figure 5.10 through Figure 5.13, the variation of the specific heat, thermal conductivity, viscosity, and density over a temperature range of 273K – 5000K for hydrogen gas at 290 kPa as computed by Aspen Plus are compared with the results of using the polynomial coefficients that are generated by FLUENT CO wrapper. The polynomial coefficients are generated using sampled data from Aspen Plus, which are obtained by calling the “GetProp” method of the ICapeThermoMaterialObject interface implemented in Aspen Plus thermo-physical properties package. The data accumulated over the user-specified temperature range is then employed as input to the generating algorithm.

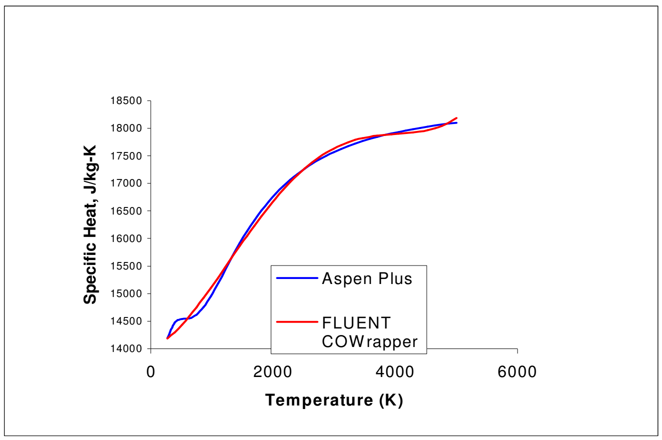


Figure 5.10: Comparison of the calculated specific heat of hydrogen versus temperature, using Aspen Plus and FLUENT CO wrapper

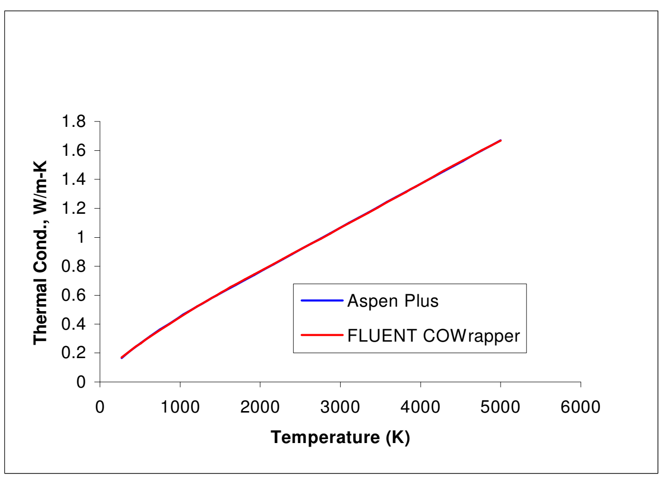


Figure 5.11: Comparison of the calculated thermal conductivity of hydrogen versus temperature, using Aspen Plus and FLUENT CO wrapper

The results of similar comparisons for liquid water are shown in Figure 5.14 through Figure 5.17 at a pressure of 101.342 kPa, and a temperature range of 273-373 K. In some of the plots, the two sets of results match to the extent that Aspen Plus results are completely overlaid by the results of FLUENT CO wrapper. It is important to note that "plain" polynomial coefficients fits are incapable of faithfully predicting property profiles with discontinuities. In such cases, a piecewise polynomial or piecewise linear fit would be more appropriate, where separate coefficients are generated for each continuous segment of the profile. The project team intends to incorporate this additional functionality in a future enhancement of the wrapper.

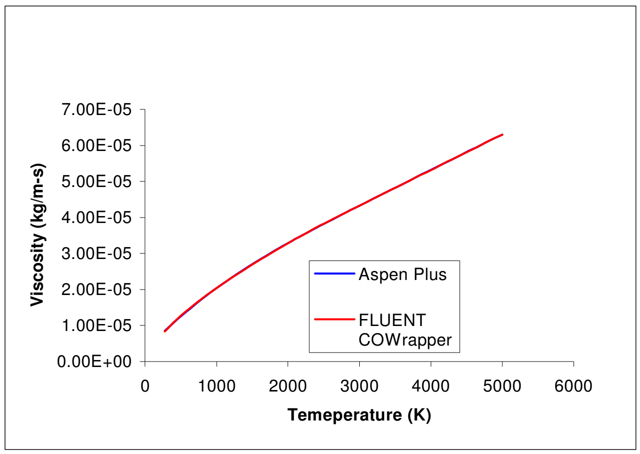


Figure 5.12: Comparison of the calculated viscosity of hydrogen versus temperature, using Aspen Plus and FLUENT CO wrapper

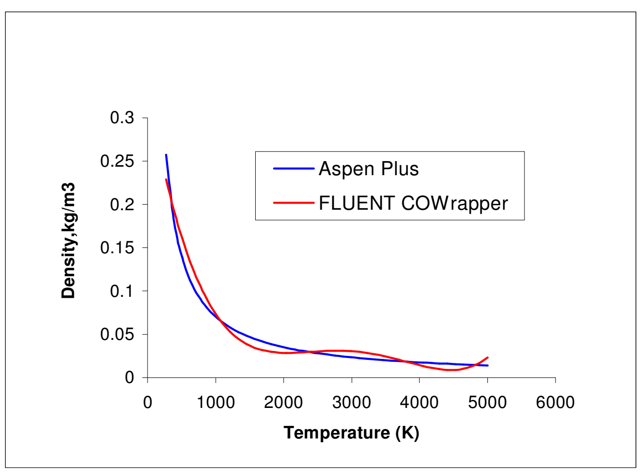


Figure 5.13: Comparison of the calculated density of hydrogen versus temperature, using Aspen Plus and FLUENT CO wrapper

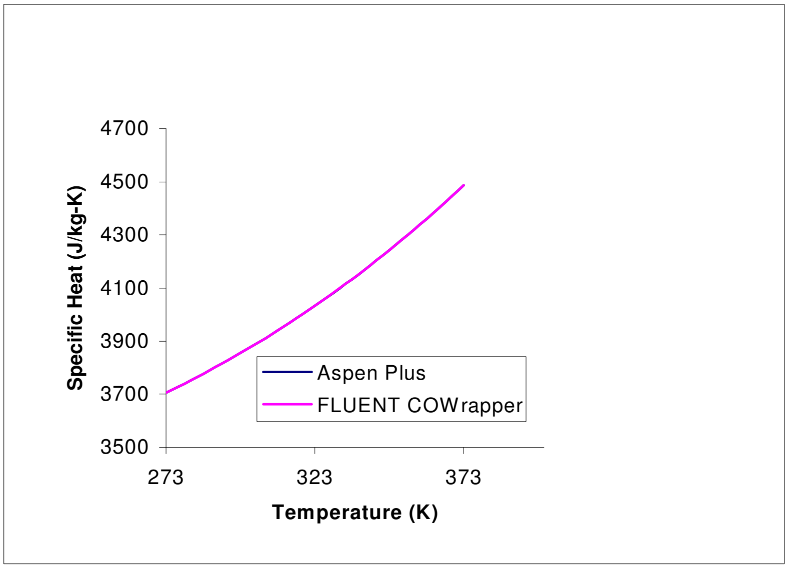


Figure 5.14: Comparison of the calculated specific heat of liquid water versus temperature, using Aspen Plus and FLUENT CO wrapper

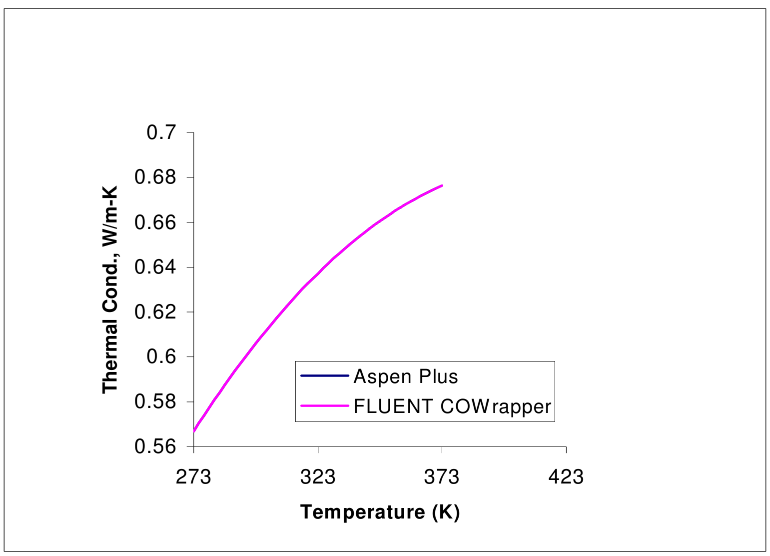


Figure 5.15: Comparison of the calculated thermal conductivity of liquid water versus temperature, using Aspen Plus and FLUENT CO wrapper

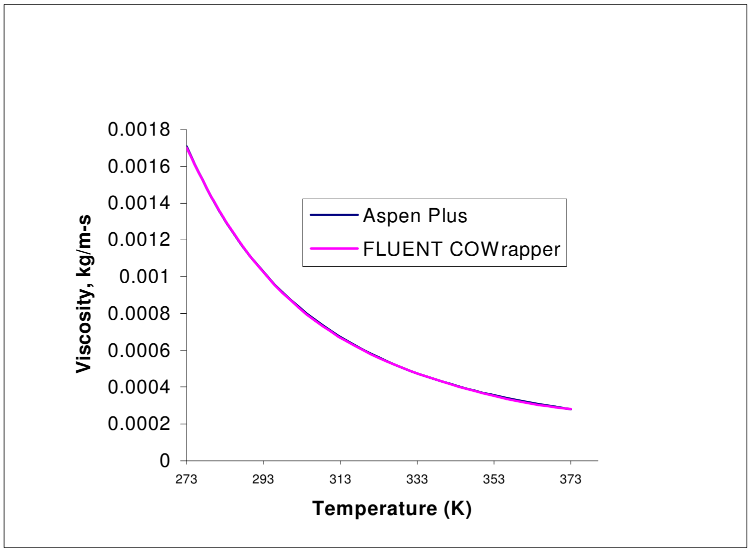


Figure 5.16: Comparison of the calculated viscosity of liquid water versus temperature, using Aspen Plus and FLUENT CO wrapper

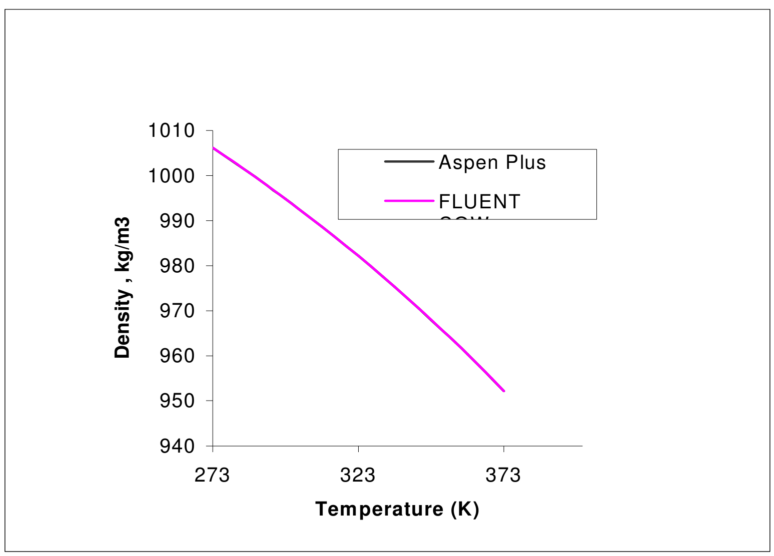


Figure 5.17: Comparison of the calculated density of liquid water versus temperature, using Aspen Plus and FLUENT CO wrapper

5.1.3.8 Remote execution

The plant designer may employ remote simulation of equipment models for a variety of reasons. The IC is equipped to handle remote simulation of equipment models in two ways: local-area network (LAN) and virtual private network (VPN). However, since the issue of network security is not addressed in the current implementation of the framework, we recommend that the integration toolkit be employed in a Local Area Network (LAN) environment only. A common "distributed computing" scenario is to access a cluster of commodity processors in a LAN for compute-intensive operations. Another possible reason to employ a remote simulation model is when an equipment model is only available in a different geographic location and cannot be made available locally due to data security constraints.

The capability currently requires that Port Number 44556 be dedicated and opened up to external traffic on the remote server machine in order to make communication with the process simulator feasible. In cases where FLUENT is being employed for the remote equipment simulation, Port Number 36000 must also be available on the remote machine for the registration of the Fluent CAPE-OPEN launcher. For cross-domain or cross-network coupling where there are firewall restrictions, the full range of ephemeral port numbers (usually from 32000 to 65556) on both the machine hosting the process simulator and the remote machine must be opened up to external packet transmission by making the required changes to the network routing table.

Although there are indications that this rather insecure requirement to open up the full range of ephemeral ports for cross-domain and cross-network connections may be circumvented by the use of CORBA bi-directional General Internet Inter-ORB Protocol (GIOP), the GIOP option has not been pursued in this project. For external solvers other than FLUENT, the only "listening" port number that must be available is 44556. All other required port numbers can be selected at the discretion of the solver developer after due consultation with the systems administrator so as to avoid conflict with the already assigned ports numbers on the remote machine.

The CORBA Implementation Repository (IR) service process is required for remote execution; it is spawned on Port Number 44556. The IR spawns the Fluent CO launcher on Port Number 36000, using the remote solver information that is registered with it. The IR essentially serves as a broker; its principal task is to return a valid ICapeUnit CORBA object reference of FLUENT or custom model solver to the process simulator so that all requests can be made on the ICapeUnit proxy reference via the Controller.

To connect with the IR, the user specifies the IP address or domain name of the remote machine in the server address column of the Model Edit GUI, i.e., www.wv.fluent.com or as shown in Figure 5.18. Note that there are no requirements to relax existing firewall rules if both the local machine hosting the process simulator and remote solver machines belong to the same LAN.

The following steps are required to set up the remote server:

1. Build a server executable (.EXE), which either implements the complete set of interfaces specified in the CORBA CAPEOPEN100::Business::UnitOp::Unit namespace, or a lightweight executable that only implements the ICapeUnitFactory interface in the same namespace. In either case, the ICapeUnitFactory interface must be implemented to handle the launching of the remote solver process using the CreateUnit method of the ICapeUnitFactory interface.
2. Build a lightweight executable, which only implements the ICapeUnitFactory interface, if a separate CO wrapper DLL is loaded at run-time by the external solver process. FLUENT employs this loading mechanism for its CO wrapper library.
3. Carry out a custom install of the Controller, which only installs the folder containing the program files required for setting up and running the remote CORBA server.
4. After carrying out the above installation steps, type "coimr <machine name> &" at the command prompt, and hit the return key to install the Implementation Repository. The <machine name> is the fully qualified Domain Name of the server machine.
5. Next, type "coservice <machine name> -i" at the command prompt, and hit the return key to register the server executable with the IR database. The IR is installed to listen to client requests for the added CO service on port number 44556.

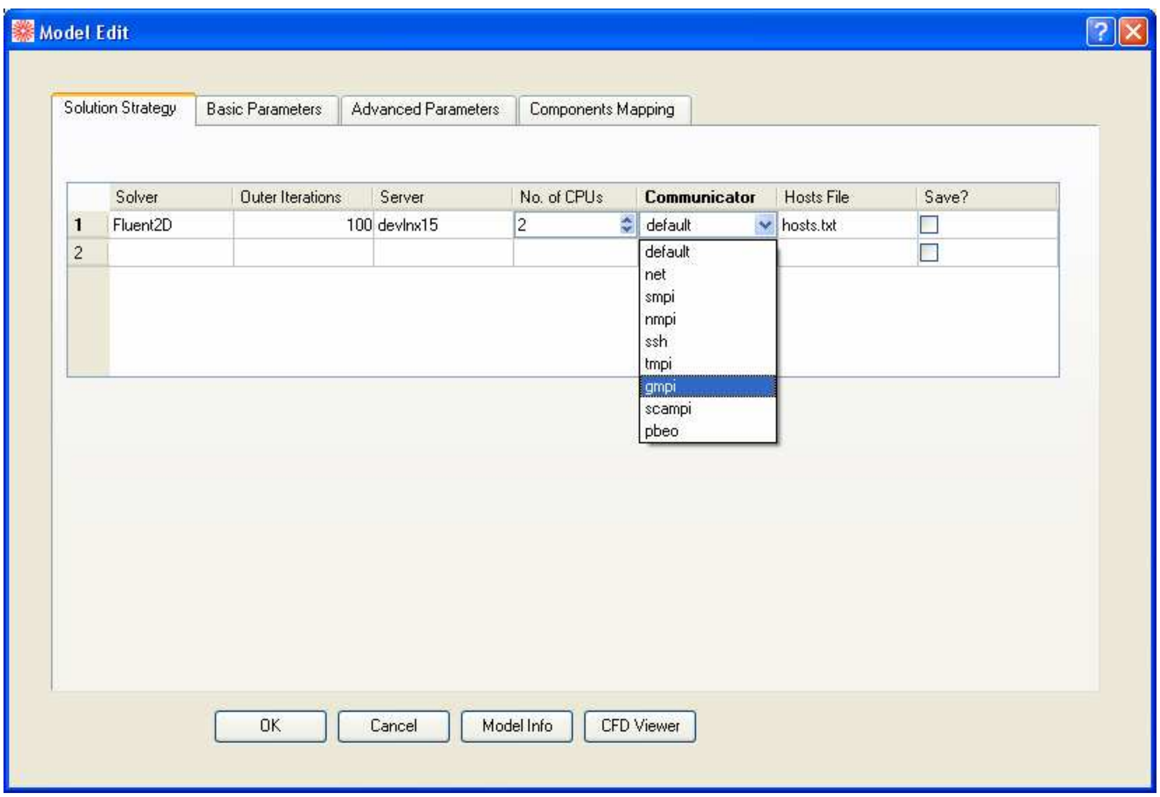


Figure 5.18: Dialog showing the specification of server address in the Solution Strategy tab of the Model Edit GUI

The build procedure is unnecessary if the remote solver is FLUENT, since the integration toolkit is bundled with remote simulation capability on both Windows and LINUX.

In Figure 5.18, the remote server name is shown as “devlnx15”. The simplified server name maps internally to the so-called four-dotted decimal address. For example, devlnx15 may map internally to the dotted address: 12.123.146.20, which is resolved internally by the IR. Also shown in Figure 5.18 are the columns for the specification of the number of CPUs and the inter-process communication protocol that is employed for parallel execution. As shown in the drop-down list in Figure 5.18, the gmpi communication protocol is being selected, which is the publicly available version of the message passing MPI library for parallel execution

We found it necessary to modify the ICapeUnitFactory interface specification by adding a couple of string arguments to the “CreateUnit” method of the interface to allow the specification of the solver command and launch directory. A recommendation based on this modification has been put forward to CO-LaN for possible inclusion in the next release of the CORBA IDL specification document.

5.1.3.9 CFD Viewer

The CFD Viewer functionality is designed to enable the process analyst to view the distribution of FLUENT calculated quantities within the equipment that is being simulated. The Viewer is launched “on-demand” in the process flow sheet environment by left clicking on the CO block on the flowsheet to pop up the Model Edit GUI. The CFD Viewer can be used to display the results of both local and remote FLUENT unit operation models in graphic formats. The results that can be displayed are the velocity vectors and contours of velocity, species, temperature and pressure, custom field functions, and many other FLUENT-computed quantities. The residuals plots can also be displayed in the Viewer to monitor the convergence characteristics of the simulation.

Upon the user selecting a quantity such as velocity vectors for display in the CFD Viewer, the FLUENT CO wrapper generates a set of pre-defined CFD results, which depict the selected view of the equipment in JPEG image file format. The image file generated by the Viewer is then packaged in a binary format by the wrapper and is transferred to the IC via the CORBA middleware. The graphics displayed in the Viewer corresponds to the quantity selected by the user in the menu options provided in the Viewer.

The view of the equipment to be displayed (which may be a planar, perspective or isometric view) must be specified *a priori* using the FLUENT Configuration Wizard as described in Section 6.1.3, and as specified in the FLUENT case and data files. Currently, multiple views of a single display “scene” for each equipment model can be displayed in the Viewer. This limitation of a single display scene is tied to the number of scenes that can be persisted in the FLUENT case/data files. Since the FLUENT CO wrapper is multithreaded (currently only true on the PC), the process analyst interested in more sophisticated graphical results can interrupt FLUENT at run-time (at the end of the cycle of FLUENT iterations) to use its richer set of graphics display capabilities. The Viewer has a print capability that allows the user to generate hardcopies of the graphic files on a connected printer.

5.1.3.10 Proprietary Model Wrapper

A couple of CORBA proprietary wrapper templates are included in the installation CD-ROM, which are intended for use by the power user who intends to couple an external unit operation model with Aspen Plus or other CO-compliant process simulator. Each Visual C++ wrapper template is pre-configured for an executable output and is applicable for developing both a remote custom equipment model within a LAN or VPN and an equipment model that is co-located with the process simulation executive.

Full details of using the supplied templates are contained in the IC User's Manual [5] and will not be reiterated here. We, however, recommend writing a small test-harness for the purposes of verifying the operation of the proprietary model before coupling the developed model with the process simulator. For the purposes of using the wrapper templates to develop a custom model, the ICapeUnitImpl.cpp file, which is available in the source directory of the supplied wrapper template project, should be implemented by the user. Because it is impossible to determine *a priori* the architecture or execution path of every conceivable custom model, detailed guidance about implementing the methods of the ICapeUnit interface is not feasible. Of course, any number of helper functions and variables may be added to the template implementation files with the understanding, however, that they are not callable from the process simulation environment since they are not part of the CO CORBA IDL specifications.

The Custom Model Configuration Wizard should be employed to generate the required XML and collections files after successfully creating the required EXE for the custom model. For example, for the regression ROM, which is included in the installation program, a Regression.XML and Collection.dat file were created for the CSTR test problem. Sample data input format in the Collection.dat file for the CSTR model, which is specified for using the wrapper template, is as follows:

```
PARAMETER REAL shaft-speed 100.0 READ-WRITE
PORT MATERIAL mass-flow-inlet-6 INLET NONE domain-1
PORT MATERIAL pressure-outlet-7 OUTLET NONE domain-1
```

The first entry in each of the above lines indicates the type of collection item, e.g., a material port or a parameter. For the parameter type collection, the second entry specifies the data type, while the third, fourth, and fifth entries in the row specify the parameter name, value, and access mode, respectively. For each port item, the second, third, fourth, and fifth items are the values for the port type, port name, port direction, and the flow domain to which the port belongs, respectively.

The current version of the wrapper template requires some level of familiarity with CORBA middleware technology in order to successfully use it for any code coupling effort. In addition, experience with its use indicates that the pre-computed results database module of the IC is not robustly coupled with the wrapper template due to the use of a flat file system for archiving the results generated by FLUENT. A known weakness is that the flushing of the FLUENT pre-computed results buffer to the pre-computed results file lags the update of the ROM, which supposedly makes use of the updated data. In addition, the flushed results are sometimes not written out in the expected formats, thereby generating run-time error conditions. However, these limitations related to the use of pre-computed results do not occur if the custom model does not require input data from the pre-computed results file, or when an updated pre-computed results file is not required by the custom model at run-time. In future work it is envisioned to replace the current flat file system for storing results and equipment models with a robust relational database.

5.2 Excel spreadsheet integration

The integration toolkit is bundled with a Microsoft Excel “add-in” macro that allows the user to run FLUENT equipment models from within the spreadsheet environment. In this scenario, Excel takes on the role of the simulation executive. The FLUENT model may be simulated on the same machine as Excel or on a remote machine. This capability allows a CFD expert to create the FLUENT model and “package” it for broader use by others via Excel. Developed using Visual Basic for Applications (VBA) and C++/COM, it leverages the IC COM-CORBA bridge sub-system to effect inter-process communication between Excel and FLUENT.

The current version of the link has only been thoroughly tested on the Windows platform. It has led to modifications of the FLUENT CO wrapper in order to provide sufficient generality to run many types of FLUENT models. The link presents the user with the familiar GUIs of the IC, including the CFD Viewer, within the spreadsheet environment. The implementation provides for a “Fluent Link Tools”, which is available in the form of a “dockable” floating toolbar, as well as an optional “Fluent Link” menu item as shown in Figure 5.19. The menu item becomes visible upon activating the Excel-FLUENT “add-in”.

Upon initiating a coupled Excel/FLUENT simulation, data for the unit operation ports and parameters are displayed on the active spreadsheet so that the user may edit their default values. The Run button shown in Figure 5.19 is clicked to start the FLUENT iteration after making all the required changes to the model data on the active sheet. The FLUENT Link button (also shown in Figure 5.19) is clicked to display the CFD Viewer within the spreadsheet environment. A Terminate button (also shown in Figure 5.19) is available to shut down the FLUENT process after running the simulation. Further information on activating and using the Excel-FLUENT capability is documented in Section 4.5 of the User’s Manual [5].

