**DEVELOPMENT OF TECHNOLOGIES AND ANALYTICAL CAPABILITIES FOR VISION 21 ENERGY PLANTS**

**COOPERATIVE AGREEMENT NO DE-FC26-00NT40954**

QUARTERLY REPORT FOR JULY-SEPTEMBER 2001

FOR

Diane Revay Madden
***U.S. Department of Energy***
***National Energy Technology Laboratory***
P.O. Box 10940, MS 920-L
626 Cochrans Mill Road
Pittsburgh, PA 15236-0940

BY

Madhava Syamlal, Ph.D.
***Fluent Inc.***
Primary Recipient
10 Cavendish Court, Lebanon, NH 03766
Point of Contact: Kristi C. Fenner (Business and Financial)
Point of Contact: Dr. Madhava Syamlal (Technical)

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

***Aspen Technology, Inc.***
Ten Canal Park, Cambridge, Massachusetts 02141-2200

***Intergraph Corporation***
One Madison Industrial Estate, Huntsville, AL 35894

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

## TABLE OF CONTENTS



## DISCLAIMER

## 1. Executive Summary

DOE Vision 21 project requirements for the support of Global CAPE-OPEN Reaction Kinetics interfaces in Aspen Plus 12 was written (Task 2.4). The software design document was written and posted on the project web site. Intergraph started work on a proof of concept demo of the physical domain software (Task 2.6). The COM-side (Aspen Plus) and CORBA-side (Fluent) pieces of the Vision 21 controller code were written and independently verified. The two pieces of the code were then combined. Debugging of the combined code is underway (Task 2.7). Papers on fuel cell processes were read in preparation for developing an example based on a fuel cell process (Task 2.8). The INDVU code has been used to replace the boiler component in the Aspen Plus flowsheet of the RP&L power plant. The INDVU code receives information from Aspen Plus and iterates on the split backpass LTSH bypass and excess air quantities until the stipulated superheat outlet temperature is satisfied. The combined INDVU-Aspen Plus model has been run for several load conditions (Task 2.14). Work on identifying a second demonstration case involving an advanced power cycle has been started (Task 3.2). Plans for the second Advisory Board meeting in November were made (Task 5.0). Intergraph subcontract was signed and work on a physical domain software demo was started. A second teleconference with Norsk Hydro was conducted to discuss Global CAPE-OPEN standards and issues related to COM-CORBA Bridge (Task 7.0).

## 2. Technical Accomplishments

### Task 2.0 Software Integration

Task 2.4 Method for data exchange

SEZ wrote up the DOE Vision 21 project requirements for support of Global CAPE-OPEN Reaction Kinetics interfaces in Aspen Plus 12.

Task 2.5/2.6 GUI and Software Design

The Use Cases stated in the Design document were discussed. A new version of the Software Design Document (v.0.36) that incorporates feedback from project team members was issued. The current document covers the principal Use Cases, the top-level architecture and deployment of the major components of the Controller sub-system. The document is based on the Rational Unified Modeling Process.

The Use Cases stated in the Design document and User Requirements Document were re-cast in the form of PowerPoint slides to illustrate the high-level design – by showing which modules are interacting to accomplish what requirement in each use case. These slides will be used for presenting the Use Cases to end-users of the integrated software and for soliciting their feedback.

- Fluent-Aspen Plus Integration Prototype

Work was continued on the new integrated Fluent-Aspen Plus flowsheet for the production of a tri-acetone alcohol. The prototype was successfully tested with beta release of Aspen Plus 11.1. A number of sensitivity analyses using different reaction kinetics were performed to determine how the purity and yield of the tri-acetone alcohol product changes with respect to the CSTR shaft speed. SEZ also made use of the new Fluent case file (prepared by MXS) with two feed locations--one at the impeller and one at the top of the tank. A presentation at the ESCAPE 12 Conference based on this demo is being planned.