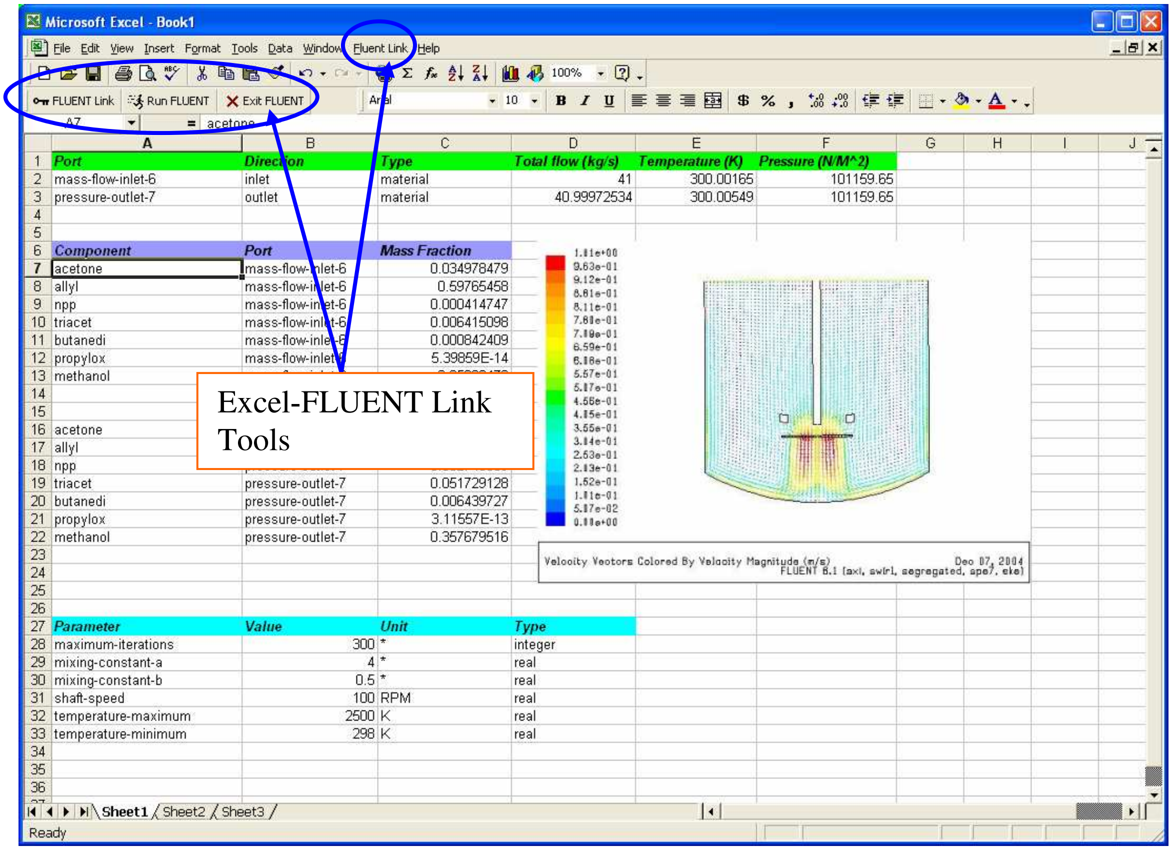


Figure 5.19: Excel spreadsheet showing the inputs and results of running the FLUENT 2-D CSTR test problem

5.3 Error detection, reporting, and recovery

To improve on the robustness and usability of the Integration Controller, various error detection and possible recovery mechanisms were implemented. Specifically, the critical CAPE-OPEN error interfaces including the exception thrown by the ICapeUnit interface as specified in the CORBA IDL were implemented in the FLUENT CO wrapper library. The C++ “try/catch” blocks and “throw” programming language features were implemented to handle function invocations and reporting of exceptional conditions respectively. Exceptional conditions such as divide by zero, inability to read in the FLUENT case/data file, and unavailability of network connections, are faithfully propagated to the Controller environment.

For an illustration, as shown in Figure 5.20, the Controller can recover from an error condition of being unable to read in a FLUENT case/data file, provided the FLUENT case and data file are subsequently made available in the directory specified in the solver XML file before a re-try is attempted in the process simulation environment.

Figure 5.20: Sample error dialog reporting the non-availability of the FLUENT case/data files as specified in the solver XML file

The FLUENT wrapper library generates a log file named cowrapper.log in the solver sub-directory in the user's model database. The log file is particularly useful for "post-mortem" analyses to identify which operation was last successfully executed, as well as the failure messages leading to the program crash. We have found the information generated in the log file to be useful for correcting specification problems for subsequent runs, and in a few cases, it has informed the decision to re-implement "buggy" class methods. The information logged in the wrapper log file includes the FLUENT model description, the machine name on which the simulation was run, the run date, the total number of outer (i.e., Aspen Plus) iterations, etc.

The IC was also enhanced by modifying the implementation of the critical methods of the COM implementation classes to include C++ "try/catch" and "throw" code segments to catch and report on the exceptional conditions that are being generated within the IC code, and from the active external unit operation models. In addition, the IC SolverManager and SessionManager classes that handle solver activation were enhanced to detect and report errors and/or exceptional conditions resulting from unsuccessful activation of the remote solvers due to network problems, as shown in Figures 5.21 and 5.22.

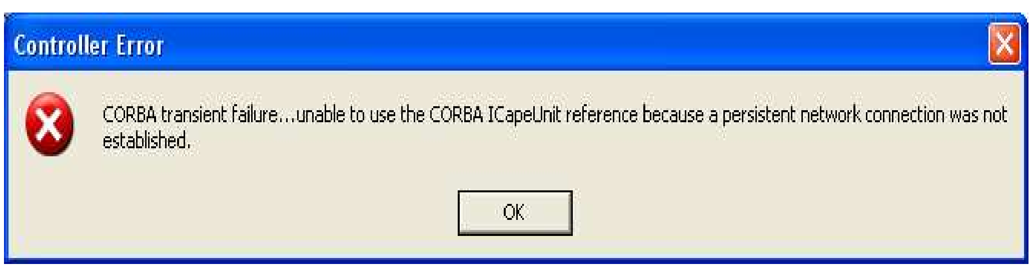


Figure 5.21: Error dialog reporting unavailability of the remote solver process, which is encapsulated in the ICapeUnit CORBA object reference

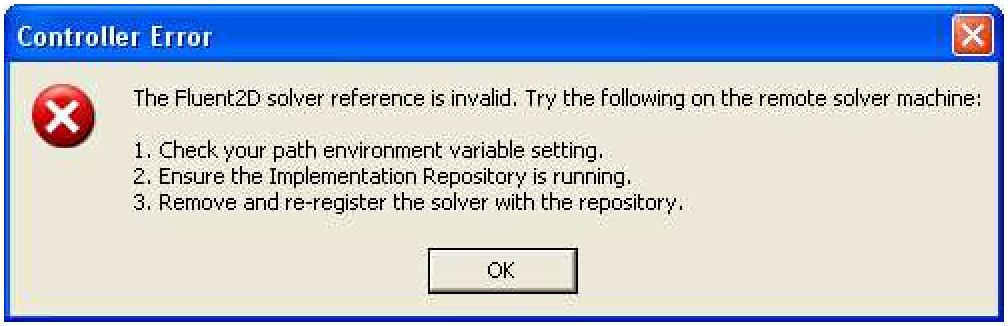


Figure 5.22: Error dialog reporting a bootstrapping problem due to unavailability of the FLUENT launcher, which is registered with the Implementation Repository

Preliminary research was conducted to explore how to ensure numerical convergence for coupled Aspen Plus/FLUENT simulations without manual interruptions to modify the FLUENT solver settings. A Scheme (programming language) code was tested with the FLUENT wrapper library that can automatically adjust the FLUENT under-relaxation factors. Although this feature was not fully implemented due to lack of resources, the approach was successful and would be highly useful to novice CFD users. If included in a future release of the IC, it will be associated with a Boolean parameter (automatic-solution-steering) that the user can activate in the Model Edit GUI. Upon activation, the FLUENT wrapper will dynamically adjust the under-relaxation factors during the simulation without any user intervention. The adjustments are based on Fast Fourier Transform analysis of the convergence history of the solution residuals. Experience showed that the capability was able to "guarantee" convergence without user intervention, although usually at the expense of a higher iteration count and CPU time. Nevertheless, from the perspective of the non-expert user, the benefits of automatic solver steering far outweigh the known drawbacks.

5.4 Software deployment

A User's Manual for the integration toolkit [5] was prepared with thorough instructions on software installation and administration. An outline of the topics discussed in the manual is as follows:

- Software Installation
  - Installation requirements
  - Installing the integration software
  - Adding a model library to the database
- Usage of Configuration Wizards
  - Using the FLUENT Configuration Wizard to make a CFD model CO-compliant
  - Using the Custom-Model Configuration Wizard to make a custom model CO-compliant
- Graphical user interfaces of the Integration Controller
  - Requirements for using the Controller GUIs
  - Model Selection GUI
    - Accessing the model selection GUI
    - Adding a model database to the IC environment
    - Removing a EM database
    - Obtaining model information
    - Selecting an equipment model from the EM database
  - Model Edit GUI
    - Defining the solution strategy
    - Setting basic parameters
    - Setting advanced parameters
    - Mapping the chemical components
    - Using the CFD viewer
- Advanced features
  - Aspen Plus sensitivity analysis and optimization based on equipment model parameters
  - Including a custom equipment model
    - Build requirements
    - Using the CORBA wrapper template
    - Starting a remote server
    - Running the remote custom model
    - Collocated server
  - Guidelines for accessing CAPE-OPEN parameters from FLUENT files
    - FLUENT native variables (RPVAR)
    - Text user interface
    - Writing Scheme functions
    - Specifying physical model ports in a coupled simulation
    - Extracting stream data from physical model outlet ports
  - LINUX installation

6 Integrated Model Development and Usage

The integration approach used here is for Aspen Plus and FLUENT to model different unit operations and exchange stream information at flow boundaries (see Table 6.1 for the details of the information exchanged). Integrated system-level and equipment-level simulations offer new opportunities to design and analyze overall power plant performance with respect to fluid flow, mass and heat transfer, chemical reactions, and related phenomena.

Table 6.1: Information transferred between Aspen Plus and FLUENT models

From Aspen Plus to FLUENT

- Chemical species names
- Constant properties: standard state enthalpy, entropy, and molecular weight
- Pure component physical properties as a function of temperature: density, specific heat, viscosity, and thermal conductivity. (The mixture properties are calculated internally in FLUENT using user-selected mixing laws).
- Reaction stoichiometry and power-law parameters
- User defined CFD model parameters
- Stream data at inlets: total flow rate, species mass fractions, temperature and pressure

From FLUENT to Aspen Plus

- Stream data at outlets: total flow rate, species mass fractions, temperature, and pressure

Using the Vision 21 environment, the creation of an integrated model involves four key tasks: 1) system-level model development, 2) equipment-level model development, 3) equipment-level model configuration and 4) model integration (Table 6.2). In one typical scenario, the process design engineer develops the system-level model (Task 1) and combines it with the equipment-level models (Task 4), while the equipment design engineer develops the equipment-level models (Task 2) and configures them for use in the system-level model (Task 3).

Tasks 3 and 4 constitute the model integration activities. The time required to complete these tasks has been reduced significantly by the technology developed in this project. Tasks 1 and 2 represent traditional system-level and equipment-level modeling activities and, depending on plant size and complexity, can be time-consuming processes. Even so, the integration of such models using the technology described here will help reduce the number of very costly Vision 21 demonstration plants, decrease the plant development time, reduce costly mistakes in their design, and improve their operating performance.

Table 6.2: Key tasks for integrated model development

Task	Performer	Tools	Time before V21	Time after V21
1. Develop system-level model	Process engineer	Aspen Plus	Hours to weeks, depending on plant size & complexity	Same as time before V21.
2. Develop equipment-level models	CFD engineer	FLUENT, proprietary codes, ROMs	Days to weeks, depending on number & complexity of equipment items	Same as time before V21.
3. Configure equipment-level models	CFD engineer	Configuration wizards	Weeks	Minutes
4. Integrate equipment- and system-level models	Process engineer	Aspen Plus, FLUENT, proprietary codes, ROMs	Days to weeks, depending on plant size & complexity	Hours to days, depending on plant size & complexity

The workflow used in the integration of CFD and process models is depicted in Figure 6.1 and in Table 6.3. The two boxes shown at the top, entitled "CFD Analysis" and "Process Analysis", are the usual steps of developing CFD and process models. The seven boxes shown below depict the steps required in the newly developed integrated environment. The boxes in yellow (or light gray) on the left-hand column are steps taken by the CFD engineer. The boxes in gold (or dark gray) in the center and right-hand column are steps taken by the process engineer. The CFD engineer develops the CFD model, uses the Configuration Wizard to make the model CO-compliant, and adds the model to the CFD model database. The process engineer sets up the process model and selects CFD models to represent one or more unit operation blocks in the process flow diagram. The Model Selection GUI comes up when the process engineer edits a CFD model icon placed on the process flowsheet. The process engineer changes parameters in the CFD model by using the Model Edit GUI. (Those are the parameters that the CFD engineer specified as editable during the configuration step.) The process engineer may also set up a solution strategy using the Model Edit GUI. The solution strategy consists of a choice of different models used to represent the unit operation block and may, for example, include CFD models (coarse grid/fine grid, 3-D/2-D), reduced order models, or proprietary models. The purpose of the solution strategy is to interchange models during the simulation to improve the overall accuracy/speed. The process engineer then conducts the integrated simulation, which is as simple as the usual step of hitting the "Start" button in Aspen Plus. After obtaining a converged solution, the process engineer views CFD results by choosing from the graphics options in Model Edit GUI. The stream information at the outlet of the unit operation block is obtained as usual from the Aspen Plus "Results…" menu.

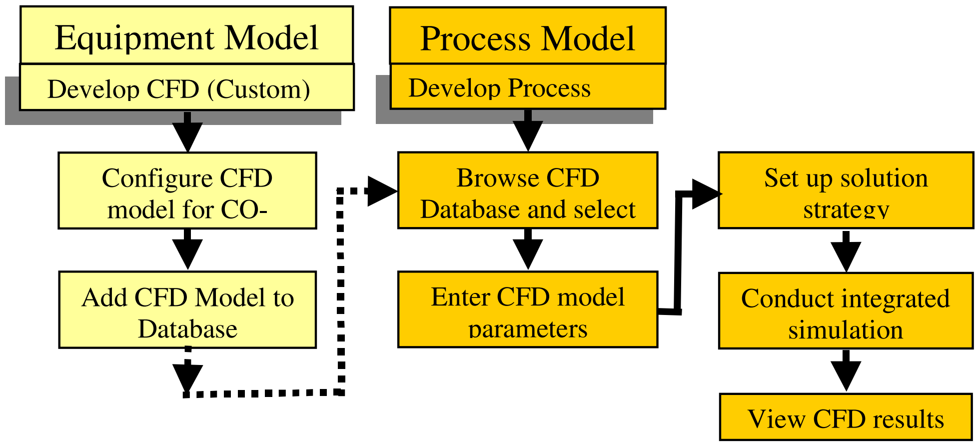


Figure 6.1: The workflow in integrating CFD and process models

To demonstrate the development and use of integrated process and CFD models, we utilize the following two examples:

1. Aspen Plus reaction-separation-recycle flowsheet coupled with a FLUENT 2-D CFD reactor model;
2. Aspen Plus fuel cell flowsheet coupled with a FLUENT 3-D CFD reformer model.

We use the reaction-separation-recycle example to describe the integrated Aspen Plus and FLUENT workflow. The second example is used to expand on several aspects of the workflow and to highlight the use of flexible solution strategies and ROMs.

In the next four sections, we discuss in some detail the tasks presented in Table 6.2, focusing largely on the integration activities (Tasks 3 and 4) for coupling a CFD reactor model into a process flowsheet.

Table 6.3: Integrated workflow using Integration Controller

- **Equipment**
  - Develop CFD model (FLUENT) or custom equipment model
  - Configure CFD model (FLUENT Configuration Wizard) or custom equipment model (Custom Model Configuration Wizard) for CO-compliance
  - Add CFD model to database

- **Process**
  - Develop process model (Aspen Plus)
  - Load CAPE-OPEN library (Aspen Plus)
  - Place Fluent block icon on process flowsheet (Aspen Plus)
  - Select CFD model from database (IC Model Selection GUI)
  - Edit CFD parameters (IC Model Edit GUI)
  - Define solution strategy (IC Model Edit GUI)
  - Connect material streams to Fluent block (Aspen Plus)
    - Map chemical components (IC Model Edit GUI)
    - Run integrated simulation (Aspen Plus)
    - View CFD results (IC CFD Viewer)

6.1 Example 1: Reaction-separation-recycle flowsheet

6.1.1 System-level model development

To demonstrate the integrated workflow, we use an Aspen Plus reaction-separation-recycle flowsheet consisting of several typical chemical process unit operations including reactors, heat exchangers, and distillation columns (Figure 6.2). Development of this process flowsheet represents Task 1 in Table 6.2.

The first reactor (PORXN), a stoichiometric reactor based on fractional conversion, is used to model a solvent-enhanced isomerization reaction, which produces feed for a second continuous-stirred tank reactor (CSTR). The Aspen Plus CSTR model uses a built-in power law expression for calculating the Arrhenius rates for the rate-controlled reactions. The downstream separation section consists of three distillation columns in series. The first column recovers solvent and recycles it back to the stoichiometric reactor. The second and third columns separate products from impurities.

The process engineer uses Aspen Plus input forms to specify the chemical species, physical properties, and reaction information used in the process simulation. All of this input information is transferred automatically from Aspen Plus to FLUENT via the CAPE-OPEN interfaces when running the integrated simulation.

In preparation for the coupled simulations, we first perform a steady-state simulation of the reaction-separation-recycle flowsheet using the Aspen Plus CSTR model. This not only generates a good starting point for the integrated Aspen Plus and FLUENT simulation, but also provides good estimates of the inlet boundary conditions for use in developing the FLUENT CSTR model.

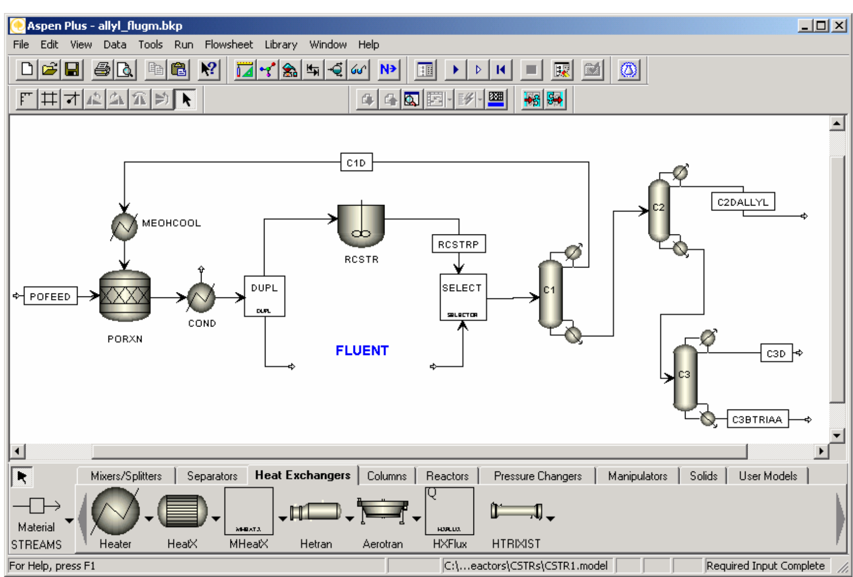


Figure 6.2: Aspen Plus flowsheet for reaction-separation-recycle example

6.1.2 Equipment-level model development

A detailed CFD model is required for the CSTR because mixing impacts the performance of the reactor. The development of the CSTR model using FLUENT represents Task 2 in Table 6.2. Given the tank style, feed location, and impeller geometry and location, we used the MixSim software (an application-specific tool developed by Fluent) to generate the computational grid used by FLUENT (Figure 6.3). The CSTR used here consists of a high-efficiency axial impeller in a baffled, dish-bottomed tank with feed injection at the impeller. FLUENT solves the fluid flow problem using this structural information along with the inlet boundary conditions, operating conditions (e.g., impeller speed), and solver settings. The impeller speed used for the base case is 100 rpm.

The FLUENT CSTR model used here considers axisymmetric swirling flow in the reactor. The standard k-ε model is used to model fluid turbulence. This is a semi-empirical model based on model transport equations for the turbulent kinetic energy (k) and its dissipation rate (ε). In regions of the reactor where turbulence levels are high, the eddy lifetime k/ε is short and mixing is fast. As a result the reaction rate is not limited by small-scale mixing of the reactants. On the other hand, in regions with low turbulence levels, small-scale mixing may be slow and limit the reaction rate.

The FLUENT CSTR model uses the finite-rate/eddy-dissipation model (for turbulent flows) that computes both the Arrhenius rate and the mixing rate and uses the smaller of the two rates. When mixing is rapid, formation of the desired product is favored. When mixing is very slow, formation of impurities is favored. Since FLUENT incorporates the effect of mixing, its prediction of yield differs from the results predicted using the perfectly-mixed CSTR model in Aspen Plus.

In preparation for integrating the FLUENT CSTR model into Aspen Plus, we use the results of the process simulation with the Aspen Plus CSTR to set reasonable boundary conditions for the FLUENT model. The corresponding CFD results are then saved in a FLUENT data file to be used as a starting point in the coupled simulation.

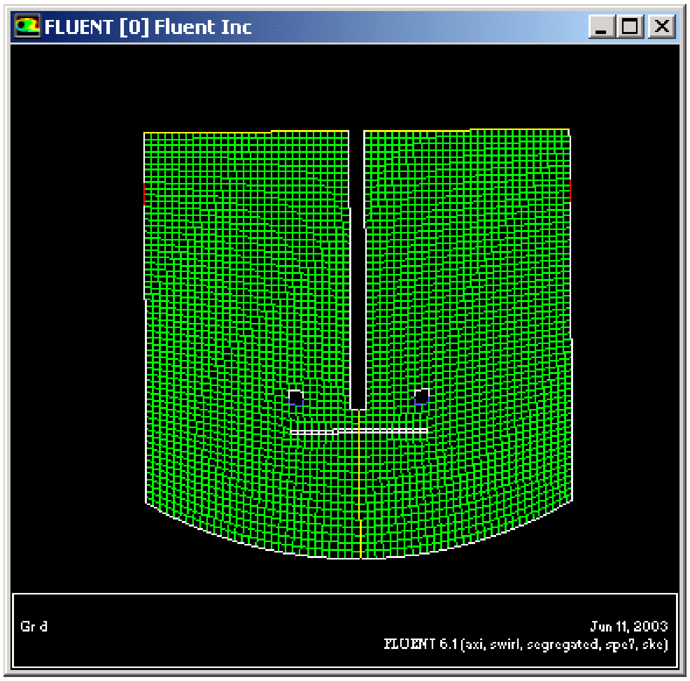


Figure 6.3: Grid display for FLUENT CSTR model

6.1.3 Equipment-level model configuration

The CFD engineer uses a configuration wizard to convert the equipment-level model into a CO-compliant model that is then stored in a EM database for use in Aspen Plus. The FLUENT Configuration Wizard is used to convert FLUENT CFD models, while a general Configuration Wizard is available for preparing equipment-level models solved by other proprietary and commercial software packages. The wizard is loaded by typing the following command at the FLUENT text user interface prompt (i.e., caret, >):

> (load "cowizard")

The FLUENT Configuration Wizard allows the CFD engineer to navigate systematically through the basic steps required (Figure 6.4). The Navigator field shows that the configuration procedure involves the following steps:

- Specifying FLUENT file and model information
- Selecting ports and domains corresponding to FLUENT boundaries made available as ports for material stream connection in Aspen Plus
- Defining basic parameters (CFD model) and advanced parameters (FLUENT solver) for access by the process design engineer in Aspen Plus

Figure 6.4: Navigator panel for the FLUENT Configuration Wizard

The first configuration step involves specifying the Model Files, starting with the FLUENT Case File (Figure 6.5). If required, the CFD user can specify a Model Scheme File that contains scheme functions used to set and get FLUENT parameters. For the FLUENT CSTR example, we use a scheme file named *cstr_params.scm* to set and get the FLUENT turbulent mixing rate constants. Finally, the Model View File contains the saved FLUENT view used to display the CFD results in Aspen Plus.

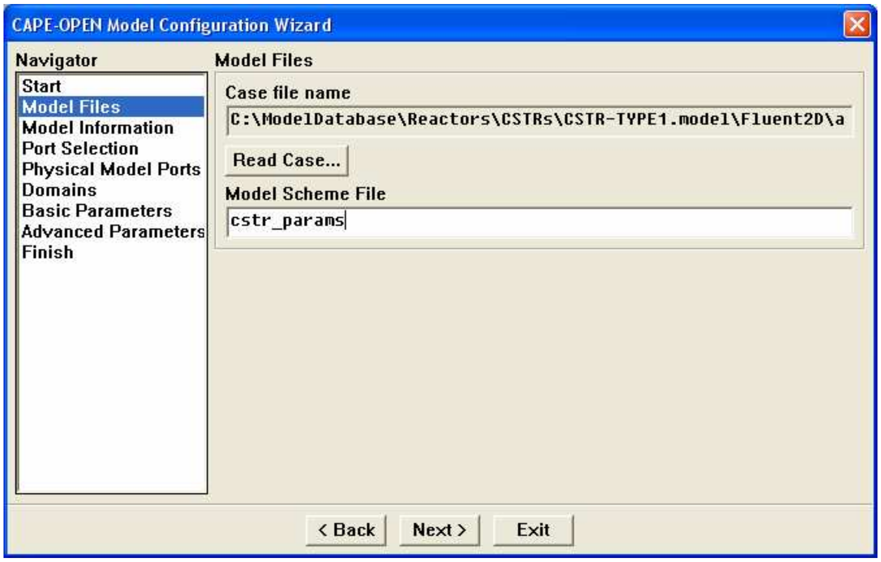


Figure 6.5: Model Files panel in the FLUENT Configuration Wizard

In the second configuration step, the CFD user specifies general model information, including a model name, description, category, and type (Figure 6.6). The category and type determine where the model is stored in the EM database. The categories shown in the drop down list correspond to model categories available in the Aspen Plus model palette (see lower portion of Figure 6.2). For the CSTR example, the model category is *Reactors* and the model type is *CSTR*.

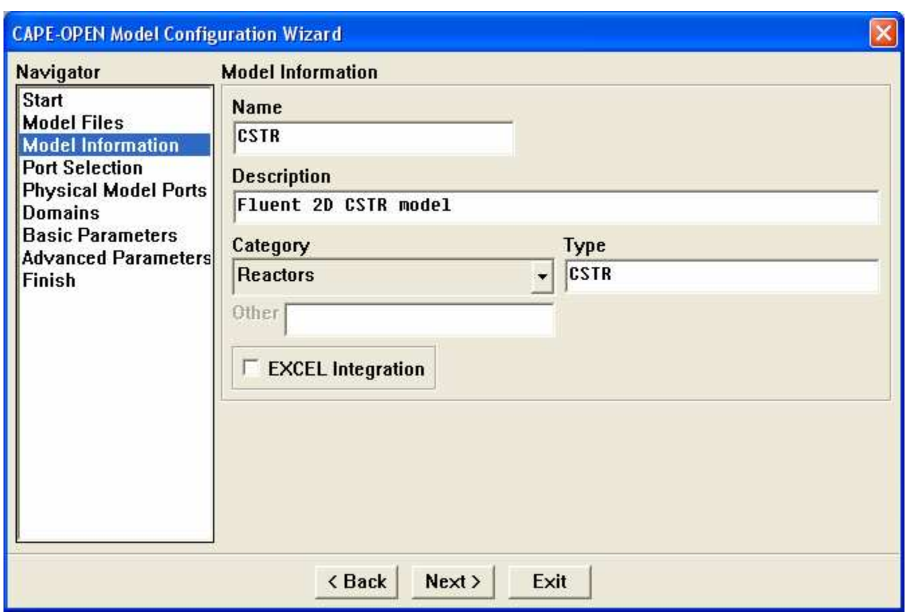


Figure 6.6: Model Information panel in the FLUENT Configuration Wizard

The third FLUENT configuration step is Port Selection (Figure 6.7). This step entails selecting the FLUENT boundary zones to make available as stream connections in Aspen Plus. The Zones field contains a selectable list of zones from which you can choose a zone of interest. By clicking on the right-arrow buttons, you can move the zone over to the Inlet Ports or Outlet Ports field. For each selected zone, you can provide a port description and bring up the FLUENT Boundary Conditions panel, which allows you to set the type of a zone and display other panels to set the boundary condition parameters for each zone. As shown in Figure 6.7, the CSTR has one inlet (*mass-flow-inlet-6*) and one outlet (*pressure-outlet-7*).

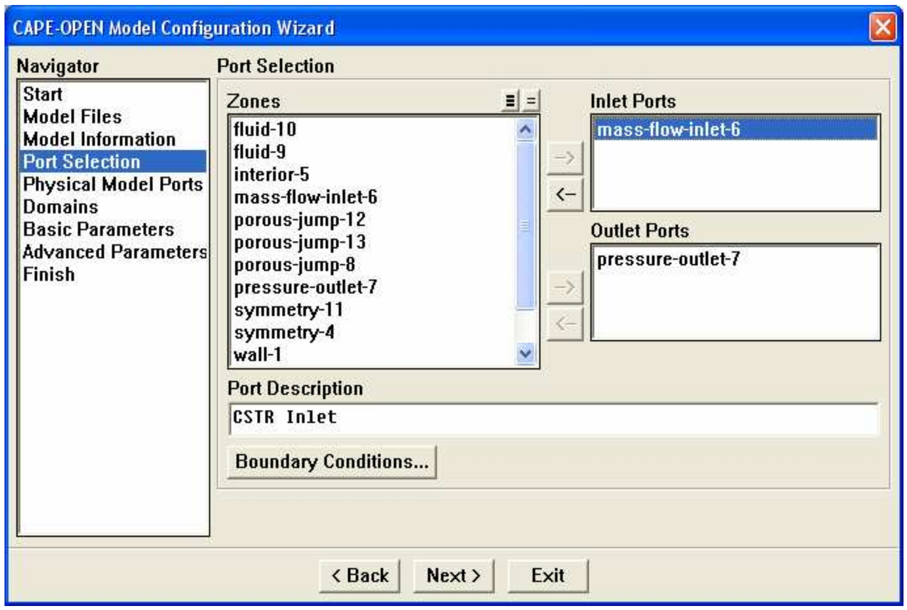


Figure 6.7: Port Selection panel in the FLUENT Configuration Wizard

The fourth configuration step involves the optional specification of the physical model ports (see Figure 6.8), which are *ad hoc* ports not based on a FLUENT zone. For example, using the physical model ports representation, the coolant stream properties of the FLUENT Heat Exchanger Model capability can be included as inlet/outlet ports on the process flow sheet. In a similar fashion, discrete particle injections in FLUENT can also be modeled as physical model ports, thereby enforcing mass balance in the flow sheet calculations in a coupled simulation. Without the use of the physical model ports, the heat exchanger coolant thermo-physical properties would have to be modeled using CO parameters, and fluid generation in a FLUENT equipment model involving discrete phase injections would be unaccounted for in on the process flow sheet model. Further details on the use of the physical model representation are contained in the IC User's Manual [5].

The fifth configuration step requires the CFD engineer to assign the inlet and outlet ports to their respective domains (Figure 6.9). In the CSTR example, the reactor represents a single domain, so we simply assign the one inlet port and one outlet port to the first domain with Domain ID of 1. For a jacketed reactor, we would have had two domains—

one domain corresponding to the reactor and the other to the jacket. In that case, we would have assigned one inlet and outlet port to each domain.

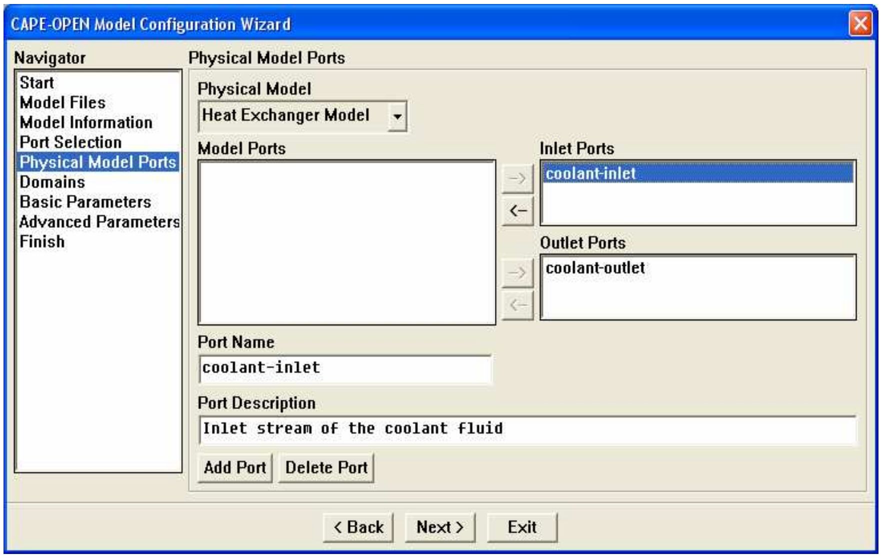


Figure 6.8: Specification of a heat exchanger coolant fluid using the physical model ports functionality

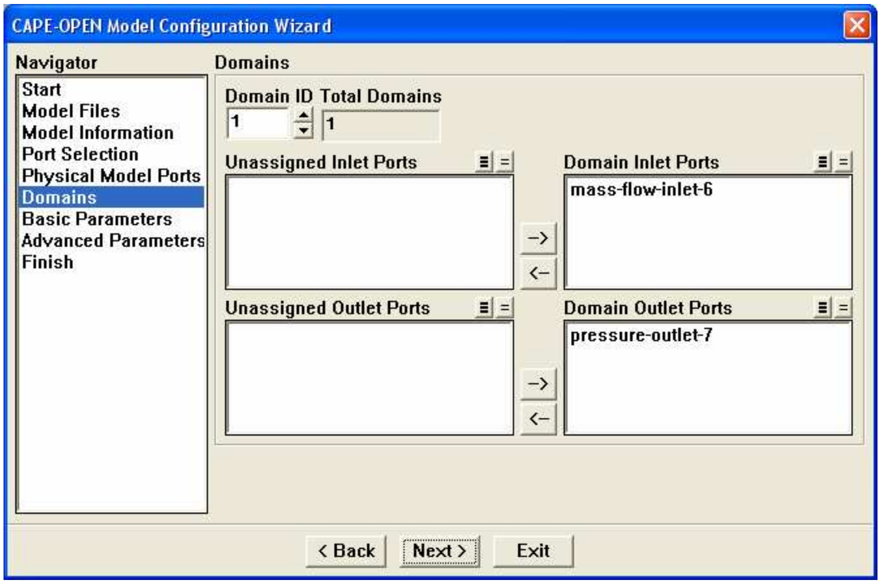


Figure 6.9 Port-to-domain assignment panel in the FLUENT Configuration Wizard

The final two configuration steps involve the CFD design engineer specifying which parameters to make available to the process design engineer for access during the model integration task. The parameters are categorized as Basic Parameters and Advanced

Parameters, where the former correspond to *model*-specific parameters and the latter to *solver*-specific parameters.

Figure 6.10 shows the Basic Parameters panel with three default parameters related to the automatic transfer of physical properties from Aspen Plus to FLUENT. The properties-transferred parameter allows you to transfer *temperature-dependent* and *constant* physical properties (i.e., density, viscosity, specific heat, thermal conductivity) from Aspen Plus to FLUENT. The temperature-maximum and temperature-minimum parameters specify the temperature range over which to calculate the coefficients for the *temperature-dependent* property functions in Aspen Plus. Selecting the *none* option for the properties-transferred parameters means that no physical property information will be sent from Aspen Plus to FLUENT. In this case, the properties specified in FLUENT will be used.

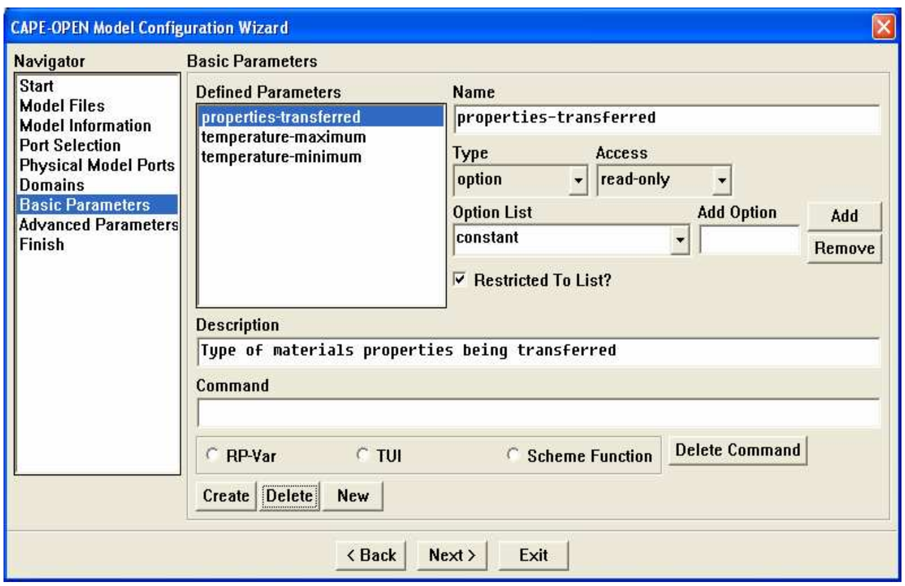


Figure 6.10: Basic Parameters for physical property transfer from Aspen Plus to FLUENT

Figure 6.11 illustrates the creation of a new Basic Parameter of type *real*, called *udf/shaft_speed*, which is specific to the CSTR example. The attributes of this parameter include Name, Type, Access, Default Value, Lower Bound, Upper Bound, Dimensionality, Description, and Command. The new *udf/shaft_speed* parameter is an rpvar, which is executed with the scheme code shown in the Command field. By selecting the appropriate radio button, you can specify the mechanism used to get and set the parameter value. The available mechanisms are rpvar, FLUENT text user interface (TUI) commands, and Scheme Function commands.

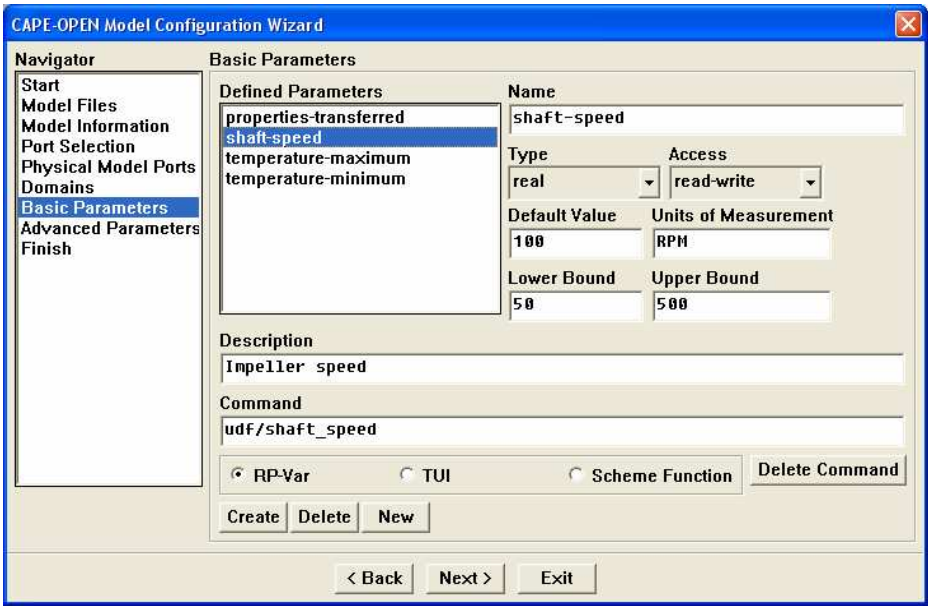


Figure 6.11: Basic Parameters for shaft speed in FLUENT Configuration Wizard

The Advanced Parameters panel allows the CFD engineer to create and configure FLUENT solver parameters (Figure 6.12). The default parameters provided include FLUENT's maximum-iterations and two turbulent mixing rate constants, mixing-constant-a and mixing-constant-b. Figure 6.12 shows an example of specifying the use of a Scheme Function to access the mixing-constant-a parameter. Recall that we specified the file containing the scheme function in the Model Scheme File field in the Model Files panel (Figure 6.5).

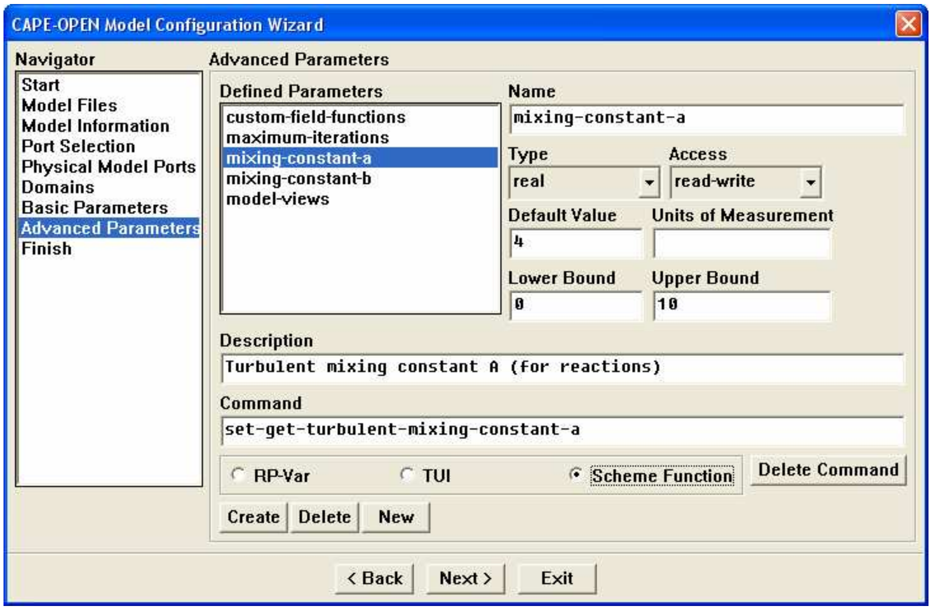


Figure 6.12: Advanced Parameters panel in FLUENT Configuration Wizard

Upon clicking **Finish** in the Finish panel (Figure 6.13), the Configuration Wizard prompts the CFD user to save an updated case file. At the same, the configuration information is saved in a model XML file (Master.xml) and a solver XML file (e.g., Fluent2D.xml). These two XML files are stored in the EM Database and are used by the Integration Controller to populate the forms in the Model Selection GUI used when a CAPE-OPEN CFD block is placed on the Aspen Plus flowsheet.

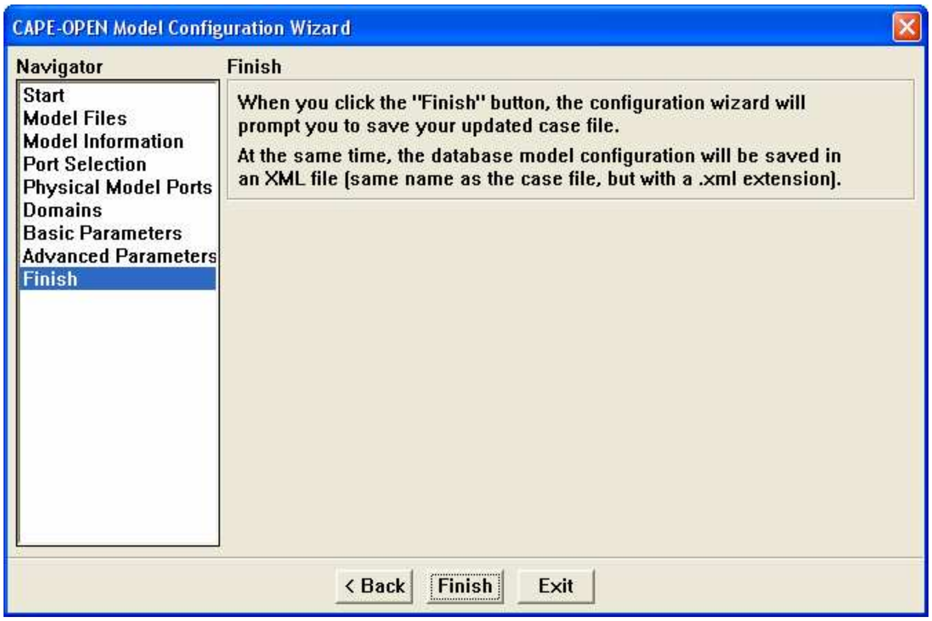


Figure 6.13: Finish panel in FLUENT Configuration Wizard

6.1.4 Model integration

The integrated simulation involves an iterative, sequential-modular solution process. To generate a good starting point, we first perform a steady-state simulation of the reaction-separation-recycle flowsheet using the Aspen Plus CSTR. As mentioned above, the Aspen Plus CSTR uses power law kinetics to calculate reaction rates and does not consider mixing effects. To account for the impact that mixing in the CSTR has on overall process performance, we now introduce the FLUENT reactor model into the flowsheet.

6.1.4.1 CFD model selection and edit

In Aspen Plus we load the CAPE-OPEN (CO) model library by selecting **References** from the Library menu and then selecting **CAPE-OPEN**. This automatically adds the CAPE-OPEN model category on the Aspen Plus model palette. From the CAPE-OPEN tab in model palette, we select the **CO-CFD Block** and drag and drop the icon on to the process flowsheet window (Figure 6.14).

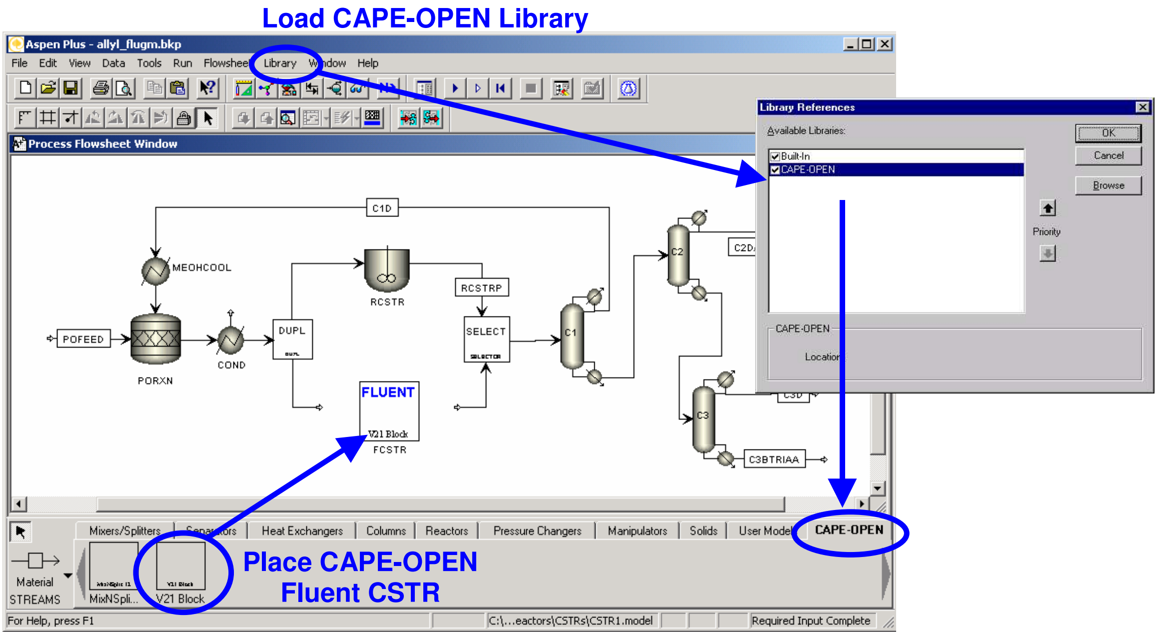


Figure 6.14: Load and place CAPE-OPEN CFD block on process flowsheet

Double-clicking on the **CO-CFD Block** in the process flowsheet window automatically brings up the Model Selection GUI, which is a window into the EM Database. As shown in Figure 6.15, the Model Selection GUI consists of three main fields: Model Database Selection, Model Browser, and Model Information.

After selecting the model database, the Aspen Plus user can browse the model tree for the equipment model of interest. Upon highlighting a model in the browser, the tabbed forms are populated with “read-only” model information from the XML files generated by the FLUENT Configuration Wizard (Figure 6.13). The model information, including ports and parameters, helps process design engineers find the appropriate model and is more effective than making a selection simply based on file names.

For the CSTR example, we look under the Reactors category and CSTRs model type. Note here that the model category and type in the model browser match those specified in the Model Information panel of the FLUENT Configuration Wizard (Figure 6.6). Click the **OK** button to select the CSTR model.

Upon selecting a model, the Model Edit GUI (Figure 6.16) automatically pops up. The first tabbed form on this GUI is used to define a Solution Strategy. For this example, the Fluent2D solver is used for the entire coupled simulation. The other two tabbed forms in the Model Edit GUI are for reviewing and modifying Basic Parameters and Advanced Parameters. The initial parameter values correspond to those set by the CFD design engineer in the FLUENT Configuration Wizard.

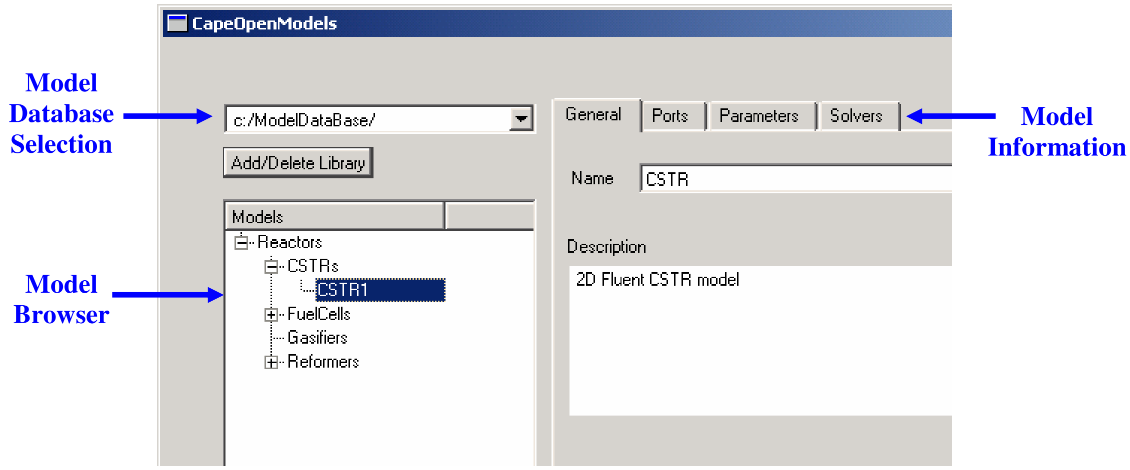


Figure 6.15: Model Selection GUI

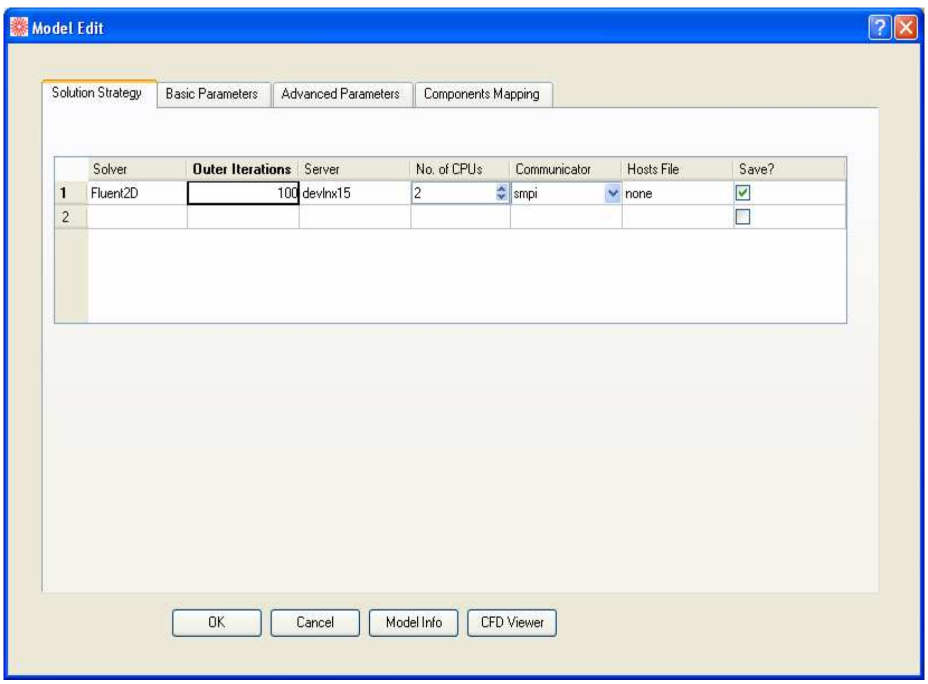


Figure 6.16: Model Edit GUI

Upon clicking **OK** at the bottom of the Model Edit GUI, the FLUENT CSTR model is now associated with the corresponding block in Aspen Plus. The Model Edit GUI can be accessed again by double clicking on the block in Aspen Plus or right-clicking and selecting **Edit Model** from the list.

The next step in the integration procedure is to connect material streams to the FLUENT block. To assist with connecting streams to a CO-CFD block, position the mouse over a displayed inlet or outlet port and a text box with the FLUENT port name(s) appears. For the CSTR example, you can see *mass-flow-inlet-6* for the inlet port and *pressure-outlet-7* for the outlet port. After connecting the streams to the block, the FLUENT model is now fully integrated into the Aspen Plus flowsheet. Before running the CSTR example, we must first deactivate the Aspen Plus CSTR and select the FLUENT stream as the inlet stream specification for the selector block.

6.1.4.2 Running the integrated model

The process engineer interactively runs the integrated simulation by clicking **Run** on the Aspen Plus Simulation Run toolbar. The progress of the combined simulation can be monitored from within Aspen Plus using the Control Panel. Aspen Plus controls the integrated simulation and automatically executes the FLUENT model at each flowsheet iteration. The CFD results are saved at each Aspen Plus iteration so that subsequent FLUENT simulations converge more quickly.

The integration approach employed here is for Aspen Plus and FLUENT to model different unit operations and exchange stream information at flow boundaries (i.e., mass flow rate, composition, temperature, and pressure). The stream information, along with physical properties and reaction kinetic data, are transferred automatically from Aspen Plus to FLUENT using CAPE-OPEN interfaces.

For a given impeller speed in the CSTR example, FLUENT computes the flow pattern and chemical species distribution. The base case impeller speed used here is 100 rpm. The mass-weighted averages of the flow field variables at the CSTR outlet zone are sent back to the corresponding Aspen Plus stream using the CAPE-OPEN interfaces.