- Physical domain software

Intergraph started work on a proof of concept demo of the physical domain software. The integration of physical domain software with process domain software is not intended in this project; the demo will be used only to guide the software integration task. The scope of the demo was discussed and the necessary technical staff was identified. Intergraph has requested Alstom Power for a 3D model of a power plant for use in the demo

Task 2.7 Step One

The development plan for accomplishing this task is as follows: In an intermediate Step A, IBL will implement part of the V21 Controller that interfaces with Aspen Plus (in COM). Fluent's journal/transcript file facility will be used to test this part of the V21 Controller. The first test will be that of the Fluent-Aspen Plus prototype developed by SEZ. In an intermediate Step B, MOO and PEF will implement the part of the V21 Controller that interfaces with Fluent (in CORBA). The intermediate Step B will be tested with a test harness developed to mimic Aspen Plus. Steps A and B will proceed in parallel. After successfully testing the two steps, the codes will be merged and tested.

The intermediate Step A was completed by IBL. The C++ code to enable the V21 Controller to interface correctly to Aspen Plus and to Fluent by a file exchange mechanism was completed. Re-arranged and cleaned the C++ code to adhere to a coding standard. Redundant code was removed; specific initialization and calculations for test unit were put into a separate file. Doxygen-style documentation was added after fixing the problem with macro definitions of COM functions. A selector for Fluent version was added. It was verified that the C++ prototype gives nearly identical results to that of the Visual Basic prototype developed by SEZ. The C++ prototype and installation notes were posted on the project Website so that project team members and advisory Board members may test the prototype.

The intermediate Step B was completed by PEF and MOO. The necessary CORBA wrappers were developed. PEF developed the required Cortex APIs that effect the final transfer and exchange of data between the CORBA wrapper and Fluent 6. MOO verified the exchange of data between the CORBA wrapper and a test-harness to mimic the Vision 21 Controller functionality.

Two methods for combining the intermediate steps were discussed and a "pure Cape-Open COM-CORBA Bridge" method was adopted. The detailed integration document was prepared by MOO based on PEF's "pure bridge" design. COM automation data to C++ types and back conversion functions were created for the following automation types stored in a Variant: CapeArrayBoolean, CapeArrayLong, CapeArrayDouble, CapeArrayString, and CapeString. The functions were successfully tested and put into a static class for easy access. Put all COM CapeOpen interface definitions and classes into COM:: namespace for further integration with CapeOpen CORBA interfaces. The implementation of ICapeIntegerParameterSpec interface was created and added as a CO class to the project. The C++ project was successfully compiled and linked within the new namespace and with CORBA libraries. Resulting *.dll was successfully tested without making CORBA calls. Interface class was added successfully but IParamteter class, which is using ICapeIntegerParameterSpec interface, needs to be changed. In addition, creation of ICapeOptionSpec implementation was found necessary. MOO has completed the COM-CORBA Bridge coding and is now debugging the code with the help of IBL.

Task 2.10 Step Two

SEZ attended the AspenTech Webex Seminar on Rapid Fuel Development and read several fuel cell papers in preparation for developing a fuel cell example involving the use of Fluent CFD.

Task 2.14 Step Five

Because the Air Preheater module (company proprietary) was removed from the INDVU code, simplistic correlations for the AHGO and AHAO temperatures were inserted into the INDVU code to simulate the air preheater.

The INDVU code has been used to replace the boiler component in the Aspen Plus flowsheet of the RP&L power plant. The current implementation is based on the User2 utility in Aspen Plus. The INDVU code receives information from Aspen Plus and iterates on the split backpass LTSH bypass and excess air quantities until the stipulated superheat outlet temperature is satisfied. It operates in a post-processing mode only. (Additional details about the User2 functionality are provided in Appendix A.)

The RP&L cycle was operated over a range of discrete loads, in order to map out the impact of the bypass and excess air on the operational characteristics. Specifically, (a) the bypass was allowed to change for a specified excess air (or exit O2 concentration), (b) the excess air was allowed to change while the bypass was held constant, and (c) the excess air was allowed to change while the bypass was decreased at prescribed rates as a function of load. It appears that damper control can be exercised over the range from "maximum load" to the "control load" while excess air control can be utilized below that point. The Aspen Plus design and calculator specifications will need to be implemented in such a way as to ensure a smooth changeover. Feedback to the cycle remains to be investigated. Efforts continued to focus on implementing the Aspen Plus design and calculator specifications in such a way as to ensure a smooth transition from bypass control to excess air control when using the INDVU program.

**Task 3.0 Select Demonstration Cases**

<u>Task 3.2 Demo Case 2</u>
In an effort to identify an appropriate advanced cycle, communication has been established with an Alstom Power subsidiary, specifically the Swedish Gas Turbine Center. The center is trying to understand the Vision-21 project needs and provide a suitable cycle.

**Task 5.0 Advisory Board Activities**
Preparations were made for the second Advisory Board meeting in November during the Vision21 Program Review Meeting (Nov 6-7, Morgantown, WV).

**Task 7.0 Project Management**

WAF and MXS attended the Vision 21 Workshop for Virtual Simulation (Sept. 11-12, Pittsburgh, PA). Participants discussed enhanced visualization, how to use the visualization concepts in industry, and the need for additional funding for this type of government sponsored research.

Intergraph contract was signed and Intergraph personnel started work on the project.

MXS and PEF had a follow on teleconference with Norsk Hydro. Norsk Hydro informed that the COM-CORBA Bridge being developed at University of Achen is not yet functional. They described the different approaches that were being considered in the absence of such a bridge.

<u>Presentations</u>

- SEZ prepared an abstract entitled "Integrated Process Simulation and CFD for Improved Process Engineering" for submission to the ESCAPE 12 Conference.
- SEZ prepared an abstract for submitting to the SIMS2001 Conference http://www-pors.hit.no/tf/sims2001/sims.htm in Norway in October 2001.

**3. Issues and Resolution:**

- The completion of Task 2.7 has been delayed by about four months because of an underestimation of the effort required in establishing the COM-CORBA connection between Fluent and Aspen Plus. This is not expected to affect the schedule of the final deliverable: After establishing this critical part of the software several tasks will be done in parallel by increasing the (development) manpower applied to this project.
- Design review meeting scheduled for this quarter was postponed to October 11/12 in the aftermath of the September 11 tragedy. This postponement does not impact any project deliverable.

**4. Progress forecast for the next quarter**

- Task 2.5/2.6 GUI and Software Design
  - Conduct a design review meeting (re-scheduled for Oct 11-12 in Toronto, Canada) and revise software design document.

- Task 2.7 Step One
  - Write software development plan based on the discussions at the design review meeting.
  - Complete the COM/CORBA bridge system functionality and develop Vision21 controller version 0.1
  - Run test case1

- Task 2.10 Step Two
  - Start working on a second demonstration of the Aspen Plus – Fluent interface that includes a fuel cell.

- Task 2.14 Step Five
  - Fine-tune the linkage between the INDVU program and the Aspen Plus code using the User2 block. Attempt to provide feedback to the cycle for operating conditions below "Control Load."
  - Assess potential for utilizing the prevailing Cape Open controller interface rather than the User2 methodology.
  - Prepare outline of RP&L flowsheet demo and possibilities for INDVU code visualizations.

- Task 3.2 Selection of Demo Case 2
  - Pursue identification of advanced cycle case.

- Task 5.0 Advisory Board Activities
  - Organize second Advisory Board meeting.

- Task 7.0:
  - Prepare for the presentation and demonstration in the Vision 21 Program Review Meeting (Morgantown, WV, November 6-7, 2001)