Upon completion of the integrated simulation, the process engineer reviews results for streams, blocks, and overall convergence in Aspen Plus. The stream information at the outlet of the FLUENT block is obtained as usual from the Aspen Plus Results menu.

6.1.4.3 CFD Viewer

To view the CFD results, the process engineer uses the CFD Viewer from the Model Edit GUI. This tool enables the display of basic CFD results such as velocity vectors and contours of temperature, pressure, velocity, and species mass fractions in the equipment model from within the Aspen Plus simulation environment (Figure 6.17). For more advanced display options, the FLUENT GUI can also be used.

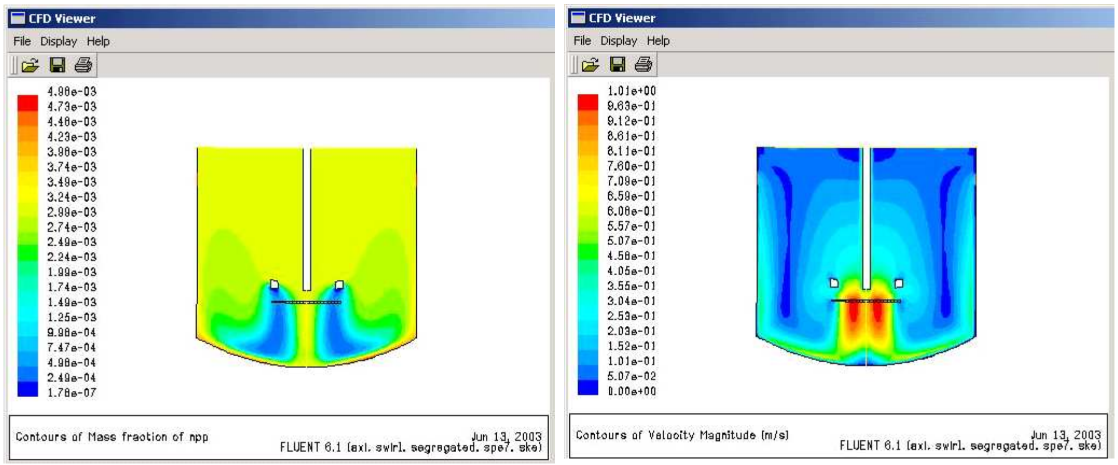


Figure 6.17: CFD Viewer showing contours of mass fraction of an impurity (left) and velocity magnitude (right) in CSTR

6.1.4.4 Aspen Plus analysis tools

When using the FLUENT finite-rate reaction model (i.e., well mixed), the results for product mass fraction and flow rate are independent of impeller speed and match the Aspen Plus CSTR results. After the base-case simulation, the process engineer can use Aspen Plus analysis tools (e.g., Case Study, Sensitivity, Optimization,) to analyze and optimize performance of the FLUENT equipment model with respect to the entire process flowsheet.

6.1.4.4.1 Sensitivity analysis

The Sensitivity analysis tool in Aspen Plus can be used to determine how a process reacts to varying key operating and design variables. The process engineer can use it to vary one or more flowsheet variables and study the effect of that variation on other flowsheet variables. It is a valuable tool for performing "what if" studies.

In the Sensitivity setup, CAPE-OPEN parameters selected in a FLUENT Configuration Wizard can be selected as Manipulated Variables on the Vary sheet. Figure 6.18 shows how the FLUENT CSTR shaft speed parameter, namely udf/shaft_speed, is selected as a variable to be manipulated in a sensitivity analysis.

Figures 6.19 and 6.20 show the results of an Aspen Plus sensitivity analysis to determine how product yield and purity react to varying the shaft speed (i.e., mixing) in the FLUENT CSTR using an hybrid finite-rate and eddy-dissipation model (i.e., partially mixed). The results in Figure 6.19 show that product yield is maximized at a shaft of about 175-200 rpm. In Figure 6.20, product purity approaches the finite-rate results of nearly pure product at shaft speeds of about 300 rpm and greater. Product purity decreases as shaft speed is decreased and goes through a minimum at approximately 125 rpm when the reaction rate for an impurity-generating reaction switches from eddy dissipation to finite rate.

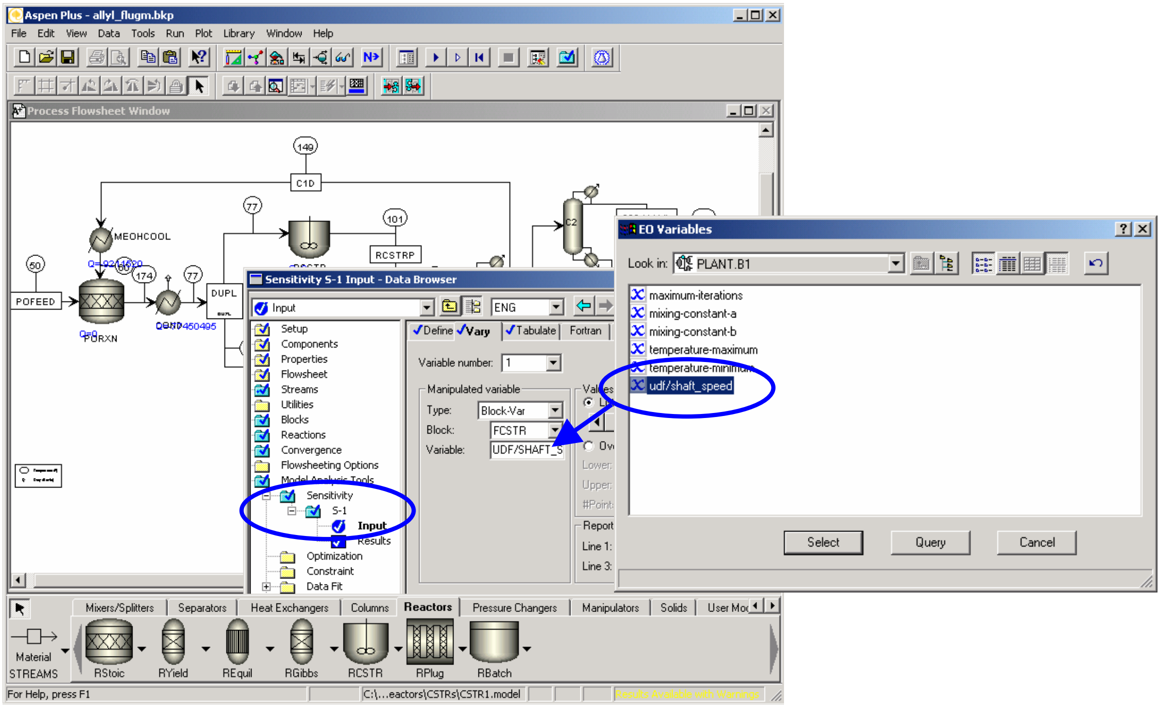


Figure 6.18: Selection of CAPE-OPEN CFD parameter (shaft speed) as a manipulated variable in an Aspen Plus sensitivity analysis

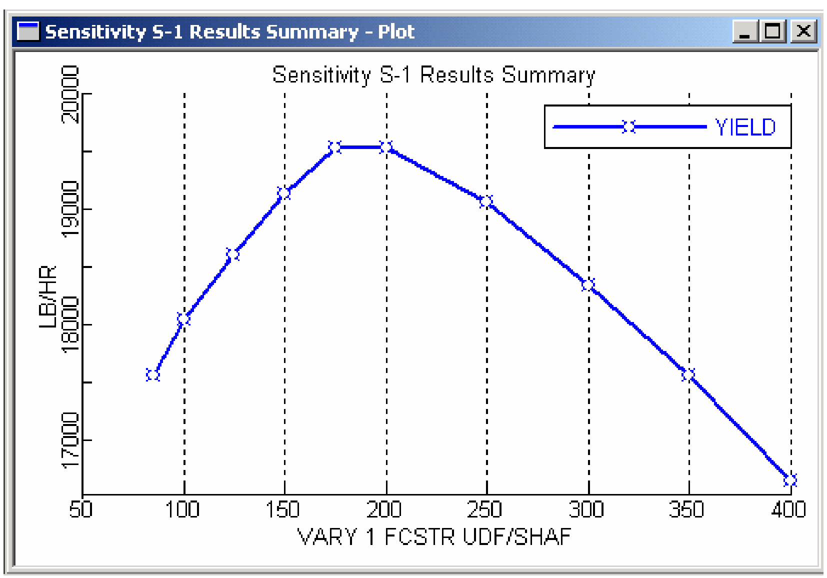


Figure 6.19: Sensitivity analysis for product yield vs. shaft speed

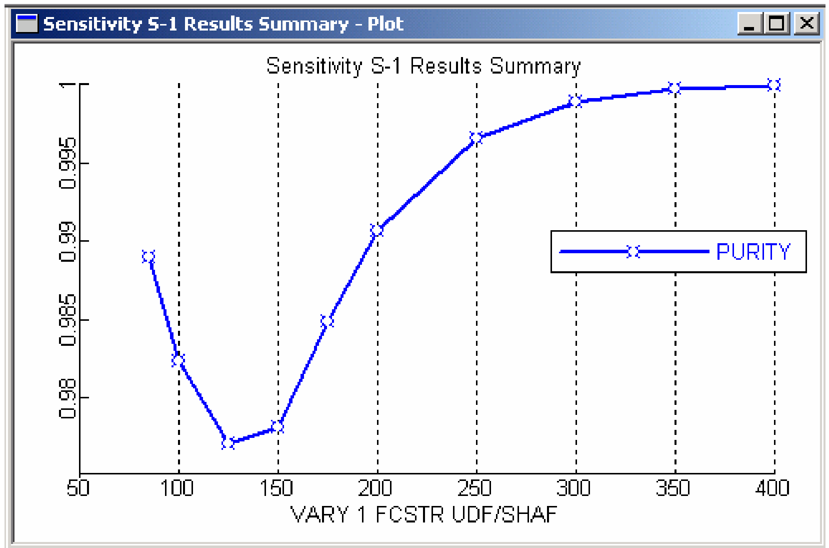


Figure 6.20: Sensitivity analysis for product purity vs. shaft speed

6.1.4.4.2 Optimization

The Optimization tool in Aspen Plus can be used to maximize or minimize an objective function by manipulating decision variables (feed stream, block input, or other input variables). In the Optimization setup, CAPE-OPEN parameters selected in a FLUENT Configuration Wizard can be selected as Manipulated Variables on the Vary sheet.
For the FLUENT CSTR, the shaft speed parameter, namely udf/shaft_speed, is selected (Figure 6.21) as a variable to be manipulated in maximizing the product yield – the mass flow of product in the bottom stream of the final distillation column (Figure 6.22).

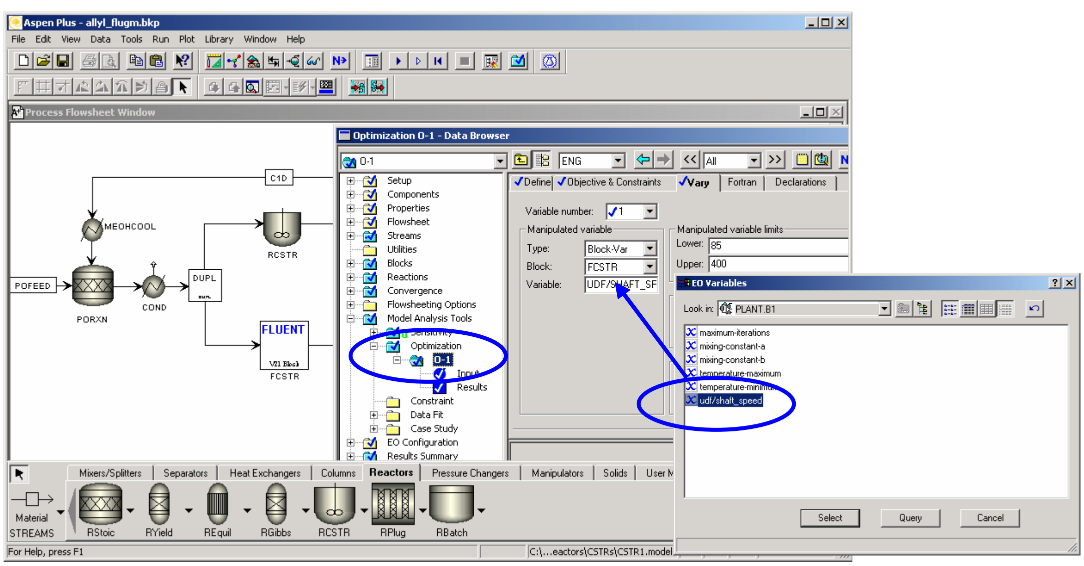


Figure 6.21: Selection of CAPE-OPEN CFD parameter (shaft speed) as a manipulated variable in an Aspen Plus optimization

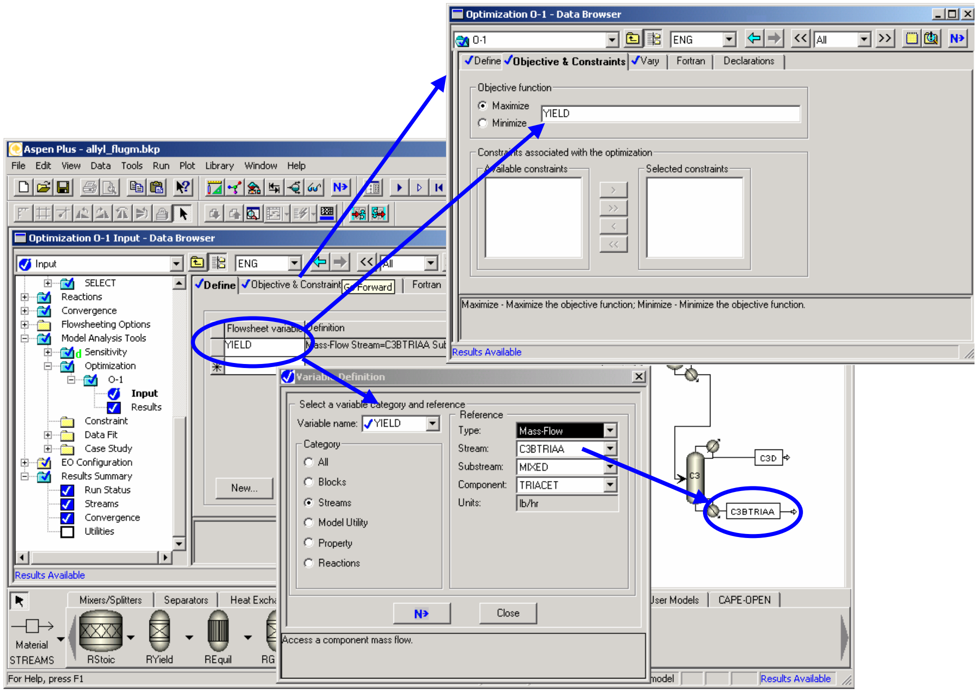


Figure 6.22: Definition of product yield as the variable to be maximized in an Aspen Plus optimization

Using lower and upper limits of 85 and 400 rpm, respectively, for shaft speed, the Aspen Plus optimization finds a maximum product yield of 19567 lb/hr at a shaft speed of about 190 rpm. This result is consistent with the results of the sensitivity analysis shown in Figure 6.19.

Clearly, such integrated simulations can be used to analyze overall process performance with respect to important design and operational parameters simulated using CFD. The optimization of CFD parameters is not done in isolation but within the context of the whole process, so that a global improvement is achieved, especially for processes that depend on mixing and fluid dynamics. Note here that it may also be possible to improve product quality and yield by modifying equipment geometry or feed location.

6.2 Example 2: Fuel cell flowsheet with reformer CFD model

The second example of integration is that of a solid oxide fuel cell power system coupled with a FLUENT CFD model of the reformer.

6.2.1 System-level model development

The fuel cell system flowsheet used for this example, taken from (Virji et al.1998), is shown in Figure 6.23. The fuel (natural gas) is mixed with steam, preheated, and fed into a reformer. The reformer consists of a tube packed with catalyst pellets and heated on the shell-side with hot gases from the energy recovery unit, which burns the unconverted hydrogen and supplemental fuel. The reformer converts natural gas into a mixture of carbon monoxide (CO) and hydrogen (H_2). The mixture is then passed through the shift converter to convert the CO into H_2. In this flowsheet we did not include a PROX (preferential oxidation) reactor, which is used to remove the traces of CO remaining in the gas stream. The hydrogen-rich fuel is cooled and fed into the anode side of the fuel cell. Air is compressed and fed into the cathode side of the fuel cell. The exhaust gas from the fuel cell is burned with supplemental fuel in the energy recovery unit, and the hot gas is used to supply heat for the endothermic reactions in the reformer and to preheat the fuel. Work extracted from the hot gas is used to power the air compressor.

The entire fuel cell process was modeled in Aspen Plus. Initially a plug flow reactor with co-current coolant model (Rplug model in Aspen Plus) was used to represent the reformer. A rigorous reaction equilibrium model based on Gibbs free energy minimization (Rgibbs) was used to model the shift reactor. The fuel cell and exhaust gas burner were represented with a stoichiometric reactor with a specified fractional conversion (Rstoic).

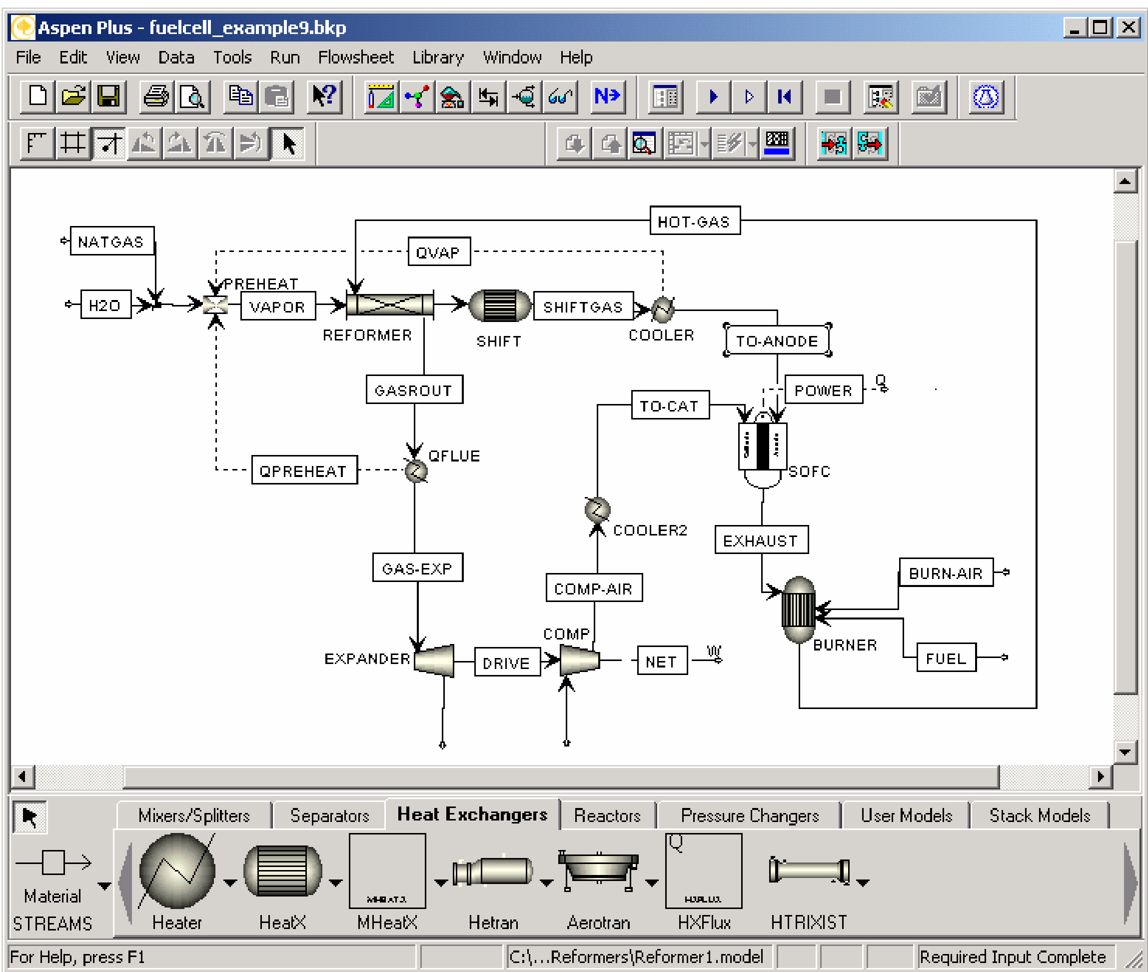


Figure 6.23: Aspen Plus flowsheet for fuel cell system example

6.2.2 Equipment-level model development

In this study we replaced the flowsheet reformer model (Figure 6.23) with a 3-D CFD model to demonstrate the integration of a detailed component model with a model of the entire fuel cell process. The use of a FLUENT model gives the following advantages over the Aspen Plus RPlug model:

- The CFD model calculates important parameters that are required inputs to the RPlug model: the shell-to-tube heat transfer coefficient and the pressure drops through the shell side and the tube side of the reactor. The CFD model requires inputs that are not needed in the RPlug model, such as the details of the reactor geometry, the catalyst particle size, and the void fraction in the catalyst bed. However, these quantities are easily measurable and are independent of the operating conditions such as the flow rate, pressure, and temperature. The heat transfer coefficient and the pressure drop, on the other hand, vary significantly with operating conditions.

- The CFD model calculates the 3-D distribution of the flow field, temperature, pressure, and species concentrations in the reactor. Thus the CFD model is able to account for effects that cannot be readily accounted for in a 1-D model such as RPlug. For example, the CFD model accounts for the radial variation in the temperature in the catalyst bed (in the tube). The temperature is higher near the tube wall because of the heat transfer from the hot gas on the shell side than at the center of the bed. Therefore, the reaction rate is higher near the tube wall than near the center. The average rate of reaction and hence the conversion is affected by the radial variation in temperature. In a 1-D model such an effect can be included as an effectiveness factor, which, however, is not easy to calculate. In some cases, such as in pollutant formation reactions, predictions based on 1-D temperature distributions could be grossly inaccurate. The predicted flow rate, composition, and temperature at the outlet subsequently affect all other calculations in the process model.

- The detailed calculations provide the process analyst information that, although not required for the process simulation, is important for the overall system design. In the case of the reformer model, the detailed temperature distribution in the catalyst bed would be useful to ensure that the temperature anywhere in the catalyst bed does not exceed the sintering temperature. In general, local peak temperatures in comparison with allowable material limits are a common issue in advanced power plant design.

When the reformer model is executed from within the fuel cell flowsheet, the CFD model benefits from the ability to account for the effect of recycle streams. The fuel gas is heated with the products of combustion from the energy recovery unit. The conversion in the reformer is limited by the energy available from the hot gas, which in turn depends upon the conversion in the reformer. Furthermore, the feed stream to the reformer is preheated with the outlet stream from the shift converter and the shell outlet gas.

The reformer is modeled with an idealized single-step reaction:

$$CH_4 + H_2O \rightarrow CO + 3\,H_2$$

An Arrhenius rate expression is used to calculate the reaction rate (kmol/m^3·s):

$$r_{CH4} = -k_0\, e^{-E/RT}\, p_{CH4}^{\;0.9845}\, p_{H2O}^{\;0.05}$$

where the partial pressure of methane and steam are stated in kPa, $k_0 = 5 \times 10^{-7}$, and E = 2.09x10^7 (J/kmol). The rate expression, entered using the Aspen Plus Reactions menu, is transferred from Aspen Plus to the CFD model at run time. (A routine was included in FLUENT to calculate the rate based on information from Aspen Plus because the default concentration basis in FLUENT is molarity rather than partial pressure).

The reformer is 5 m long and 0.4 m in outer diameter. The volume was discretized with 33,432 cells. The inner catalyst tube has a diameter of 0.1 m. The operating pressure is about 390 kPa on the shell side and 290 kPa on the tube side.

6.2.3 Equipment-level model configuration

The basics steps for using the FLUENT Configuration Wizard to convert the reformer model into a CO-compliant model are similar to those described for the CSTR example in Section 6.1.3. The main difference is that the reformer model has two domains, the hot gas domain (shell side) and the process domain (tube side). Figures 6.24-6.26 show the port and domain selection process for this case.