## 5. Project Milestones

| Task Number | Milestone/Deliverables | Completion Date | | |
|---|---|---|---|---|
| | | Original | Revised | Actual |
| 1.0 | Project Management Plan | 1-30-01 | | 1-23-01 |
| 2.2 | User Requirements Document (URD) | 3-15-01 | | 3-28-01 |
| 2.3 | Software Requirements Specifications (SRS) | 4-15-01 | | 5-13-01 |
| 2.6 | Software Design Documentation | 5-15-01 | 7-15-01 | 8-10-2001 |
| 2.7 | Software Development Plan | 6-30-01 | 11-15-01 | |
| 2.7 | Working Test Case 1 | 6-30-01 | 10-30-01 | |
| 2.10 | Working Test Case 2 | 9-30-01 | 12-30-01 | |
| 2.12 | Working Test Case 3 | 1-15-02 | | |
| 2.13 | Working Test Case 4 | 3-30-02 | | |
| 2.14 | Working Test Case 5 | 1-1-02 | | |
| 2.15 | Working Test Case 6 | 6-15-02 | | |
| 2.17 | Working Test Case 7 | 9-15-02 | | |
| 2.17 | Beta version of Controller | 9-15-02 | | |
| 2.18 | User documentation for Controller | 12-30-02 | | |
| 2.20 | Integrated Software suite and demonstration | 6-30-03 | | |
| 3.1 | Demonstration Case 1 selection | 1-31-01 | 5-15-01 | 4-30-01 |
| 3.2 | Demonstration Case 2 selection | 9-30-01 | 12-30-01 | |
| 4.1 | Demonstration Case 1 simulation completed | 6-30-02 | | |
| 4.2 | Demonstration Case 2 simulation completed | 6-30-03 | | |
| 4.3 | Report on Demonstration Case simulations | 7-30-03 | | |
| 5.1 | Advisory Board Meeting | 3-31-01 | | 6-6-01 |
| 5.2 | Advisory Board Meeting | 9-30-01 | 11-7-01 | |
| 5.3 | Advisory Board Meeting | 3-31-02 | | |
| 5.4 | Advisory Board Meeting | 9-30-02 | | |
| 5.5 | Advisory Board Meeting | 3-31-03 | | |
| 5.6 | Advisory Board Meeting | 7-30-03 | | |
| 7.0 | Quarterly reports to DOE | Every quarter | | 1/30/01, 4/20/01, 7/20/01, 10/20/01 |
| 7.0 | Final project report | 12-31-03 | | |

**6. Personnel initials, List of Abbreviations and Glossary**

| Personnel Name | Affiliation | Initials |
|---|---|---|
| Woodrow Fiveland | ALSTOM Power | WAF |
| John L. Marion | ALSTOM Power | JLM |
| David G. Sloan | ALSTOM Power | DGS |
| Herb Britt | AspenTech | HB |
| Randy Field | AspenTech | RF |
| Steve Zitney | AspenTech | SEZ |
| Joe Cleetus | CERC | KJC |
| Igor Lapshin | CERC | IBL |
| Lewis Collins | Fluent | RLC |
| Paul Felix | Fluent | PEF |
| Ahmad Haidari | Fluent | AH |
| Barb Hutchings | Fluent | BJH |
| Maxwell Osawe | Fluent | MOO |
| Lanre Oshinowo | Fluent | OSH |
| Madhava Syamlal | Fluent | MXS |
| Bob Fisher | Intergraph | RJF |

| Name | Description |
|---|---|
| ActiveX | A Microsoft technology built on top of COM that extends the basic capabilities of OLE to allow components to be embedded in Web sites. |
| 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. |
| C++ | C++ programming language. |
| CERC | Concurrent Engineering Research Center, WVU. |
| CFD | Computational Fluid Dynamics. |
| CAPE-OPEN | Computer Aided Process Engineering – Open Simulation Environment Interface definitions for exchanging information with process simulation software (www.quantisci.co.uk/Cape-Open). |
| CASE | Computer Aided Software Engineering. |
| COM | Component Object Model – Refers to both a specification and implementation developed by Microsoft Corporation that provides a framework for integrating software components. |
| CORBA | The Common Object Request Broker Architecture is a specification of a standard architecture for object request brokers (ORBs). A standard architecture allows vendors to develop ORB products that support application portability and interoperability across different programming languages, hardware platforms, operating systems, and ORB implementations (www.omg.org). |
| COM-CORBA Bridge | Software for translating COM objects to CORBA objects and vice versa. |
| 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. |