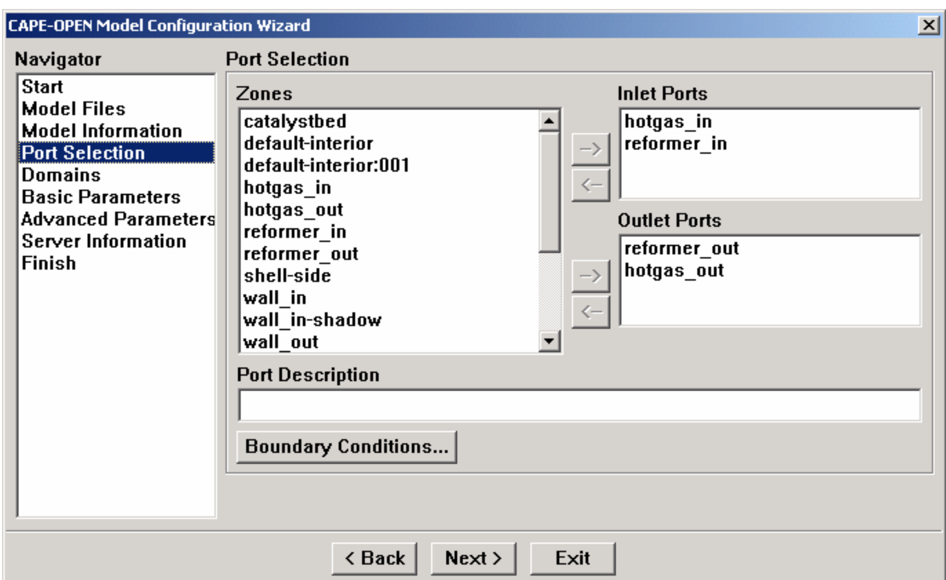


Figure 6.24: Port Selection for reformer with two inlets and two outlets

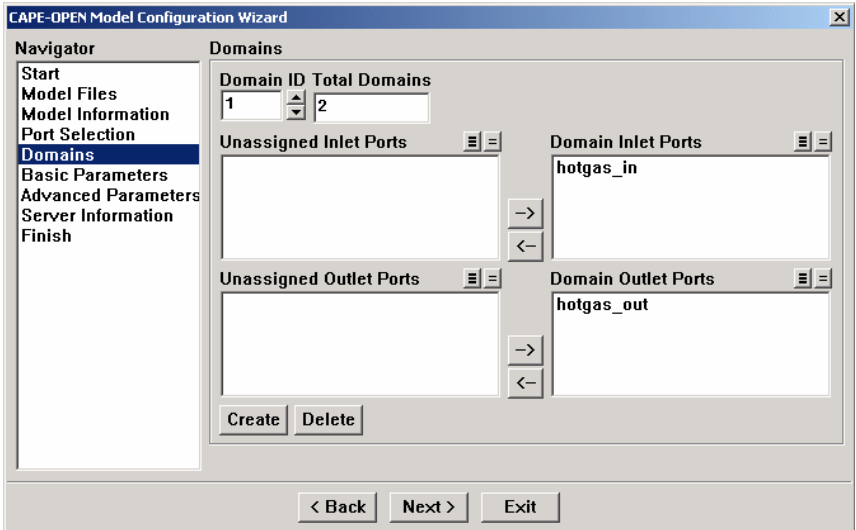


Figure 6.25: Domain 1 representing the hot gas (shell side) of the reformer

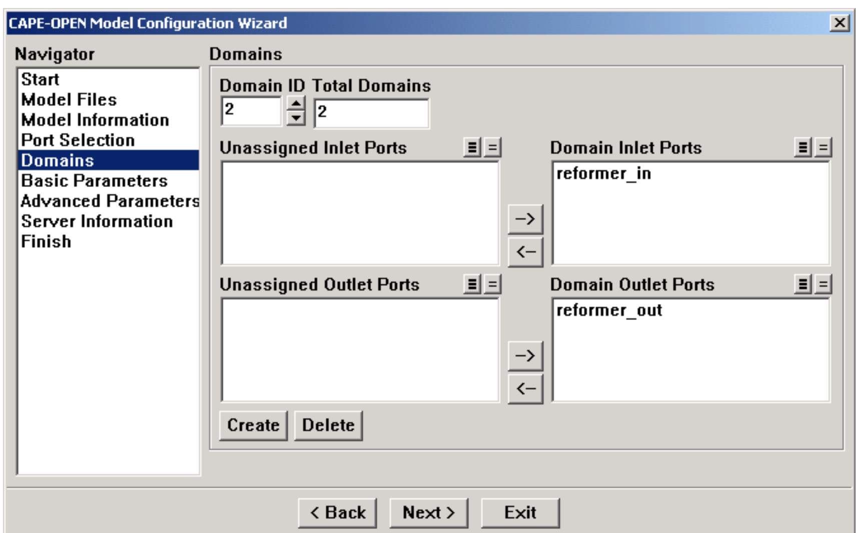


Figure 6.26: Domain 2 representing the process (tube side) of the reformer

6.2.4 Model integration

6.2.4.1 Solution strategy

The solution strategy for a given equipment model is specified using the grid on the Solution Strategy form of the Model Edit GUI (Figure 6.27). It consists of the solver(s) to be used and the solver switching criteria based on the number of outer iterations, an integer number corresponding to the maximum number of times the block is called by Aspen Plus. The default solver is set to be the most rigorous solver (e.g., FLUENT2D, FLUENT3D) associated with the equipment model under consideration. The user can select another solver from the combo box in the Solver column of the grid. The default number of outer iterations (100) can also be modified to another integer value. The user can also specify on which computer system each solver is to be executed. This is done by typing in the server name or IP address in the appropriate column.

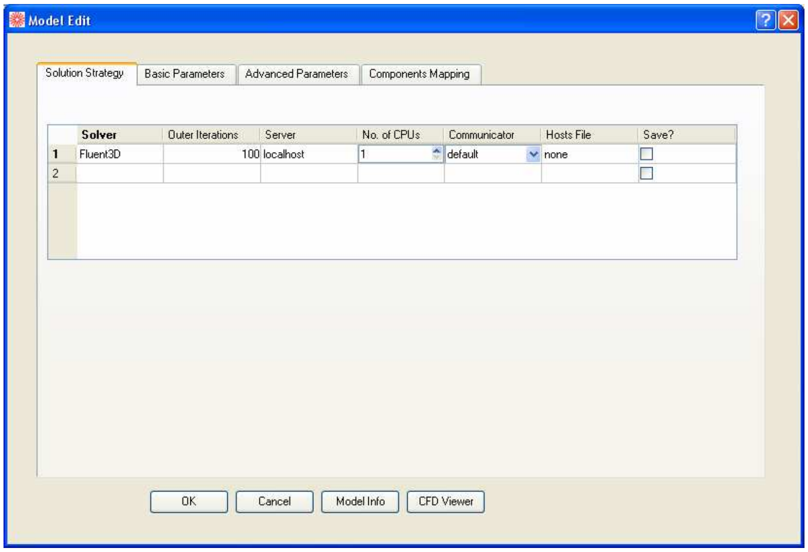


Figure 6.27: Default Solution Strategy form on the Model Edit GUI

The Integration Controller v1.0 also enables the process engineer to automatically generate a ROM based on a linear regression of pre-computed CFD results. As shown in Figure 6.28, this can be done by selecting the save option on the Solution Strategy form on the Model Edit GUI when running a coupled simulation using the FLUENT solver. To use the ROM in subsequent integrated simulation, simply select the **Regression** solver in the Solver column of the Solution Strategy grid (Figure 6.28).

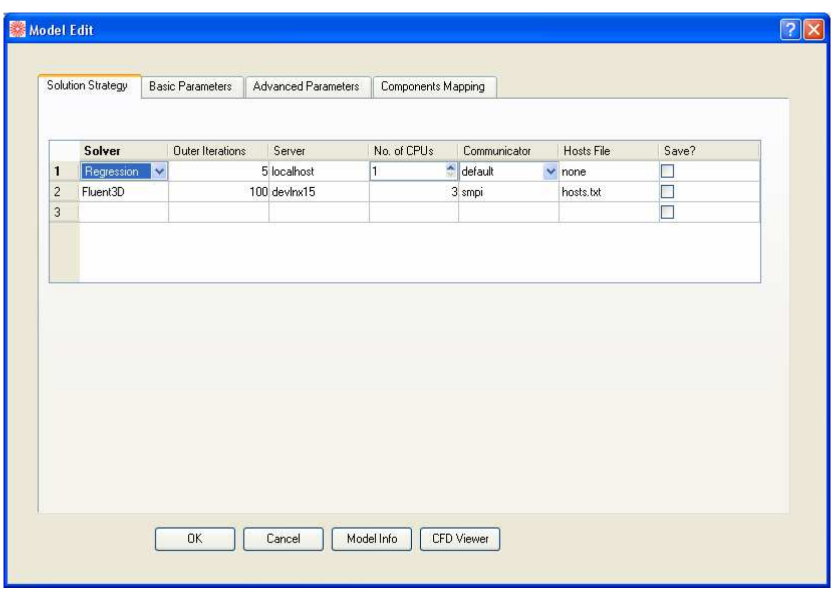
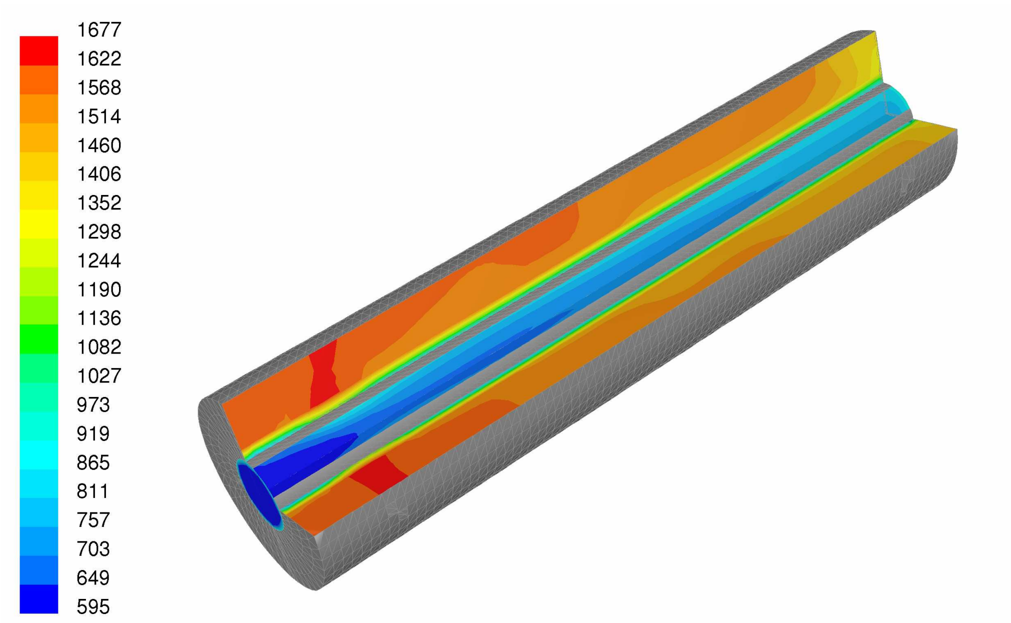


Figure 6.28: Hybrid solution strategy using a ROM and FLUENT

6.2.4.2 Running the integrated simulation

The temperature profile in the reformer calculated with the 3-D model integrated with the process model is shown in Figure 6.29. The plot shows the temperature distribution on a pie-slice through the reformer. The gray color indicates the outer tube surface and part of the inner tube at the center. The reactor is not drawn to scale; the aspect ratio in the figure is much less than the real aspect ratio of 12.5:1 so that the temperature variation in the radial direction can be better visualized. The mixture of natural gas and steam enters the inner tube from the left at a temperature of about 600 K (dark blue), travels through the catalyst bed, and exits from the right at a temperature of around 1100 K (light blue). The fuel gas is heated with the hot gas flowing on the shell side. The hot gas enters the shell near the left end at 1677 K (as indicated by the red bands), flows co-currently with the fuel gas, and exits near the right end. It cools down to about 1300 K by supplying the heat of endothermic, reforming reaction taking place in the inner tube. The radial variation in the temperature on the tube side is especially pronounced near the entrance. Near the reformer exit, the temperature distribution on the tube side is more uniform, since the endothermic reactions are nearly complete. The radial variation on the shell side is even more pronounced because the hot gas enters and exits from the side, and unlike the inner tube there is no catalyst bed to moderate the temperature variation.



Contours of Static Temperature (k) May 14, 2002
FLUENT 6.1 (3d, segregated, spe7, ske)

Figure 6.29: Temperature distribution in the reformer calculated with CFD

Results of the integrated simulation are provided in the following tables. Table 6.4 compares the conditions at the reformer outlet from a simulation using the plug flow reactor (PFR) model and the CFD reformer model. The CFD model predicts a lower conversion because of the effect of the radial temperature variation on the reaction rate. The heat transfer coefficient predicted by CFD is smaller than that assumed in the PFR model. Because of the lower conversion in the reformer, the power output predicted by the integrated simulation (72 kW) is significantly lower than that predicted by the simulation using the default PFR model (90 kW).

Table 6.4: Reformer outlet conditions

	CFD	PFR
Flow rate, kg/h	22.4	22.4
Temperature, K	1166	1170
Composition, Mole fraction		
O_2		
N_2	0.019	0.002
CO	0.195	0.220
CO_2	0.002	
CH_4	0.036	0.002
H_2	0.586	0.661
H_2O	0.161	0.115

Table 6.5 summarizes the outlet conditions on the shell side. Because of the lower predicted heat transfer coefficient in the CFD model, the shell side temperature does not drop as much as in the PFR model.

Table 6.5: Shell outlet conditions

	CFD	PFR
Flow rate, kg/h	213.7	213.7
Temperature, K	1500	1171
Composition, Mole fraction		
O_2	0.016	0.007
N_2	0.655	0.648
CO		
CO_2	0.079	0.082
CH_4		
H_2		
H_2O	0.250	0.262

Even the reformer inlet conditions in the two simulations are different because of the recycle stream. The reformer inlet gas temperature for the CFD model (631 K) is lower than that for the PFR model (635 K). The shell inlet gas temperature for the CFD model (1635 K) is also lower than that for the PFR model (1649 K). For the shell inlet gas even the compositions are different (Table 6.5) because the different conversion in the tube side ultimately affects the hot gas recycled to the shell side.

The results reported above are from a preliminary simulation and are intended to demonstrate the feasibility of integrated simulations and the ability to account for the effect of recycle streams. We cannot, however, judge the relative accuracy of the results without a comparison with experimental data.

6.3 Vision of organizational usage

Our vision is that the integration software will foster collaboration between CFD engineers and process engineers. (In the following discussion, somewhat equivalently, the term CFD model may be replaced with custom model.) The process engineers will benefit by having access to improved equipment-level models based on CFD or custom models or ROM. CFD engineers will benefit from being able to use physical properties consistent with process models and to account for recycle loops in CFD-based equipment optimization.

Although the robustness and user-friendliness of CFD software are rapidly improving, CFD analysis still typically requires an expert. Therefore, collaboration between CFD engineers and process engineers during the initial phases of integrated model development is essential for success. Three levels of collaboration between CFD and process engineers are envisioned, as follows:

<u>One on one</u>. In this case the CFD engineer and the process engineer are the same person or are two closely collaborating colleagues. The CFD engineer discusses the equipment model requirements with the process engineer and develops and tests the model. The model is then configured and added to a local EM database. The CFD engineer may also create a ROM if needed. The CFD engineer then gives the process engineer access to his EM database. This scenario assumes that both the engineers are working on computers located on the corporate LAN. (This methodology will also work on corporate VPNs over the Internet. For example, we have demonstrated that an Aspen Plus simulation conducted on a computer at Fluent headquarters in Lebanon, NH can launch a FLUENT model on computer in the Fluent office in Morgantown, WV. These two computers are on a VPN.) The process engineer will include the model in the process simulation and conduct integrated simulations. The process engineer can choose to run the CFD model on the machine on which the model was tested, so that there are no problems caused by a change in the computer hardware. This collaboration can be effected with the current version of the software. The only requirement is that the system administrator sets up the appropriate file access privileges for EM database.

<u>Corporate database</u>. In large organizations a case can be made for a corporate model database in which mature and well-documented CFD models are archived. The process engineers will then browse the corporate EM Database and select appropriate CFD models to conduct integrated simulations. We hope that this vision becomes a reality in the future. However, several issues need to be resolved:

a) Reusing CFD models is not easy because CFD models are intimately linked to an underlying geometry. Therefore, most of the archived CFD models must accept certain geometric information as inputs; e.g., the model accepts boiler width as an input and automatically scales and regrids a baseline geometry. The current version of the software does not provide this flexibility. There may be exceptions where such flexibility is not needed. For example, certain plant equipment may have a finalized geometry.
b) Sometimes when model parameters are changed, the CFD model may not converge. Typically process engineers are not trained to solve such convergence issues. A need therefore exists to have CFD software with a more autonomous "solution steering" capability. Research in this area is ongoing; such software is expected to be available in the near future (e.g., Fluent's FloWizard product).
c) The database administrator must ensure that the models are well parameterized and well documented in order to facilitate model reuse is. This database administrator needs to have a working knowledge of integrated model development.
d) The capability of the current file-based EM Database is not adequate when a large number of CFD models needs to be maintained. In a future phase of the project a better database may be implemented.

<u>Inter-company collaboration</u>. The purpose of the Vision 21 program is to design advanced power plants by combining technology modules developed by different companies. To create a virtual-plant simulation by assembling models from different companies, at present, it will be necessary for the companies to transfer the integrated model, which includes relevant sections of the Model Database, to the entity that is designing the advanced power plant. One difficulty of this approach is that companies may be reluctant to share proprietary models. However, by linking geographically distributed models over the Internet, it will be possible for companies to share models without compromising their confidential information and hence lead to novel integration schemes and Vision 21 plants. This capability is available in a limited form with the existing software. Companies may make their CFD models accessible for integrated simulation by providing a "hole" in their firewall through which a specific computer may access the model. We have demonstrated this approach by including a CFD model located on a Pittsburgh Supercomputing Center computer in an integrated simulation running on a laptop in Morgantown, WV, over the Internet. However, this mode of communication is not secure; considerable additional development is required to make model access over the Internet secure and acceptable to commercial companies. In a recently initiated project sponsored by Department of Trade and Industry, UK, in which Fluent is a participant, such a capability is being developed.

7 Application of Integrated Modeling to Power Plant Simulations

To assist the project team in developing and demonstrating the capabilities of the advanced integration software, ALSTOM Power was tasked with selecting and running two industrial demonstration cases, modeled on the basis of existing power plants. Both of these demonstration cases included the coupling of Aspen Plus with FLUENT and ALSTOM legacy design codes and were run over a load range on a PC. Subsequently, a third demonstration case was also run, based on an advanced FutureGen concept, in which Aspen Plus was coupled with FLUENT alone. The coupled running of Aspen Plus and FLUENT on both a PC and over a LAN were demonstrated.

The philosophy of progressing in a step-by-step manner, from the relatively simple to the more complex, was adopted in this aspect of the project. The demonstration cases are defined as:

- *Demonstration Case 1* – a conventional steam cycle, containing a wall-fired coal boiler and post-combustion cleanup equipment, fuel handling equipment, steam turbine and generator, heat exchange equipment, and pumps. A 30 MWe coal-fired power plant for municipal electricity generation was selected for the cycle study.

- *Demonstration Case 2* – a natural gas combined cycle (NGCC), consisting of a gas turbine, steam turbine, heat recovery steam generator (HRSG), etc. A 270 MWe, natural gas-fired, combined cycle power plant was selected.

- *Demonstration Case 3* – a 250 MWe FutureGen IGCC, with an air separation unit (ASU), CO_2 capture, a gas turbine burning hydrogen enriched syngas, a pressure-swing absorption (PSA) section for hydrogen stream production, a heat recovery steam generator (HRSG), and acid-gas cleanup.

Although the first two demonstration cases do not constitute Vision 21 concepts, a number of the cycle components in the cases are present in a Vision 21 plant. Three separate runs were completed for these two cases to demonstrate the software interfaces (Sloan, et al., 2002; Sloan and Fiveland, 2003; Sloan, et al., 2004).

- Run 1: An initial (baseline) run, which utilized exclusively the existing component libraries in Aspen Plus to determine the overall cycle performance and characteristics.
- Run 2: A second run, in which (one or more) cycle components were replaced with an ALSTOM design code.
- Run 3: A third run, in which cycle components were "replaced" with a FLUENT CFD code simulation.

Each of the three stipulated runs was performed over a range of loads (e.g., to simulate power demand changes from 100% to 50% in a pseudo-steady state fashion). Insofar as possible, for both Demonstration Cases 1 and 2, plant data was used to first calibrate the ALSTOM Power proprietary design codes, and then the computations of the design codes were used to calibrate and align portions of the cycle flowsheets. The separate runs for each of the cases are tabulated in Table 7.1:

Table 7.1: Summary of Demonstration Case Runs

Demonstration Case	Run 1 (Baseline)	Run 2 (ALSTOM Design Codes)	Run 3 (CFD)	Component(s) Replaced
1	Library Modules	BPS	FLUENT	Boiler "island"
2	Library Modules	HRSGPS	FLUENT	HRSG "island"

Since the power plant for Demonstration Case 3 is in the conceptual stages of development and does not yet exist, a FutureGen cycle Aspen Plus flowsheet was provided by NETL. An HRSG was designed using ALSTOM's HRSGPS design code to match the constraints and boundary conditions of the HRSG tube banks represented on the flowsheet. A corresponding FLUENT model of the HRSG was constructed and calibrated, again using the results of the design code. The calibrated FLUENT model was then instantiated on the flowsheet and run, in coupled fashion, with the Aspen Plus cycle at the design load condition.

The FLUENT code was used to treat, in a coupled fashion, both the gas-side flow, as well as the steam-side flow associated with the heat-exchanger tubebanks within a boiler or HRSG module. The pseudo 1-D tube bank model in FLUENT was used to represent the steam-side processes within the computational domain.

The first two demonstration case runs were largely computed on a single-processor (500 MHz) PC. Because of the CPU-intensive nature of these initial CFD runs, the CFD models were understandably reduced in size and simplified considerably (relative to their industrial design counterparts) in order to make the computations more practical on a PC. The size of the FLUENT mesh for Demonstration Case 3 was also kept relatively small (less than 40,000 cells) in order to promote quick turnaround times for debugging and testing purposes. After the IC and wrapper technologies were extended to LINUX platforms, both Demonstration Case 3 and Demonstration Case 2 (Run 3) were run over a LAN, with the CFD case being computed on a LINUX platform and the Aspen Plus cycle running in the PC Windows environment.

7.1 Demonstration Case 1: Conventional Steam Cycle

A municipal power station, which provides electricity to a city in the United States, was selected for the Case 1 conventional steam cycle. The steam generator is a (non-ALSTOM) 1950s vintage front wall-fired, balanced draft, natural circulation steam generator with a nominal superheated steam flow of approximately 41 kg/s (325,000 lb/hr) at 755 K (900°F) and 6.2 MPa (900 psig). Nominally rated at 33 MWe, the unit has six ALSTOM Power burners, arranged in two elevations of three burners each, which are supplied pulverized coal. Each burner has three air register zones to supply three different air annuli at the burner exit. The combustion flow field is controlled by means of individualized flow splits and swirl numbers for each air annulus.

Hot combustion gases from the firing zone region flow upward through a superheater platen section, after which the flue gas flows vertically downward through a low temperature superheater (LTSH) section and a bypass cavity, which are configured to be in parallel with each other. The extent to which the flue gas flows preferentially through either the bypass cavity or the LTSH section is determined by the backpressure supplied by the bypass damper. The flue gas then flows vertically upward through a “boiler bank” and then vertically downward through a final exhaust cavity or duct. The flue gas passes through an air preheater, heating the air to the secondary air temperature required by the burners.

Only two types of heat exchanger sections are utilized in the Case 1 power plant – evaporative sections and superheat sections. Saturated vapor leaves the drum through the overhead steam line and travels through the LTSH section, followed by a higher temperature superheat platen section, where the steam is superheated to the temperature required by the turbine. The relative amount of the total absorption that is allocated to superheating versus the relative amount of the total absorption that is allocated to evaporative heating is controlled by the operator. The amount of superheat absorption is controlled primarily by factors such as bypass damper position and excess air. For example, if the bypass damper is closed, then the bypass cavity is back-pressured, causing most of the flue gas to pass through the low temperature superheater, thereby preferentially giving most of the remaining energy to the superheat section, at the expense of the boiler bank.

7.1.1 Case 1: Run 2 – Aspen Plus Coupled with BPS

Run 1 (not shown) consists of the baseline cycle using Aspen Plus library modules alone. Run 2 consists of the coupling of an ALSTOM Power proprietary design code with Aspen Plus. The ALSTOM design package selected for use was an industrial-boiler performance simulation (BPS) code. The BPS package is a legacy design code that was built upon proprietary empiricisms, refined over time through experiments and accrued experience, and effectively recalibrated with each completed contract. As is typical of many such codes in industry, it was written in FORTRAN with fixed format inputs and batch output files. The virtual-plant simulator should be expected to have the ability to accept the valuable information contained in such legacy codes. A rendition of the Demonstration Case 1 industrial steam cycle, showing the baseline cycle with the Aspen Plus library modules, as well as the proprietary code instantiation, is shown in Figure 7.1.

When the BPS code is instantiated as a CAPE-OPEN block upon the process flowsheet from the Aspen Plus CAPE-OPEN model palette, it essentially replaces the entire gas-side of the cycle. The BPS code contains all of the required information about the gas-side components, including the air preheater, the pulverizers, etc. In the present case, the BPS code constitutes a single block icon on the gas-side of the flowsheet that must interact with and exchange information with the steam-side of the cycle. The BPS package has not been coded with material stream or port connections. Consequently, all of the information exchange between Aspen Plus and the BPS code must occur through a transfer of shared variable or parameter values. The CAPE-OPEN interfaces between the BPS code and Aspen Plus assist in the transfer of this information. A schematic showing the Integration Controller (V21 Controller) interface and COM-CORBA Bridge configuration for the legacy BPS code is shown in Figure 7.2.

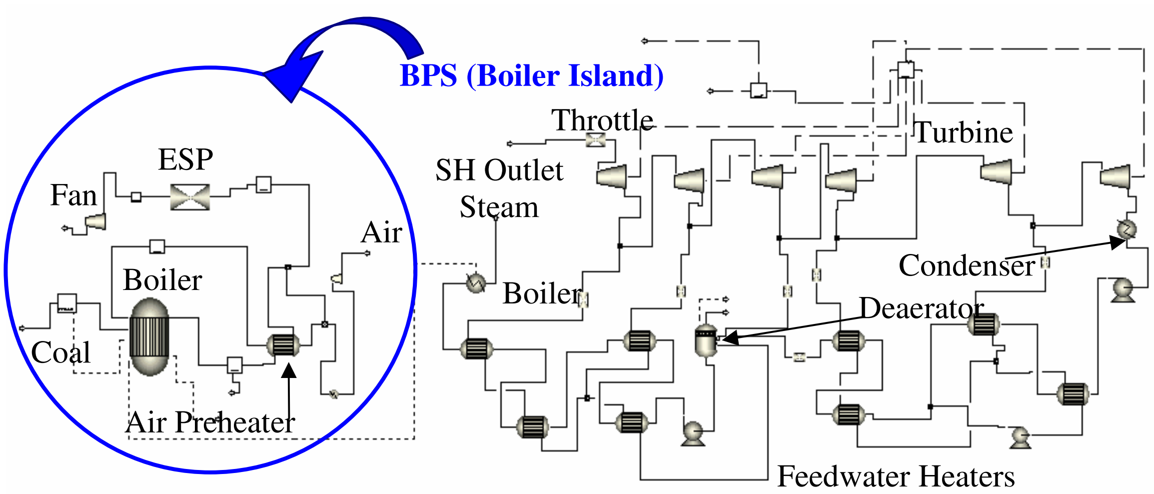


Figure 7.1: Aspen Plus Steam Cycle Model for Demonstration Case 1

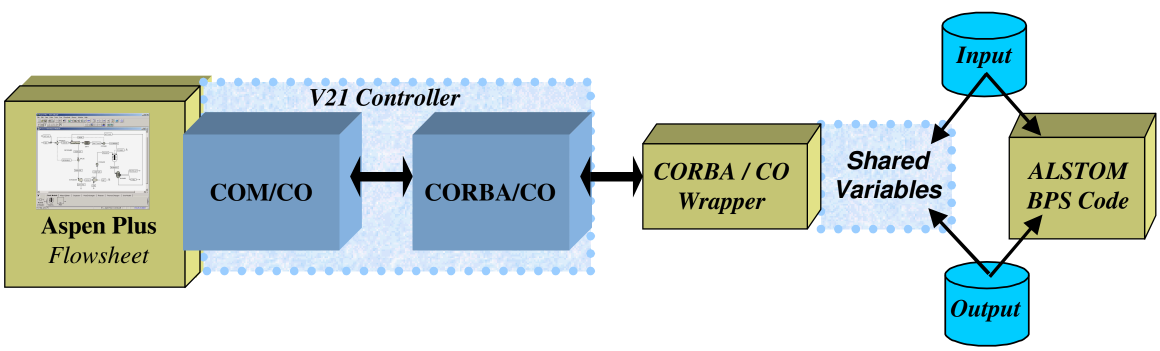


Figure 7.2: Software Controller Configuration for the Proprietary BPS Code

The IC exchanges information with the CORBA/CO Wrapper. For the sake of simplicity and convenience, a file I/O approach was adopted, where the BPS code is sequentially re-initialized and re-executed as a batch run with each flowsheet iteration. Fluent provided a template for the CO-compliant wrapper (written in C++); appropriate modifications and additions were made to the wrapper to accommodate the BPS code requirements. The compiled wrapper was implemented as a dynamic link library (.dll). Approximately three man-weeks were required to complete the coding for the wrapper and debug it. The CO wrapper methodology is essentially similar to the functionality served by the user-defined subroutines in Aspen Plus. However, the CO methodology is an open standard that allows proprietary codes to execute on platforms other than Windows (with any CO-compliant simulator), and permits the code to be used as one of multiple plug-and-play modules in a "Solution Strategy" specification.

In order for the BPS code to execute properly and provide meaningful results, it must receive updated information from the Aspen Plus cycle. A CO variable has been defined for each informational item or parameter that must be passed from Aspen Plus through the IC to the BPS wrapper. The wrapper receives the updated CO parameter values and overwrites the BPS input file to reflect the current state of the shared variables. The wrapper then spawns the execution of the BPS code. It subsequently reads the BPS output file, extracts the specific parameters required by the Aspen Plus cycle, and updates the state of the shared parameters. As with the input, a CO variable has also been defined for each informational item or parameter that must be passed from the BPS wrapper back to the Aspen Plus cycle. A total of 19 CO variables were exchanged between the BPS wrapper and the Aspen Plus cycle.

In general, depending upon the size of the unit and customer needs, the boiler operator may potentially employ any number of sequential control strategies over the range of loads. At high loads, the damper control strategy is employed. For a single excess air value, the damper is moved from the open position (at maximum load) to a closed position. At moderate loads, the excess air control strategy is used. The excess air is increased to a maximum value (e.g., 35% excess air) with the damper locked in the closed position. The high and moderate load range is denoted as the “steam temperature control range”. Over this control range, the boiler is able to make the steam temperature dictated by the turbine. In essence, the cycle dictates the feedwater flow rate and enthalpy, as well as the superheat outlet temperature, and the boiler is able to accommodate both the desired outlet steam temperature and the total heat duty, by adjusting its internal control mechanisms. Over the “steam temperature control range,” the coupling of the boiler component with the rest of the cycle is relegated to “one-way” coupling and the BPS computations become a post-processing operation.

At low loads, designated as the “below control range” loads, the damper is locked in its closed position and the excess air is locked at its allowable maximum. (In practice, the total air flow rate may be locked instead, to prevent fan stall and to permit purging.) At such loads, the boiler is not capable of making the desired steam temperature and the turbine must accept the prevailing boiler superheat outlet temperature. In this instance, rather than matching the “desired steam temperature”, the boiler itself dictates what the turbine inlet temperature will be, thus providing feedback to the steam cycle in the form of “two-way” coupling. The results are shown in Figure 7.3.

Additional results are shown in Figure 7.4 for the mass flow rates of various streams. The linkage of the BPS code to the Aspen Plus package clearly demonstrates that a proprietary industrial legacy code can be effectively utilized as a module on a process flowsheet. Legacy codes are usually well calibrated from many years of use in industrial design applications and bring a measure of accuracy and sophistication to the computation that would not otherwise be available. The versatility to adapt to different control strategies over the load range is an indication of that added sophistication; default Aspen Plus library modules could not respond in that manner.

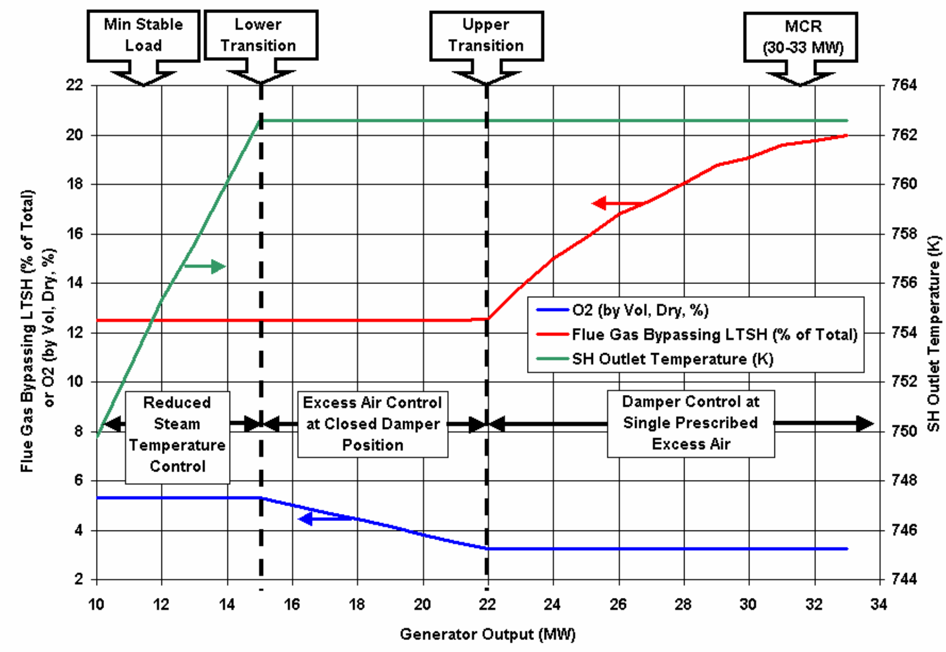


Figure 7.3: Control Parameters and Results for Case 1 with BPS-Aspen Coupling

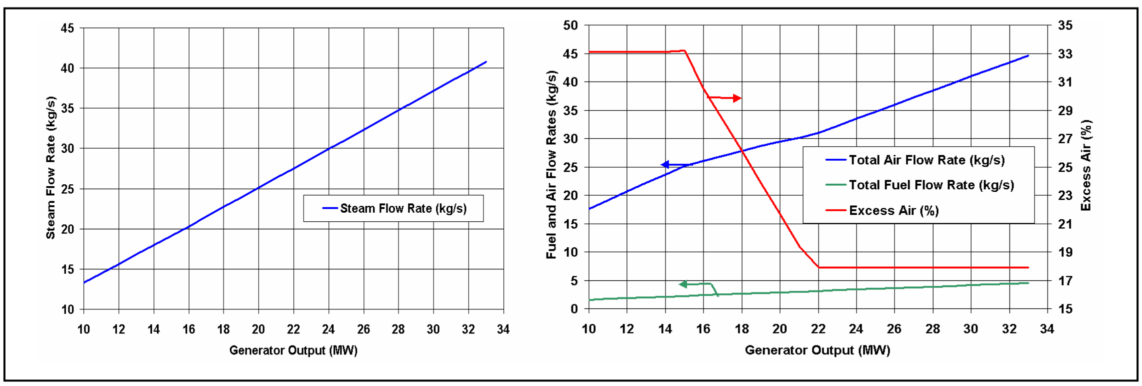


Figure 7.4: Case 1: Run 2 Results – Steam, Fuel, and Air Flow Rates

7.1.2 Case 1 : Run 3 – Aspen Plus Coupled with FLUENT

Run 3 consists of the coupling of the Aspen Plus package with FLUENT Version 6.1.15. A CFD block was instantiated as a CO module on the flowsheet. The resultant cycle is shown in Figure 7.5. For the sake of convenience, auxiliary modules such as the ESP, fans, etc, were not included.

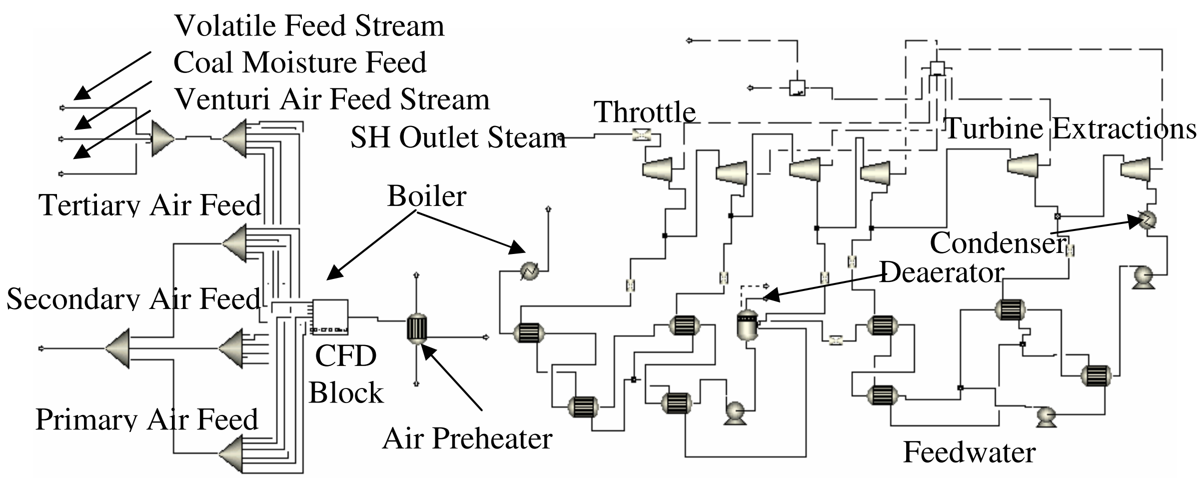


Figure 7.5: Coupled FLUENT and Aspen Plus Cycle for Demonstration Case 1

The boiler has six burners, arranged in two elevations of three burners each. Each burner contains four air inlets, for a total of 24 inlets for the overall boiler. Since coal particle models are much more CPU intensive than gas-only models, the coal was approximated as a gaseous fuel.

As discussed previously for Run 2, at a given load, Aspen Plus externally manipulates the damper position and the excess air (or exit oxygen concentration) in order to achieve the specified steam superheat outlet temperature (or superheat absorption/duty). For any given fuel mass flow rate and excess air, the total air mass flow rate is calculated from stoichiometric reaction relationships.

Ordinarily, for Demonstration Case 1, proper boiler operation at any given load should involve the manipulation of two sets of independent variables: (1) damper position or excess air (which primarily controls the superheat absorption), and (2) the total coal mass flow rate (which controls the total absorption). Converging the boiler CFD model within the Aspen Plus environment as a function of two sets of manipulated variables for a range of loads was viewed as being rather CPU intensive on a PC. Consequently, the link between the boiler and the air preheater air outlet (AHAO) stream was broken, and the coal mass flow rate and the air temperatures to the boiler were hard-wired as a function of load. Only the first set of independent variables was varied in the present computations.

A 40,000-cell FLUENT case/data file was prepared and calibrated prior to coupling with Aspen Plus. Three heat exchanger tube banks or sections were defined in the CFD case as porous media -- a “boiler bank”, a low-temperature superheater (LTSH), and a superheater platen. The porous media inertial resistances were determined from a knowledge of the tube bank geometry and ALSTOM Power design standards. To simulate the bypass damper, the bypass channel was also defined as a porous medium; the inertial resistance for the channel was varied as a control variable.

The 24 material streams in the Aspen Plus flowsheet representing the gaseous inlet streams connect directly with the corresponding port connections defined for the CFD block. A single material stream was connected to the pressure-outlet boundary condition. A user-defined function (UDF) was utilized to calculate the tangential velocity for each of the burner inlet planes.

The cases used the FLUENT Heat Exchanger (HX) Model, a standard software feature allowing simplified two-fluid overlay calculations without explicitly resolving the steam-tube geometry. The HX Model representation was calibrated to match the BPS design code results at the 30 MW condition. The calibration was performed by changing the resistances for the external evaporative walls and by adjusting the surface effectiveness factors of the tube banks. The calibration accounts for the unknown effects of fouling and slagging, radiation shadows, and various other heat transfer inefficiencies. Based solely on the calibration at the 30 MW condition, FLUENT calculations were found to reasonably match the results of the BPS code at lower loads.

A total of 19 CO parameters was defined to exchange information between Aspen Plus and FLUENT. The Scheme language was used to define internal FLUENT variables that could be associated with the corresponding CO parameters.

Typically, for any given load point and steam conditions, Aspen Plus (through a series of "design specs") determines the steam mass flow rate required to produce a particular generator output (e.g., 30 MW). Subsequently, when the boiler or FLUENT block is reached in the cycle, Aspen Plus typically executes another "design spec" which varies the bypass damper control (or excess air) until the superheat outlet temperature achieves the desired target temperature of 762.6 K (913 °F). Some representative convergence characteristics for Demonstration Case 1 are provided in Figure 7.6, where the superheat outlet temperature from the superheat platen is being tracked as a function of FLUENT iteration number. The maximum number of FLUENT iterations allowed, within any given design spec iteration, was set at 600. Good initial conditions were provided in this instance so that convergence would occur within only a few Aspen Plus iterations. At the load points of both 26 and 24 MW, it can be seen that the FLUENT block is executed three times, and on the third attempt, the convergence algorithm was able to estimate the bypass channel resistance that produced the desired superheat outlet temperature. The convergence tolerance arbitrarily prescribed for the superheat temperature is 0.28 K (0.5 °F). In each FLUENT execution, the convergence criteria have been set such that convergence of FLUENT is attained before the maximum iteration limit of 600 iterations is reached. The first iteration at each load required on the order of 500 iterations for the superheat outlet temperature to stabilize; successive iterations stabilized and converged much more quickly.

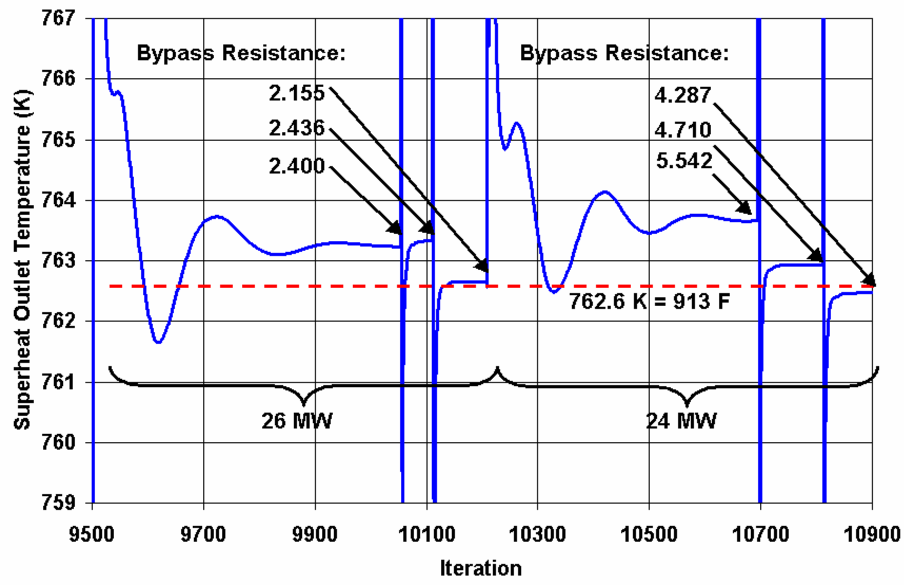


Figure 7.6: Convergence of Superheat Outlet Temperature for Case 1

Results over the load range for the FLUENT/Aspen Plus coupling are similar to those shown previously in Figure 7.3. Aspen Plus, as the executive software, was able to successfully manipulate the specific control parameters in order to produce the desired superheat outlet temperature. The porous media inertial resistance in the bypass channel was the controlling parameter at high loads and the exhaust oxygen concentration was the controlling parameter at moderate loads.

7.2 Demonstration Case 2: Natural Gas Combined Cycle

A natural gas combined cycle (NGCC) power plant was selected for the Case 2 advanced cycle. The power plant consists of an advanced gas turbine, steam turbine, generator, and heat recovery steam generator (HRSG) all supplied by ALSTOM Power. In combined cycle mode, the power plant operates at a net efficiency of 57.5%. The gas turbine generates approximately 2/3 of the 270 MW of electrical output from the NGCC power plant. The gas turbine generator has an efficiency of 38.5% when firing natural gas fuel. The exhaust gas exits the gas turbine at a temperature around 923 K where it enters an HRSG. The HRSG contains both high and low pressure evaporative and superheat surface as well as HP reheat.

The HP feed pump also takes water from the LP steam drum, a small part of which is sent to the gas turbine (GT) cooler. Most of the HP feedwater flows through the HP economizer and then into the “once-through” (i.e., no drum) evaporator section where it exits as slightly superheated steam. The steam is then sent to the HP separator where it is mixed with superheated steam from the GT cooler. The steam is then superheated and conditioned in the HP desuperheater and sent to the HP steam turbine. From the steam turbine outlet, the steam passes through a reheat (RH) section and into the RH desuperheater.

At the maximum continuous rating (MCR), the HRSG is designed to provide a superheat outlet steam flow of approximately 60 kg/s at 838 K and 16.5 MPa. The design reheat steam flow is approximately 59 kg/s at 836 K and 3.6 MPa. The HRSG component and "island" will be the focus of Runs 2 and 3.

7.2.1 Case 2: Run 2 – Aspen Plus Coupled With HRSGPS

Run 1 (not shown) consists of the baseline cycle using Aspen Plus library modules. Run 2 consists of the coupling of an ALSTOM Power proprietary design code with Aspen Plus. The ALSTOM design package selected for use was a heat recovery steam generator performance simulation (HRSGPS) code. It was constructed with Visual Basic and permits user interaction through a graphical interface.

As shown in Figure 7.7, the HRSGPS code essentially represents the entire HRSG island, and it replaces a large segment of the cycle. When the performance simulation code is instantiated as a CAPE-OPEN block in the Aspen Plus flowsheet, it replaces ten of the tube bank modules, in addition to pumps and drums/separators. In the present case, the HRSGPS code constitutes a single block icon on the steam-turbine side of the flowsheet that must interact with and exchange information with the gas turbine portion of the cycle, as well as the remainder of the steam side of the cycle.

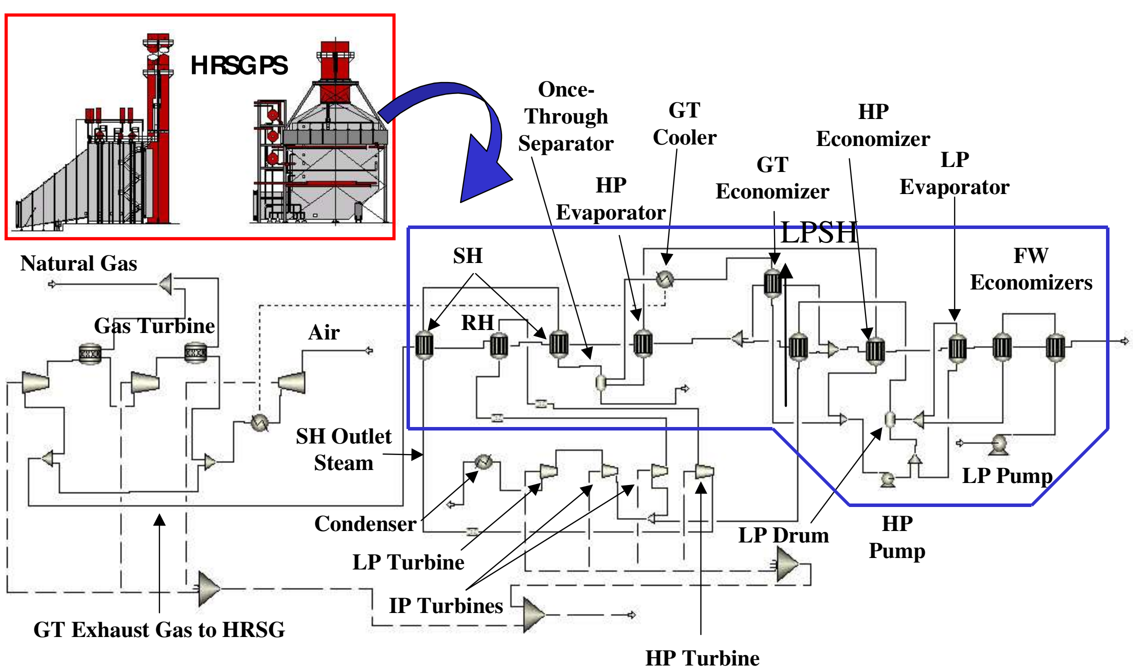


Figure 7.7: Portion of Cycle Replaced by the HRSGPS Code

The HRSGPS package has not been coded with material stream or port connections. Consequently, all of the information exchange between Aspen Plus and the design code must occur through a transfer of CAPE-OPEN parameters. The IC and software wrapper, which act as an interface between the HRSGPS code and Aspen Plus package, assist in the transfer of this information. In stand-alone mode, the user interfaces with the HRSGPS code and provides input through GUI panels. The execution of the code is prompted by the user and is interactive, rather than batch.

For the sake of simplicity and convenience, it was decided to couple the performance simulation code with Aspen Plus in much the same manner as the BPS code. Appropriate modifications and additions were made to the wrapper template by an ALSTOM Power programmer to accommodate the HRSGPS code. Access was granted to the HRSGPS "source" code, so that modifications could be made directly to the Visual Basic coding. This permitted the wrapper interface to function in a somewhat more sophisticated manner than was possible with the BPS code. The compiled wrapper was implemented as a dynamic link library (.dll). Approximately three man-weeks were required for the programmer to complete the wrapper coding.

HRSGPS was converted into a batch execution code that Aspen Plus could launch each time the CO block was encountered in the cycle simulation. In order for the HRSGPS code to provide meaningful results, updated information must be received from the Aspen Plus cycle (see Figure 7.8) prior to each execution. A CO variable was defined for each parameter passed from Aspen Plus through the IC to the HRSGPS wrapper. The wrapper receives the updated CO parameter values and creates a "list file" that contains those values. The wrapper then spawns the execution of the HRSGPS code, which displays the iterative solution results on the screen so that the user can monitor its progress. The HRSGPS code reads its input file and the list file, which overwrites the designated CO parameters in memory to reflect the current state of the shared variables. When the HRSGPS code completes its execution, it writes its normal output file, as well as another "list file" containing the CO parameters to be transferred back to Aspen Plus. As with the input, a CO variable was also defined for each parameter passed from the HRSGPS wrapper back to the Aspen Plus cycle. The wrapper reads the output list file and passes the values to the CO collection in the IC, and from there to Aspen Plus. A total of 44 CO variables were defined.

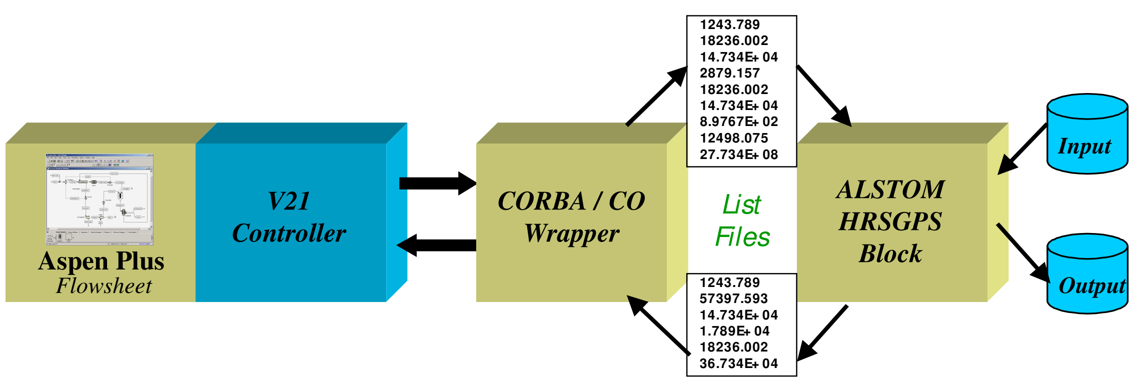


Figure 7.8: Parameter Exchanges Between HRSGPS Code and the Wrapper

The HRSGPS code internally varies the superheat outlet flow rate until the desired superheat outlet temperature is achieved. The superheat outlet flow rate essentially determines what the upstream HP steam flows will be, and the feedwater flow rate is adjusted to provide the requisite mass balance around the LP drum.

Additional results are shown in Figure 7.9 for the mass flow rates of the various streams. The predicted trends are sufficient to demonstrate and assess the viability of the coupling between Aspen Plus and an industrial legacy code, using the Vision 21 Controller and CAPE-OPEN interfaces.

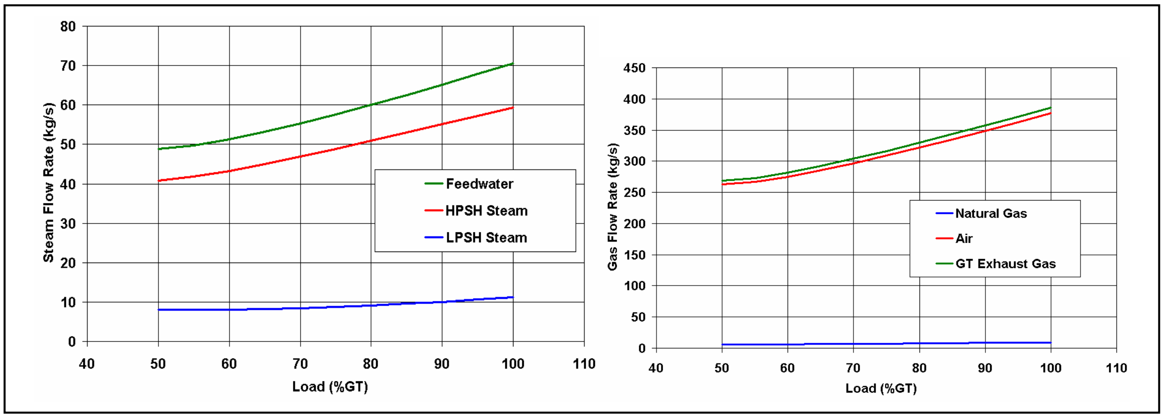


Figure 7.9: Case 2 : Run 2 Results – Steam, Fuel, and Air Flow Rates.

7.2.2 Case 2 : Run 3 – Aspen Plus Coupled with FLUENT

Run 3 consisted of the coupling of Aspen Plus with FLUENT v6.2.13. The CFD block was constructed to represent only the tube bank components within the HRSG. The FLUENT case and data files were prepared and calibrated prior to coupling with Aspen Plus. In order to decrease the computational expense, the CFD grid was reduced to approximately 40,000 cells. The FLUENT HX model was calibrated to match the HRSG design code results at the MCR condition.