| | |
|---|---|
| Doxygen | A documentation system for C++, Java, IDL (Corba/COM) and C. |
| GCO | Global CAPE-OPEN, an extension of the CAPE-OPEN project. (www.global-cape-open.org) |
| GUI | Graphical User Interface. |
| IDL | Interface definition language, which is used for defining the communications between software components linked through a middleware. |
| INDVU | ALSTOM Power in-house code for the analysis and design of the gas side of a powerplant. |
| Java | Java programming language. |
| LTSH | Low temperature super heater. |
| Middleware | Connectivity software that consists of a set of enabling services that allows multiple processes running on one or more machines to interact across a network. |
| OLE | Object Linking and Embedding.  Builds on COM to provide services such as object "linking" and "embedding" that are used in the creation of compound documents (documents generated from multiple tool sources). |
| PFD | Process Flow Diagram. |
| Python | Python programming language. |
| RP&L | Richmond Power and Light power plant. |
| SDD | Software Design Document. |
| SRD | Software Requirements Document. |
| Swing | A Java GUI tool kit. |
| UML | Unified Modeling Language. |
| URD | User Requirements Document. |
| Use Case | The specification of a sequence of actions, including variants, that a system can perform, interacting with actors (users) of the system. |
| Visual Basic | Visual Basic programming language. |
| V21 Controller | The software being developed in this project for linking CFD and other proprietary equipment-level models with process simulation models. |
| WVU | West Virginia University. |
| 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 (music, chemistry, electronics, hill-walking, finance, surfing, CFD, process simulation). |

**APPENDIX A. Summary Of User2 Approach For The INDVU Code**

Following Steve Zitney's previous work as an example, the User2 coupling consists of the following: (a) A single design spec (Broyden method) encompasses a couple of calculator blocks and the INDVU code and controls the number of times that the INDVU code iteratively executes. The design spec manipulates a PARAMETER-1 variable and samples the SH outlet temperature from the INDVU code output until that temperature is equal to the SH outlet temperature stipulated by the Aspen Plus cycle. (b) In a calculator block, the FW flow rate, FW pressure, FW temperature, SH outlet pressure, and SH outlet temperature from the cycle connectivity are equated to the appropriate elements in the User2 working array (REAL vector). All of the above variables are part of the INDVU input. (c) In another calculator block, based on the value of the PARAMETER-1 variable, values of the excess air (through an exit O2 variable) and the relative LTSH bypass quantities are determined and equated to their respective elements in the REAL array. Both of these variables are part of the INDVU input. (d) From within the User2 subroutine, a called subroutine reads and parses the INDVU input file. A data statement in the subroutine specifies the fixed format field length and location of the specific variables that will be overwritten. The subroutine then loads the appropriate elements from the REAL array into the input file, overwrites the previous values, and writes out a new input file. (e) A C++ utility (RunCmd), previously written by Fluent and declared as an external function in the User2 subroutine, spawns a process and launches the INDVU code, which reads the input data file, executes the code, and generates an output file. (f) Within the User2 subroutine, another called subroutine reads the output file, parses the lines, and extracts specific results, such as the calculated SH outlet temperature and pressure, the AHGO and AHAO temperatures, and the total boiler heat duty. These variables are equated to elements in the REAL array. The design spec utilizes the INDVU-calculated SH outlet temperature to determine, by comparison with the cycle-stipulated SH outlet temperature, if the boiler component has converged.

Remaining work items:

- Presently, the iterations (by direct substitution) on the AHGO temperature (and the AHAO temperature in tandem) are done internally by the INDVU code, although that functionality may perhaps be turned over to Aspen in a calculator block in the future.

- The interplay between the PARAMETER-1, the excess air, and the LTSH bypass, and their relative limits, remain to be further refined and validated/calibrated over the entire load range.

- Presently, the INDVU component receives information from the cycle, but does not provide coupled feedback to the cycle, as may occur below the "Control Load" point. A feedback link between the INDVU component and the cycle will need to be developed.