The “physical model port” capability was exploited in this run. The term “physical model” refers to FLUENT submodels (such as the discrete particle model or the HX model), which have stream connectivity requirements that are different from those of the typical inlet and outlet boundary conditions for the computational domain. The physical model port connectivities are illustrated in Figure 7.10. Each tube bank in the HRSG (or group of tube banks in series) is represented by a single heat exchanger (HX) icon on the Aspen Plus flowsheet. The inlet stream to each HX icon is duplicated (by a “duplicator block”) and connected to the FLUENT block icon via physical model ports. The physical model port feature allows the steam/water lines on the Aspen Plus flowsheet to be connected directly to the overlay “coolant” fluid of the HX model in FLUENT and decreases the number of CO variables that would otherwise have to be defined in order to transfer the requisite inlet mass flow rate, temperature, pressure, and quality for each tube bank model. The physical model ports complement the regular material stream port connections for the gas-phase flow field.

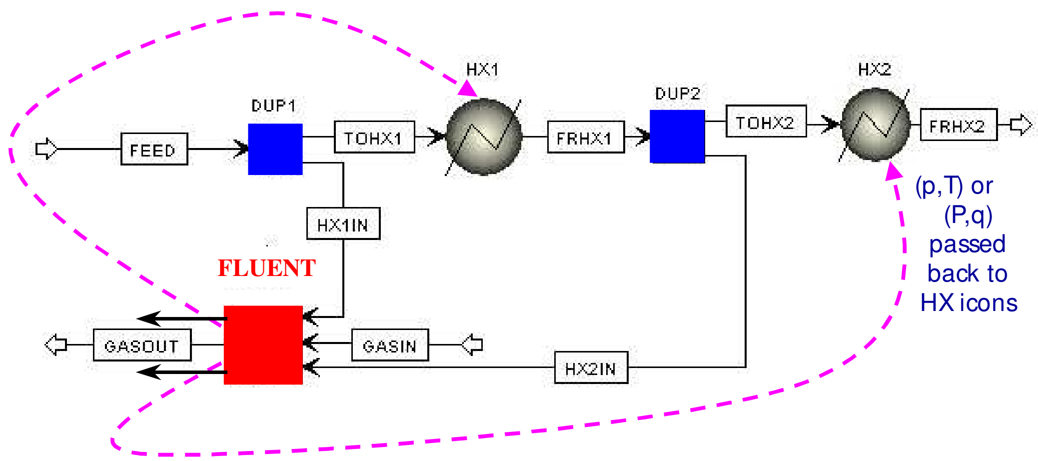


Figure 7.10: Case 2 : Run 3 Solution Strategy

Although physical model ports for each tube bank outlet stream are also constructed and connected to the FLUENT icon, they are not accessed. Instead, CO variables are defined for the outlet parameters of each tube bank and such "informational" CO variables are passed back to Aspen Plus and are used to overwrite the corresponding HX icon variables. For example, if the simplistic (single-phase) HX icon calculates a heat duty on the basis of an assigned outlet temperature (T) and pressure (p), then those same CO variables of (T, p) are passed from the corresponding FLUENT tube bank outlet to the HX icon. If the HX icon represents an evaporative section, where the heat duty is calculated on the basis of a pressure and vapor fraction / quality (q), then those corresponding CO quantities of (p, q) are transferred from FLUENT. The HX icons are solved prior to the FLUENT block, thus allowing tube bank inlet conditions to be transferred to FLUENT which satisfy global mass and energy balances (based on the assigned heat exchanger characteristics). Subsequently, the FLUENT block is converged, and the updated tube bank outlet conditions are passed back to the Aspen Plus HX icons for the next iteration. It the HX icons were not retained on the flowsheet, then some of the inlet physical model port streams attached to FLUENT, which depend on upstream flows which have not yet passed through FLUENT, would not satisfy global mass and energy balances, and overall convergence might be adversely impacted. In the present strategy, mass and energy balances are ensured for all of the HRSG HX icons collectively, prior to the transfer of input stream information to the tube bank models in the FLUENT block. It is believed that the current strategy is advantageous for those CFD modules or blocks which are simultaneously connected to many parts of the cycle.

Following the control strategy for once-through HRSGs, an Aspen Plus "design spec" utility, based on the Broyden method (a modification of Newton's method), manipulates the HP flow rate until the desired superheat outlet temperature is achieved. The Broyden algorithm also simultaneously converges the CO variables (denoted as "Fortran tears") transferred between the FLUENT block and Aspen Plus. For Demonstration Case 2, the six groups of heat exchangers (single or in series) required 24 CO variables (principally for monitoring purposes), of which only 12 were transferred from FLUENT to Aspen Plus (2 per HX icon), and which were manipulated by the Broyden algorithm as "Fortran tears."

An example of the convergence characteristics is presented in Figure 7.11 for the 100% load case. It can be seen that 11 FLUENT executions were required to converge the manipulated flow rate and Fortran tears. The first iteration required on the order of 400 FLUENT gas-phase iterations in order to reach an acceptable residual level. Subsequent executions required fewer iterations.

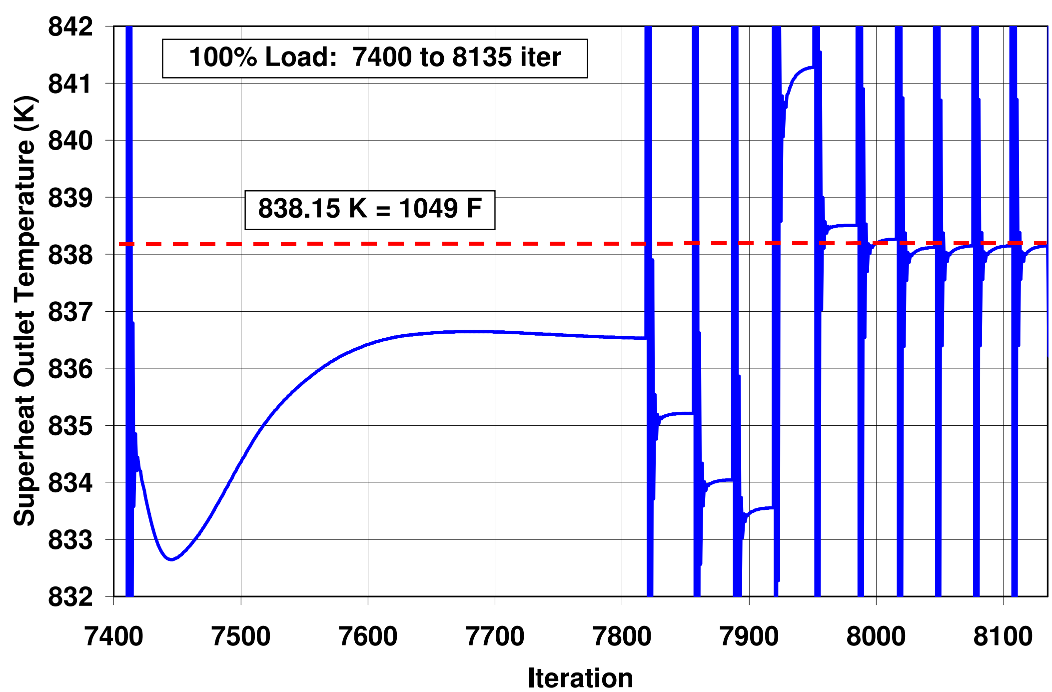


Figure 7.11: Case 2: Run 3 Convergence Characteristics

With the CFD coupling, ten load points were run sequentially and automatically over the range from 100% GT load to 50% GT load using Aspen's "sensitivity analysis" utility. The case was run over a LAN, with Aspen Plus running on a PC and FLUENT running on a single processor of a LINUX platform. Aspen Plus, as the executive software, was able to successfully manipulate the designated control parameters in order to produce the desired superheat outlet temperature. The results were very similar to those shown in Figure 7.9.

7.3 Demonstration Case 3: Integrated Gasification Combined Cycle

The FutureGen IGCC used in Demonstration Case 3 was provided by NETL (Shelton and White, 2004). The cycle consists of a 250 MWe FutureGen IGCC with CO_2 capture, a gas turbine burning hydrogen-enriched syngas, a pressure-swing absorption (PSA) section for hydrogen stream production, an air separation unit (ASU), a heat recovery steam generator (HRSG), and acid-gas cleanup. The HRSG is a three-pressure system (i.e., low pressure (LP), intermediate pressure (IP), and high pressure (HP)), with each line splitting off from the common output of a deaerator and being pressurized with dedicated pumps. Each line consists of various economizer tube banks, one or more evaporative banks with a drum, and a superheat section. The HRSG also has two reheat tube banks. Various fractions of the economizer flows are split off and recycled to other sections of the flowsheet, including the shift reactors, where heat is absorbed and then returned to the HRSG. The HRSG module is the focus of Demonstration Case 3.

As mentioned previously, a conceptualized HRSG was designed using ALSTOM's HRSGPS design code to match the constraints and boundary conditions of the HRSG tube banks represented on the flowsheet. A corresponding FLUENT model of the HRSG was constructed (approximately 40,000 cells) and calibrated, again using the results of the design code. The FLUENT model was then instantiated on the flowsheet and run, in coupled fashion, with the Aspen Plus cycle at the design load.

The Fluent HRSG model was constructed with 18 tube banks. Since some of the tube banks were considered to be in series with each other without intervening junctions (i.e., without flows being split off or added to the streams), some of them were conceptually grouped together to form a composite heat exchanger on the flowsheet. Consequently, only 12 HX icons were placed on the flowsheet. For each composite heat exchanger, 4 CO variables were defined for the tube bank outlet values (mass flow rate, temperature, pressure, and vapor fraction), principally for monitoring purposes. Of these 48 CO variables, 24 were designated as "Fortran tear" variables (half of which were pressure quantities which didn't change), which were converged with a dedicated Broyden algorithm in an interior nested convergence block/loop.

In an HRSG with drums, the usual control strategy, and hence the present solution strategy, is focused on balancing the circulation around the drums. This approach is reflected in Figure 7.12. As described before for Demonstration Case 2, each of the inlet streams to the various HX icons is duplicated; the duplicate stream is connected to the corresponding inlet of the tube bank in the FLUENT HX model. CO variables for the outlet quantities from each tube bank in the FLUENT model are passed back to Aspen plus and associated with their corresponding "Fortran tear" variables in a calculator block. The evaporative section consists of a recycle stream that is torn between the splitter (beneath the drum) and the evaporator HX icon (see the streams FCIRC and RCIRC in Figure 7.12). Presuming that forced circulation (i.e., due to a pump) is producing the recycle stream, the control strategy typically involves setting the circulation flow rate through the evaporator (based on certain criteria) and then modifying the economizer flow rate until the recycle rates are balanced. Accordingly, two design specs (using the secant method) were constructed as external convergence loops: one which manipulated the total deaerator flow until the circulation was balanced on the IP line, and the other which manipulated the split to the HP pump/line until the circulation for the HP line was balanced. The LP line had a relatively low flow rate and the balancing of its circulation rate was ignored in this demonstration.

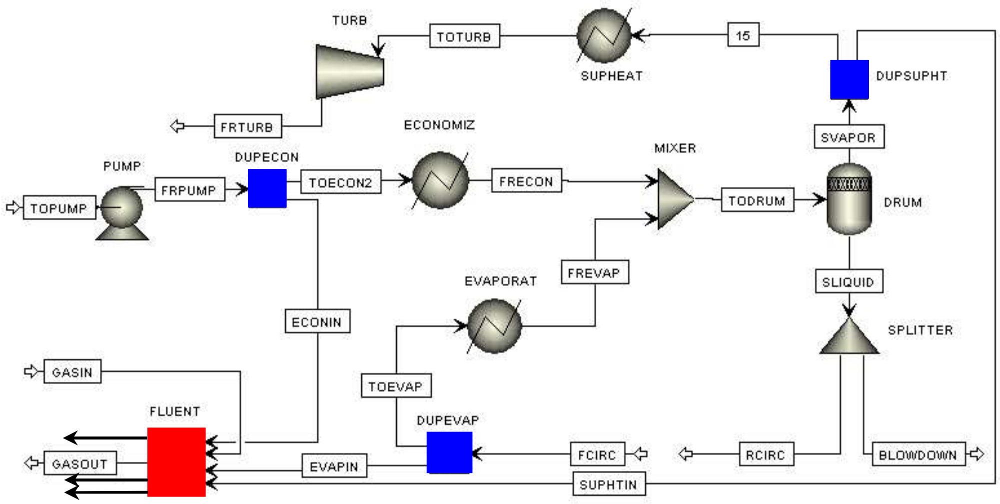


Figure 7.12: Case 3 control schematic

FLUENT executed approximately 80 times before overall convergence was achieved. In the innermost nested "Fortran tear" convergence loop, FLUENT would typically be executed between 3 and 8 times in order to reasonably converge the Fortran tears. Once the Fortran tears were converged, then the outer design spec convergence loops would manipulate their control parameters until the HP and IP evaporative section circulation rates were balanced. The results were reasonable, producing superheat and reheat temperatures that were very close to the original calibrated FLUENT case. As with Demonstration Case 2, this case was also successfully run over a LAN, with Aspen Plus running on a PC, and FLUENT running on a single processor of a LINUX platform.

8 Results and Discussion

8.1 Validation of project goals

Table 8.1 summarizes the main features of the integration toolkit developed in this project and how those help the end user. The table presents a comparison of how particular problems were solved before and how they can be solved now. From this information we conclude that the software developed through this project allows engineers to get better results by using integrated models, while considerably reducing the engineering labor needed to develop such models.

Table 8.1: Features of the integration toolkit and the benefit to end-user

Feature	Benefit	Before	Now
Configuration Wizard	Make CFD models readable by CO simulation executive, such as Aspen Plus.	Takes 2 months for an expert programmer to develop a custom solution.	Takes ½ h for a typical CFD user.
EM Database	Model selection from database based on model features	No. Selection based on filenames only	Yes.
Model edit GUI	Change CFD parameters from Process Simulator GUI	Basic parameters	All CFD parameters
COM-CORBA Bridge	Run integrated simulation on LAN or VPN	No	Yes.
Solution strategy	Swap model types during a simulation to optimize speed/accuracy	No	Yes.
CFD viewer	Display CFD results from inside Process Simulator environment	No	Yes. Select graphics pre-defined by CFD analyst.
Reduced Order Model	Use previous CFD solutions to generate faster estimates	• Manual creation of database/model • Can't switch models	• Automatic creation of database/model • Can switch models
Proprietary model wrapper	Incorporate engineering models used in industry	• Can't switch models • Can't run over a network	• Can switch models • Run on networked machines

8.2 Benefits to DOE and Industry

To ensure immediate and widespread adoption of our software solution, we focused on seamlessly integrating two widely-used commercial software products: Aspen Plus for process simulation and FLUENT for CFD equipment simulation. Each tool is the market and technology leader in its respective field. The use of leading commercial, general-purpose software as the backbone of the simulator infrastructure will ensure that the infrastructure will remain supported and available to the industry far into the future for simulating advanced power plants. The toolkit was also designed to facilitate the integration of additional simulation tools, such as custom equipment models based on legacy proprietary software (also known as “in-house codes”). Using this powerful combination to incorporate detail exactly where it is most needed, DOE and industrial users can conduct process simulations to a level of detail and accuracy never before possible.

The integration toolkit is based on the CAPE-OPEN (CO) standard for interfacing process modeling software components for use in the simulation, design, and operation of processing plants. We provide an easy-to-use template for wrapping legacy models as CO-compliant models that can be included in an integrated simulation. By using the CO standard we ensure that any CFD model or proprietary custom model that uses CO interfaces can be linked to our software framework. This openness is already yielding benefits as the toolkit is being leveraged to other tools not included in this project.

As discussed in Section 9.2, the work progress was presented five times to an Industry Advisory Board, consisting of other Vision 21 participants and representatives of the chemical and power industry, to ensure that the results of the project meet the needs of the industry and are quickly disseminated to the industry.

With the software toolkit developed through this project, DOE and industrial users will be able to capture complete and consistent information contained in models of Vision 21 technology modules and construct accurate virtual-plant simulations of Vision 21 plants. The designers will be able to optimize the overall process design while accounting for constraints imposed by equipment items and to optimize equipment design in the context of the whole power generation system rather than in isolation. Thus the designers will be able to develop conceptual designs of novel power plants, leading to reduced plant life cycle costs and increased energy efficiency dividends to the nation. The long-term potential of the approach has been recognized by the software’s selection as " … one of the 100 most technologically significant products introduced into the marketplace over the past year" in winning the 2004 R&D100 Award (*R&D Magazine*, September 2004).

9 Technology Outreach

The results of two small tasks aimed at technology outreach are discussed in this section. One was a task by Intergraph, a world leader in plant design software, to evaluate the feasibility of integrating physical domain information with the integrated model developed through this project. This is in accordance with the Vision 21 philosophy that a holistic view of plant design and operation will be required to attain the efficiency and emissions targets needed in advanced power plants.

The other task was that of forming an Advisory Board and periodically reviewing the project status. The Advisory Board review was a means for receiving early feedback on the integration software and for disseminating information about the software to the industry.

9.1 Integrating physical plant knowledge

9.1.1 Scope

Building a power plant is often subject to high risk in terms of both cost and schedule. In this project, Intergraph focused on the prospect of reducing this risk by interfacing its design software seamlessly with the simulation results from Aspen Plus and FLUENT. This interface would enable users to determine the impact in the physical design by using 3-D models, making it possible to take corrective actions or make design decisions that will save money and help users better schedule their time. Intergraph's task was to evaluate the feasibility of developing such an interface in a future phase of the project.

Another task was to evaluate the use Intergraph software for data management in the work process between the different participants. This role included version management of process scenarios and management of documents and plant engineering data.

9.1.2 Concept studies

The first phase involved gaining an understanding of the data flow created by Aspen Plus and FLUENT. Several Fluent and Aspen Tech meetings were attended by members of the Intergraph Vision 21 team. The second phase covered the scope description of a prototype. The goal was to make the process design data accessible through the physical plant design model.

A power plant model was obtained from ALSTOM Power in France. The process and engineering data was stored in SmartPlant® Foundation, an Intergraph engineering database. The 3-D model was displayed using SmartPlant Review, an Intergraph visualization environment. Additional code was developed to make two-way communication possible between the 3-D plant items and the engineering database. In addition, the functional process design was expanded into the logical design in SmartPlant P&ID, which would facilitate the design of the 3-D model and the control system in the detailed design phase of the plant lifecycle.

Including the process and plant engineering data in the SmartPlant Foundation engineering database also creates the opportunity to interface with ERP systems such as SAP to address the business side of the project lifecycle.

During plant operation, plant data access and compliance with safety and health regulations is key. Intergraph's engineering database supports these requirements with easy data access through the relationships between plant items. Also, SmartPlant Foundation retains all data over time, which enables users to set the time back to verify the plant configuration at any point in time.

The prototype was presented at the DOE meeting in November 2001. In the scenario, the plant structure was built in the SmartPlant Foundation environment, for example, Plant ABC unit 12, unit 15, etc. By navigating through the plant structure, the user selected a system in the unit. Next, the 3-D representation was displayed with the system highlighted. Items in the system such as valves, pipelines, and tanks could then be selected, and related process data and/or documents could be displayed. The data could be found by using SmartPlant Foundation or the 3-D model.

This prototype enabled users to quickly find plant systems and their related data. Different process scenarios could be attached to the plant items for optimization studies. To facilitate designing and building the plant, a link was created between Aspen Plus PFD and Intergraph SmartPlant P&ID.

The PFD from Aspen Plus was expanded using SmartPlant P&ID, which shows all plant items in a schematic representation. The P&ID is a key document for the design, building, and operation of a plant, because it is the roadmap for all plant process systems. The P&ID data and documentation are also stored in SmartPlant Foundation to support the plant lifecycle phases.

In the third phase, Intergraph focused on the issues related to better managing the information in the early stages of design. Process design involves generating numerous simulation scenarios to help specify design. Each simulation run consists of enormous amounts of data, but most importantly, it includes a significant amount of metadata that represents the characteristics of each run. Typically, each run is stored in the form of documents in a file system, and there is no simple way to review the information in each run other than to pull the results up on the native application. Most often, the run results need to be reviewed by someone who is not a direct user of the native application, resulting in confusion and potential errors.

The ability to better evaluate and utilize the information gathered in simulation runs will reduce the time and effort needed to generate these runs. It also promotes proper specification of run criteria to maximize the value per run.

Intergraph proposed and built a system using SmartPlant Foundation to manage the simulation information generated by Aspen Plus and FLUENT. As part of the system, a data model was defined that shadowed the PFD used in the simulations. Each item in the PFD and its respective connectivity were part of the data model, allowing the Intergraph team to store relevant metadata about the inputs or outputs from each run, on each object, as well as navigate from one object to another in the PFD along the connected lines. This made it possible to discover the metadata on each stream for each run, or to determine the runs that produced certain metadata for a certain stream.

A system of this nature could begin life at the process design stage and continue to accumulate data during the engineering, procurement, construction, and operations phases, providing a valuable continuity of data throughout the plant lifecycle.

9.1.3 Conclusions and recommendations

The ability to link the process design with the physical design in the front-end concept phase as well as later in the design phase offers many opportunities to lower project execution risks. This approach can help power plant designers make the right decisions at the right time and avoid any surprises.

The Intergraph activity has demonstrated that it is possible to link data using a single engineering database to store and manage all plant data for the complete plant lifecycle. In addition, this approach enables users to interface with other mission-critical plant systems, such as ERP (SAP) or maintenance analysis systems (MAXIMO).

The Vision 21 approach has great potential to reduce operational costs and increase plant safety and uptime. Also, new process cases can feed to detailed design for optimization or revamp studies. The current prototype could not show a live change, for example, vessel size changes or fluid level changes in a vessel, but this capability is certainly in the realm of future possibilities.

To achieve the project goal, Intergraph recommends building the process design and engineering solution on top of the plant-engineering database to facilitate data integration and data management between various tasks. Individual tasks can remain unchanged, and the user can select a set of solutions that best fits their own work process. The single engineering database will fit across the complete plant lifecycle, from concept design to operations. Further development and study should be performed to facilitate the ‘live’ update of the 3-D plant model, based on the process cases, to show the impacts of these cases on the physical design.

9.2 Advisory Board

The software toolkit developed under this project integrates a complex set of component models. It was expected that requirements could not be fully defined before the start of the project because the users would need to see the software in action before they could articulate their needs. Therefore, the project team felt that it was important to periodically solicit feedback from the end-user. An Advisory Board consisting of other Vision21 project participants and potential users of the software was formed. Five Advisory Board meetings were held during the course of the project, involving a total of twenty-six engineers outside of the project team. The organizations represented are listed in the Acknowledgments section of this report. During the Advisory Board meetings the project team gave an overview of the project status and future plans and demonstrated the latest versions of the software. Also the Advisory Board members were given access to a password-protected web page that contained the project documentation (user and software requirement documents, design documents, presentations and papers, and meeting minutes). The comments and questions of the Advisory Board members were summarized in the meeting minutes. Several Advisory Board members provided written comments in addition to verbal comments at the meeting. A summary of the comments and responses is presented in Table 9.1.

Table 9.1: Summary of Advisory Board comments and responses

	Advisory Board Comment	Response
1	**User Interface:** "A suitable user interface, which makes CFD accessible to process simulation resources"	Model Select and Edit GUIs implemented.
2	**Ability to make geometric changes:** "the capability to change geometric variables and automatically mesh the new geometry," "optimize … the geometric size of the FLUENT component," "hooks into the CFD model that will permit mesh refinement and geometry changes," "automate the meshing to respond to volumetric or geometric changes/scaling."	Briefly investigated but deferred because a general robust capability is beyond the scope of this project.
3	**Model consistency:** "Thermodynamic consistency and foreign object convergence need more attention. Thus, the developer is responsible for insuring that consistent physical properties are used."	Capability for transferring pure component physical properties and reaction kinetics was developed.
4	**Computational Speed:** "Computational speed is clearly the biggest issue. I would like to see Aspen and Fluent work on developing hybrid unit operation models." "I have an expectation that large flowsheets with one or more CFD-simulated unit operations might be prohibitive if it is necessary to run each CFD simulation to a fairly high degree of convergence for each iteration through the flowsheet."	ROM framework was developed to address this issue. Further development and testing of ROMs will be a priority in follow-on work beyond this project.
5	**Commercialization:** "Commercial development goals need to be delineated." "…look ahead at the marketing of this V21 controller and interface technology"	Fluent started marketing the software toolkit (as a service) in November 2003.
6	**Mixture properties:** "The evaluation of mixture properties (single or multiphase) is still left to the CFD code. There is vast untapped potential to improve the CFD calculation by exploiting the full capability of the flowsheet model to do the full thermodynamic calculations – phase equilibrium, non-ideal vapors etc. as rigorously as possible within the CFD model"	Deferred because developing this desirable capability is beyond the scope of this project.
7	**Transient simulation:** "Will this work be extended to transient simulations?" "Target areas that are important for the chemical industry include transient cycle dynamics …"	Deferred because this is beyond the scope of this project. This capability is planned for follow-on work.
8	**Access to FLUENT process:** "FLUENT solution convergence should not be gauged by residuals alone, but also by some other monitored parameter (e.g., reactor conversion). Since the FLUENT interface is somewhat hidden behind the Aspen Plus interface, the user will lose the ability to monitor convergence based on other parameters." "When Aspen launches a CFD case and the CFD case runs remotely on another machine, would it be possible for the convergence history of the residuals to be displayed on the same host that Aspen is running on?"	Implemented the display of solution residuals on the same machine as Aspen Plus, and added a multithreading capability to the GUI to allow the user to interact with FLUENT when the two processes are collocated

Table 9.1: Summary of Advisory Board comments and responses (continued)

	Advisory Board Comment	Response
9	**Tight Coupling:** "A unit operation may be either a FLUENT calculation or an Aspen library component calculation, but can it also be a hybrid of the two types – e.g., a component that uses the FLUENT hydrodynamics and Aspen kinetics solvers/properties?" "Target areas that are important for the chemical industry include…tight coupling computations, along the lines of gPROMS"	Although the present software capability does not preclude the development of tightly coupled models, this capability has not been tested. Such testing is planned for follow-on work.
10	**Link to P&ID:** "The typical procedure in industry is to do a process model in Aspen Plus or ChemCAD, and then send the results to an engineering group to complete a P&ID. If the transfer of information is done manually, the transfer of information … and the subsequent analysis and checking of the P&ID, takes a long time (several weeks). Is it possible to couple the process modeling with the P&ID in order to shorten the analysis response and feedback cycle time?"	Although this is beyond the scope of the current project, the issue was discussed with Intergraph. The capability for automatically creating a P&ID from a process model is yet to be developed.
11	**Collaboration/realistic examples:** "Why doesn't the team model a Vision-21 "zero-emissions" plant?" "The team ought to establish linkages with developers of the technology modules to get access to their process and component models. "It would be desirable to simulate more realistic (impressive) cases."	Two realistic problems and a FutureGen IGCC have been simulated in collaboration with ALSTOM Power. Opportunities for collaboration with others will be sought in the future.
12	**Extensibility of the Software:** "I am concerned about the predictive power of the models when unconventional components (e.g., fuel cells) are included in the simulation. In other words, the team is executing this project with a simple steam cycle and conventional cycle components. Will the communications and controller interface still perform adequately with Vision 21 cycles and components?"	The software design does not preclude such components. A simulation including a fuel cell validated this extensibility (Syamlal [B17], O'Brien 2003 [10], Zitney 2004 [11]).
13	**Solution Strategy:** "It may be prudent to provide a lumped parameter model that acts as an executive and allows the user to hide functionality that isn't important or essential during the initial stages of convergence, but which allows the detailed functionality to emerge when the user needs the additional refinement."	This is satisfied by the solution strategy and ROM capabilities of the Integration Controller.
14	**Parallel Computing:** "One idea is for the controller to launch multiple CFD cases simultaneously, each with a different value of the primary parameter being varied. Those cases are then held in reserve or virtual storage until they are accessed. Then, each time the cycle iterates or parametrically changes run conditions, it surveys the cases held in reserve and selects the case which is most similar to the desired run condition. The selected case may be used either as a restart file, or as a basis for interpolation, for the current run."	Although the current software permits parallel computation of CFD model, the capability for the development of ROM using parallel simulations has not been implemented. This option will be evaluated in a future phase of the project.
15	**2-D Data Transfer:** "Will it be possible to transfer maldistributions at the outlet of one unit operation to the inlet of another unit operation?"	It is not possible to do this within the context of process simulation.
16	**Remote Execution:** "Does a user have the option of launching a CFD case (remotely) from the V-21 Controller and running it in parallel?"	Implemented.
17	**Consistency of proprietary models:** "(Some) industrial partners have stipulated that they do not intend to rewrite their proprietary, legacy codes, and that they will not release them. Therefore, how does one ensure that properties are consistent between Aspen Plus and the third-party codes, and that solution or convergence discontinuities are not caused by inconsistent properties?"	A general solution for this problem is not known, because the legacy codes do not conform to any standards and usually have physical properties buried deep in the codes.

10 Conclusions

10.1 Program relevancy

This project relates primarily to the "Advanced Computational Modeling" objective listed as a Supporting Technology in DOE's Vision 21 Program description. The following table presents relevant statements from the Vision 21 Program Plan [1] and shows how the current project has made progress on those items.

Table 10.1: Fulfillment of Vision 21 Program Plan Objectives

No	Quotes from Vision 21 Program Plan	Relevancy of work completed in this project
1	The concept of the virtual demonstration is to unify all computer related activities of plant design into an integrated suite of codes, which can exchange information easily and accurately.	The project made considerable progress in unifying commercial software used for steady state process domain modeling. The integration of physical plant software and dynamics software remains to be done.
2	The geometrical and materials information can also be shared with analysis programs for use with computational fluid dynamics (CFD) or structural/stress analysis (computer aided engineering, CAE) software. This will allow "virtual" analysis of the broad details of the simulations to be determined by the process analysis software, which will also be able to communicate within this suite of codes.	We have established seamless communication between process analysis and CFD software, as well as other custom equipment models. The transfer of physical properties and reaction kinetics data between process analysis and CFD models has been achieved. The transfer of geometric information and integration of structural analysis codes remains to be done. The feasibility of integrating physical plant knowledge was investigated.
3	Develop computer simulations for complex plants, including co-production plants. To the extent possible, verify that the simulator is accurate by comparing to actual facilities.	We conducted simulations of two existing power plants as well as a conceptual FutureGen IGCC plant. The tests indicate that the simulations are able to capture actual plant behavior reasonably well.
4	Develop a computer simulation to "demonstrate" integration of new enabling and enhanced technology modules.	Although this objective was not the focus of the present project, some progress has been made in collaboration with NETL in integrating new models such as a fuel cell and gasifier (e.g., Syamlal [B17], O'Brien 2003 [10], Zitney 2004 [11]).
5	Products of the Vision 21 Program: Improved design and simulation tools: Software and design tools, including the virtual demonstration computer simulation developed for Vision 21, will be available for application to the design of other energy and environmental systems.	Based on general-purpose software, the integration toolkit can improve design of systems in the power, chemical, petroleum, and oil & gas industries. The integration software was made commercially available by Fluent in November 2003, and it won the 2004 R&D100 Award.

10.2 Future work

A fifteen-year roadmap for developing a virtual-plant simulator with the capabilities described above is shown in Figure 10.1. This roadmap is based on information from road-mapping workshops conducted by DOE (see Simulation Workshop link at www.netl.doe.gov/coalpower/vision21/index.html), Vision 21 Program Plan [1], and experience gained from this project. To complete such a roadmap, considerable discussion among the stakeholders must take place. What is proposed here is only a starting point for a recommended follow-on project.

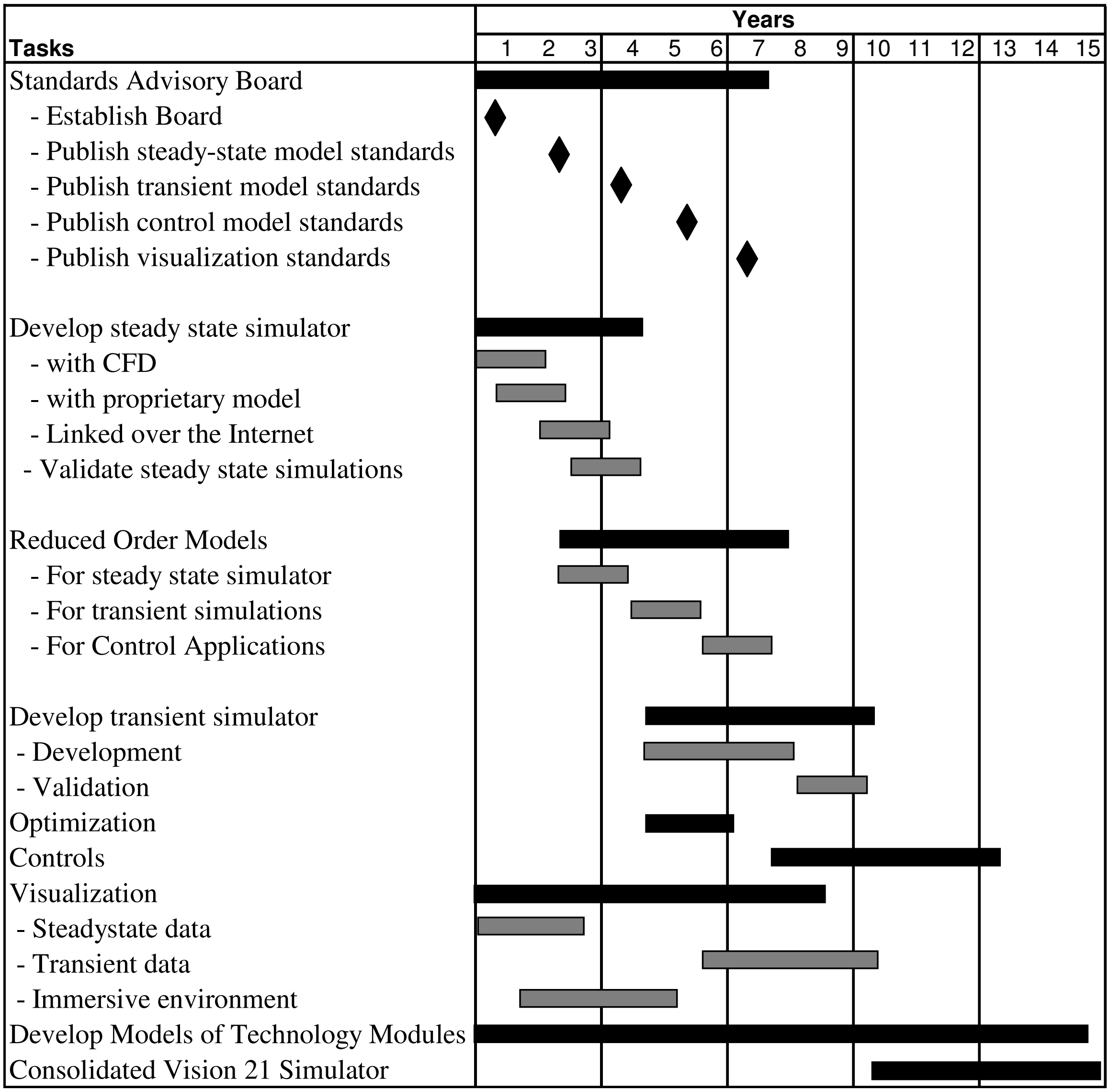


Figure 10.1: Virtual-plant simulator roadmap

An important first task is to establish a Standards Advisory Board (AB). The board should include representatives from DOE, the power generation industry, technology developers, A&E firms, and software companies. The AB should be chartered to develop or adapt a set of standards for interfacing models and information generated about the technology modules. This will involve examining the suitability of standards like CAPE-OPEN and CCA (Armstrong et al., 1999) for exchanging information between computer models. The AB should publish a series of recommendations for steady-state models, transient models, control models, and for model visualization. Technology developers should be encouraged to produce models that conform to the published standards.

The steady-state simulator needs to be further developed as a means for plant designers to evaluate plant designs. Remaining needs include simplification of case setup, more effective use of distributed computing (including multithreading), web deployment with the associated security provisions to safeguard proprietary codes and data, more thorough unification of steam tables and property databases, infrastructure for automated geometric variations, and linking with structural analysis tools. The simulator should be thoroughly validated with simulations of realistic Vision 21 plant concepts.

The computational time required by CFD models is usually too large for conducting a number of integrated plant simulations. Therefore, reduced order models (ROM) derived from CFD data should be developed. These ROM should be used initially in steady-state simulators. The ROM then should be extended (or simplified) for use in transient simulators. Finally, the ROM should be further extended for control system applications. An advanced database approach to storage of data and metadata from CFD simulations will be needed to support the ROM capability.

A software infrastructure for conducting transient simulations should be developed. The transient simulator should have the ability to incorporate CFD models, proprietary models, and reduced order models. The transient simulator should be validated using realistic plant simulations.

The capability for conducting optimization of equipment items using the simulator should be developed. This task may require the development of advanced optimization schemes. The integrated environment will allow the optimization of equipment items in the context of the entire Vision 21 plant rather than in isolation, so that a global improvement is achieved. The ability to develop control system models from the transient simulator should be developed with the introduction and integration of neural nets, genetic algorithms, annealing strategies, etc.

The simulators will generate an immense amount of data. A general visualization capability should be developed to visualize the data generated. Initial efforts should focus on visualizing the data from the steady-state simulator. The ability to visualize and manipulate the data in an *immersive* or virtual reality environment should be developed. The ability to visualize the transient data should be developed. The faster computers and algorithms available over the next decade will facilitate this.

While the software interface standards and the integration infrastructure are being developed, the technology developers should be developing technology modules, which constitute the bulk of the effort for developing Vision 21 plants. The information and data collected from the technology modules should be put into a form that conforms to the standards published by the Standards Advisory Board.

11 References

1. Vision 21 Program Plan, Clean Energy Plants for the 21st Century, Federal Energy Technology Center, April 1999. (www.netl.doe.gov/publications/others/vision21/v21.pdf)
2. Virji, M.B.V., P.L. Adcock, P.J. Mitchell, G. Cooley, "Effect of Operating Pressure on the System Efficiency of a Methane-fuelled Solid Polymer Fuel Cell Power Source, " Journal of Power Sources, 71, 337-347 (1998).
3. Spiegel, M.R. and L.J. Stephens (2000): "Schaum's Outline of Statistics", 3rd Ed. McGraw-Hill
4. Gamma, E. et al (1995): "Design Patterns: elements of reusable object-oriented software", Addison-Wesley
5. "Software Integration Toolkit for Advanced Process Simulation", User's Manual v1.0.4, Fluent Inc., 10 Cavendish Court, Lebanon NH 03766, May 2004.
6. Sloan, D. G. and W. A. Fiveland, "Final Simulation Results For Demonstration Cases 1 And 2," ALSTOM Power Report, October 15. 2003.
7. Pons, M., "Industrial implementations of the CAPE-OPEN standard," Proc. of the sixth Italian Conference on Chemical and Process Engineering (ICheaP-6), June 8-11, Pisa, Italy (2003).
8. Braunschweig, B. L., C. C. Pantelides, H. I. Britt, and S. Sama, "Process Modeling: The Promise of Open Software Architectures," Chem. Eng. Progress, 96(9), 65-76, September 2000.
9. Armstrong, R., et. al., "Toward a Common Component Architecture for High-Performance Scientific Computing," Proceedings of the 1999 Conference on High Performance Distributed Computing. (www.cca-forum.org/./index.html)
10. O'Brien, T.J., S.E. Zitney, W.A. Rogers, and A. Cugini, "Simulation Tools for the Design of Vision 21 Energy Plants," Presented at the 226th ACS National Meeting, September 7-11, New York, NY (2003).
11. Zitney, S.E., M.T. Prinkey, M. Shahnam, and W.A. Rogers, "Coupled CFD and Process Simulation of a Fuel Cell Auxiliary Power Unit," Presented at the 2nd International Conference on Fuel Cell Science, Engineering and Technology, June 14-16, Rochester, NY (2004).
12. Sloan, D.G., W.A. Fiveland, S.E. Zitney, and M. Syamlal, "Software Integration for Power Plant Simulations," Presented at The 27th International Technical Conference on Coal Utilization & Fuel Systems, March 4-7, Clearwater, Florida (2002).
13. Sloan, D.G., W.A. Fiveland, S.E. Zitney, and M. Syamlal, "Power Plant Simulations Using Process Analysis Software Linked to Advanced Modules," Presented at The 29th International Technical Conference on Coal Utilization & Fuel Systems, April 18-22, Clearwater, Florida (2004).
14. Shelton, W. W. and C. W. White, "The Application of an In Situ Oxygen Membrane for an IGCC Process with CO2 Capture," EG&G Final Report, August (2004).

12 Bibliography

This section lists the published and unpublished documentation resulting from this project.

- Papers, presentations and reports resulting

B1. Haidari, A., L. Oshinowo, M. Syamlal, and S.E. Zitney, "Integration of Computational Fluid Dynamics and Process Simulation," Presented at the AspenTech Users Conference, February 4-7, Houston, TX (2001).

B2. Zitney, S.E., M. Syamlal, and R. Hockley, "Integration of AspenTech's Process Simulation Tools with the Fluent CFD System," Presented at the 2001 Fluent UK Meeting on CFD in the Process Industries, September 20, Sheffield, England (2001).

B3. Zitney, S.E., T. Scheele, M. Syamlal, and S. Martens, and, "Improved Process Design from Integrating Aspen Plus and Fluent," Presented at the 2001 Fluent Germany Users Group Meeing, September 17-18, Bingen, Germany (2001).

B4. Zitney, S.E., R. Hockley, T. Scheele, and M. Syamlal, "Using Fluid Flow Models in a Process Simulator for Detailed Plant Design," Presented at the SIMS 2001: 42nd Conference on Simulation and Modeling, October 8-9, Porsgrunn, Norway (2001).

B5. Zitney, S.E. and M. Syamlal, "Integration of CFD and Process Simulation for Improved Reactor Design," Presented at the AIChE 2001 Annual Meeting, November 4-9, Paper No. 271g, Reno, NV (2001).

B6. Syamlal, M, P.E. Felix, M.O. Osawe, W.A. Fiveland, D.G. Sloan, S.E. Zitney, F. Joop, J. Cleetus, and I.B. Lapshin, "Virtual Simulation of Vision 21 Energy Plants," Presented at the U.S. DOE NETL Vision 21 Program Review Meeting, November 6-7, Morgantown, WV (2001).

B7. Syamlal, M. and S.E. Zitney, "Virtual Demonstration to Model Future Power Plants," Presented at the 2001 International Joint Power Generation Conference, June 4-7, New Orleans, Louisiana (2001).

B8. Zitney, S.E., "Integration of Aspen Plus and Fluent CFD using CAPE-OPEN Interfaces," Presented at the Global CAPE-OPEN Final Meeting, Cambridge, UK, December 5-6 (2001).

B9. Sloan, D.G. and W.A. Fiveland, "Discussion On Test And Demonstration Case 1" ALSTOM Power Report, April 30, 2001.

B10. Sloan, D.G., W.A. Fiveland, S.E. Zitney, and M. Syamlal, "Software Integration for Power Plant Simulations," Presented at The 27th International Technical Conference on Coal Utilization & Fuel Systems, March 4-7, Clearwater, Florida (2002).

B11. Zitney, S.E. and M. Syamlal, "Integrated Process Simulation and CFD for Improved Process Engineering," Presented at the ESCAPE-12 Conference, May 26-29, The Hague, The Netherlands (2002).

B12. Zitney, S.E. and M. Syamlal, "Integrated Process Simulation and CFD for Improved Process Engineering," *Proc. of the European Symposium on Computer Aided Process Engineering –12, ESCAPE-12*, May 26-29, The Hague, The Netherlands, 2002; J. Grievink and J. van Schijndel, (Eds.), pp. 397-402 (2002).

B13. Richards, G., D.G. Sloan, and W.A. Fiveland, "Discussion On Test And Demonstration Case 2" ALSTOM Power Report, August 31, 2002.

B14. Osawe, M.O, P. Felix, M. Syamlal, I. Lapshin, K.J. Cleetus, and S.E. Zitney, "An Integrated Process Simulation and CFD Environment Using the CAPE-OPEN Interface Specifications," Presented at the AIChE 2002 Annual Meeting, November 3-8, Paper No. 250c, Indianapolis, IN (2002).

B15. Syamlal, M, P.E. Felix, M.O. Osawe, K. Thotapalli, W.A. Fiveland, D.G. Sloan, S.E. Zitney, F. Joop, P. Simon, J. Cleetus, and I.B. Lapshin, "Virtual Simulation of Vision 21 Energy Plants," Presented at the Vision 21 Simulation Meeting, November 19-20, Iowa State University, IA, WV (2002).

B16. Syamlal, M, and W. Fiveland, "Roadmap for the Development of a Vision 21 Simulator," Presented at The 29th International Technical Conference on Coal Utilization & Fuel Systems, Clearwater, Florida, March 10-14, 2003.

B17. Syamlal, M. J.I. Madsen, W.A. Rogers, and S.E. Zitney, "Application of an Integrated Process Simulation and CFD Environment to Model Fuel Cell Systems," Presented at the AIChE 2003 Spring Meeting, March 30 - April 3, New Orleans, LA (2003).

B18. Sloan, D.G. and W.A. Fiveland, "Final Simulation Results for Demonstration Cases 1 and 2," ALSTOM Power Report, October 15 2003.

B19. Syamlal, M., S.E. Zitney, and M.O. Osawe, "Using CAPE-OPEN Interfaces to Integrate Process Simulation and CFD," CAPE-OPEN Update Newsletter, Volume 7, January (2004). (http://colan.adduce.de/portal/News/COupdateIssues/CO_Update_7/Fluent_Article.html)

B20. Sloan, D.G., W.A. Fiveland, S.E. Zitney, and M. Syamlal, "Power Plant Simulations Using Process Analysis Software Linked to Advanced Modules," Presented at The 29th International Technical Conference on Coal Utilization & Fuel Systems, April 18-22, Clearwater, Florida (2004).

B21. Syamlal, M. and M. Osawe, "Reduced Order Models for CFD models Integrated with Process Simulation," Presented at AIChE annual meeting, Austin, TX, November 2004.

- Project documents posted on password controlled project web site
  - Software Development
    - Defects and Enhancements
    - User Requirements Documents
    - Use Cases in Pictures with Annotation
    - Software Requirements Documents
    - V21 coding style
    - COM-CORBA Bridge design
    - V21 Controller Software Design Document
    - Object Modeling Process
    - June 2002 Advisory Board Demo script
    - Design Review Meeting minutes
    - Software Development Plan
    - Weekly development meeting minutes
    - User's manual [5]
  - Software Demonstration
    - Report on Demonstration Case 1 selection
    - Report on Demonstration Case 2 selection
    - Final Report on demonstration cases
  - Project Management
    - Project management plans
    - Gantt Chart
    - Project quarterly reports
    - Project meeting minutes
  - Advisory Board
    - Minutes of Advisory Board review
    - Review reports by Advisory Board members

13 List of Acronyms and Abbreviations

Name	**Description**
AHGO	Air Heater Gas Outlet (e.g., referring to the flue gas exit temperature from the air preheater)
AHAO	Air Heater Air Outlet (e.g., referring to the air gas exit temperature from the air preheater; after the air preheater, the heated air goes into the boiler)
API	Application Programming Interface.
BPS	ALSTOM Power in-house code for the analysis and design of industrial boilers.
C++	C++ programming language.
CAPE-OPEN	Computer Aided Process Engineering – Open Simulation Environment Interface definitions for exchanging information with process simulation software (www.colan.org).
CCA	Common Component Architecture
CFD	Computational Fluid Dynamics.
COM	Component Object Model – Refers to both a specification and implementation developed by Microsoft Corporation that provides a framework for integrating software components.
CORBA	Common Object Request Broker Architecture – a specification of a standard architecture for object request brokers (ORBs).
CORTEX	Fluent's user interface engine.
CSTR	Continuous Stirred Tank Reactor.
DCOM	Distributed Component Object Model – An extension of COM that allows software components to be distributed over a network.
DOE	U.S. Department of Energy.
EM	Equipment model
ERP	Enterprise Resource Planning
GCO	Global CAPE-OPEN, an extension of the CAPE-OPEN project. (www.global-cape-open.org)
GT	Gas turbine
GUI	Graphical User Interface.
HX	FLUENT heat exchanger module.
HRSG	Heat recovery steam generator.
HRSGPS	ALSTOM Power in-house code for simulating HRSG.
IC	Integration Controller. See definition under V21 Controller.
IDL	Interface definition language, which is used for defining the communications between software components linked through a middleware.
IR	CORBA Implementation Repository
LAN	Local Area Network
LTSH	Low temperature superheater.
NETL	National Energy Technology Laboratory.
NGCC	Natural gas combined cycle.
P&ID	Piping and Instrumentation Diagram.

PFD	Process Flow Diagram.
Scheme	Programming language used in CORTEX
SRD	Software Requirements Document.
TUI	FLUENT Text User Interface.
URD	User Requirements Document.
VPN	Virtual Private Network
XML	Extensible Markup Language: A metalanguage -- a language for describing other languages -- which lets one create their own markup language for exchanging information in their domain

14 Appendices

14.1 Example of a Model XML File


```
<?xml version="1.0" ?>
- <COModel>
  - <CapeUnit>
      <ComponentName>Fluent CSTR</ComponentName>
      <ComponentDescription>Fluent 2D CSTR
        model</ComponentDescription>
    - <CapePort>
        <ComponentName>mass-flow-inlet-6</ComponentName>
        <ComponentDescription />
        <Type>CAPE_MATERIAL</Type>
        <Direction>CAPE_INLET</Direction>
      </CapePort>
    - <CapePort>
        <ComponentName>pressure-outlet-7</ComponentName>
        <ComponentDescription />
        <Type>CAPE_MATERIAL</Type>
        <Direction>CAPE_OUTLET</Direction>
      </CapePort>
    - <CapeParameter>
        <ComponentName>properties-
          transferred</ComponentName>
        <ComponentDescription>Type of material
          properties</ComponentDescription>
        <Mode>CAPE_INPUT</Mode>
      - <CapeParameterSpec>
          <DefaultValue>constant</DefaultValue>
          <Type>CAPE_OPTION</Type>
          <OptionList>(temperature-dependent constant
            none)</OptionList>
          <RestrictedToList>1</RestrictedToList>
        </CapeParameterSpec>
      </CapeParameter>
    - <CapeParameter>
        <ComponentName>temperature-
          minimum</ComponentName>
        <ComponentDescription>Minimum temperature (for
          temperature dependent
          properties)</ComponentDescription>
        <Mode>CAPE_INPUT</Mode>
      - <CapeParameterSpec>
          <DefaultValue>298</DefaultValue>
          <Type>CAPE_REAL</Type>
          <Dimensionality>K</Dimensionality>
          <LowerBound>273</LowerBound>
          <UpperBound>1000</UpperBound>
        </CapeParameterSpec>
      </CapeParameter>
```

- <CapeParameter>
  <ComponentName>**temperature-maximum**</ComponentName>
  <ComponentDescription>**Maximum temperature (for temperature dependent properties)**</ComponentDescription>
  <Mode>**CAPE_INPUT**</Mode>
  - <CapeParameterSpec>
    <DefaultValue>**2500**</DefaultValue>
    <Type>**CAPE_REAL**</Type>
    <Dimensionality>**K**</Dimensionality>
    <LowerBound>**1000**</LowerBound>
    <UpperBound>**5000**</UpperBound>
  </CapeParameterSpec>
  </CapeParameter>
</CapeUnit>
- <Classifier>
  <Category>**Reactors**</Category>
  <Type>**CSTR**</Type>
</Classifier>
</COModel>

14.2 Example of a pre-computed results file

Example of archived model parameters and inlet/outlet stream properties as stored in a database pre-computed results file.

```
max-iterations    mix-constant-a    mix-constant-b    temp-max    temp-min
      udf/shaft_speed
1000              4.000e+000        5.000e-001        3.000e+003  2.980e+002
      9.000e+001

mass-flow-inlet-6
pressure    temp        totalFlow   enthalpy    ACETONE   ALLYL     NPP
TRIACET   BUTANEDI      PROPYLOX METHANOL
1.034e+5    2.980e+2    1.0488e+3   -2.063e+5   3.0354e-2 4.757e-1  6.050e-9
0.000e+0  0.000e+0 0.000e+0 4.938e-1

pressure-outlet-7
pressure    temp        totalFlow   enthalpy    ACETONE   ALLYL     NPP
TRIACET   BUTANEDI      PROPYLOX METHANOL
1.033e+5    2.979e+2    1.017e+3    -2.131e+5   7.488e-14 4.765e-1  1.212e-3
1.296e-2  4.157e-3      1.208e-13 5.051e-1



max-iterations    mix-constant-a    mix-constant-b    temp-max    temp-min
      udf/shaft_speed
1000              4.000e+000        5.000e-001        3.000e+003  2.980e+002
      1.400e+002

mass-flow-inlet-6
pressure    temp        totalFlow   enthalpy    ACETONE   ALLYL     NPP
TRIACET   BUTANEDI      PROPYLOX METHANOL
1.034e+005  2.980e+002  1.0488e+3   -2.063e+5   3.035e-2  4.757e-1  6.050e-9
0.000e+0  0.000e+0      0.000e+0 4.938e-1

pressure-outlet-7
pressure    temp        totalFlow   enthalpy    ACETONE   ALLYL     NPP
TRIACET   BUTANEDI      PROPYLOX METHANOL
1.032e+005  2.979e+002  1.0170e+3   -2.130e+5   1.375e-14 4.753e-1  1.310e-3
1.405e-2  1.879e-3      1.264-13 5.074e-1
```