DEVELOPMENT AND VERIFICATION OF THE CATHENA GUI

T. Chin
Atomic Energy of Canada Ltd
Chalk River Labs, Chalk River, ON CANADA K0J 1J0
chint@aecl.ca

ABSTRACT

This paper presents the development and verification of a graphical user interface for CATHENA MOD-3.5d [1]. The thermalhydraulic computer code CATHENA has been developed to simulate the physical behaviour of the hydraulic components in nuclear reactors and experimental facilities. A representation of the facility is developed as an ASCII text file and used by CATHENA to perform the simulation. The existing method of manual generation of idealizations of a physical system for performing thermal hydraulic analysis is complex, time-consuming and prone to errors. An overview is presented of the CATHENA GUI and its depiction of a CATHENA idealization through the manipulation of a visual collection of objects. The methodologies and rigour involved in the verification of the CATHENA GUI will be discussed.

INTRODUCTION

With the advent of sufficient computational power, the complexity and detail of the thermalhydraulic problems modelled by CATHENA have risen, along with increased expectations for ease-of-use and functionality. The input files for CATHENA have increased in size and complexity and contain many internal cross-references and dependencies. Short of running the simulation, there is no simple reliable way to error check or troubleshoot these files. Input error issues can range from a simple missing quote, a spelling mistake or a cross-reference to another model that does not return the expected information.

The CATHENA input file is an ASCII text file and is organized into eight basic sections; simulation control, components and geometry, component connections, boundary conditions, system models, control models, wall models and initial conditions for the simulation. Each object in the section must be given a unique eight character label. It is possible for a single component, such as a pipe, to be referenced in all eight sections, and due to the label limitations, the ability to give the objects meaningful names can become compromised increasing the possibility for error.

The dependencies between the sections can cause additional areas for modelling problems. For example, to create a pump trip condition, a control model construct is used, and it is dependent upon a labelled pump model. The pump model is attached to a hydraulic link between two pipe components, creating a dependency for a defined connection element in the list and this connection requires the existence of its two component models (pipes, tanks, etc). The control construct for the trip may also be dependent on additional control constructs to determine the trip application during the simulation. A dependency tree showing these relationships appears in Figure 1. The possibilities for errors to be accumulated in the creation and maintenance of a large simulation file, despite the care that goes into producing it, begin to increase.

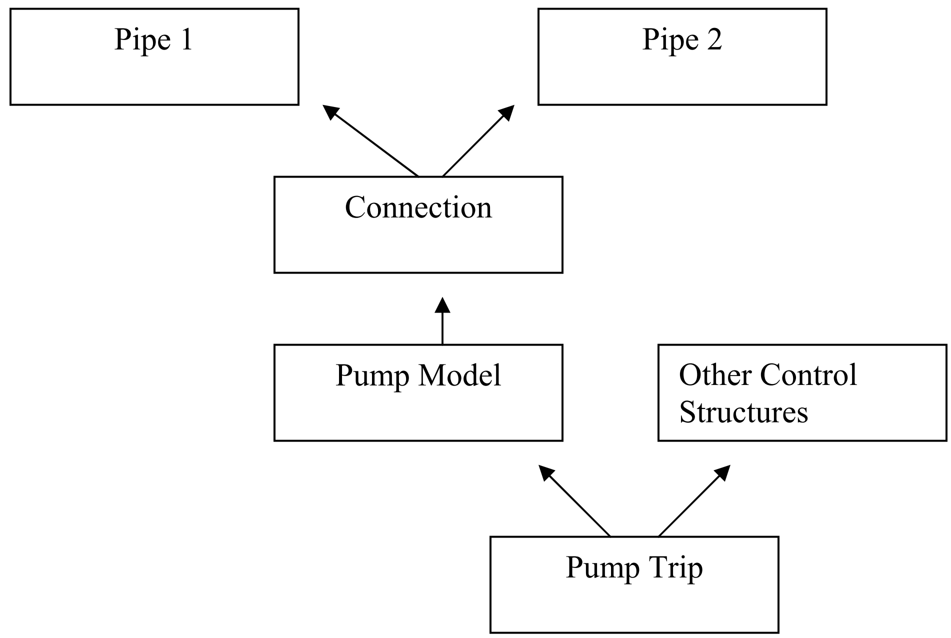

Figure 1 Dependency Tree for Pump Trip Model

This paper will discuss the basic functionality of the CATHENA GUI and how it will address these types of issues. An integral part of the GUI development is verification and testing to ensure the correctness of the program itself and for use with CATHENA.

The current development version of the CATHENA GUI will include all but 2 of the CATHENA control parameter models, the component models, the connection models, the boundary condition models, the system models with the exception of the DELAY LINE, ECI ACCUMULATOR MODEL, and REACTOR KINETICS models, all system control models with the exception of the ITERATE model, the initial conditions and the GENHTP models with the exception of the 4 Auxiliary models. At this time the ability to perform uncertainty analysis for a simulation is not part of the CATHENA GUI.

The CATHENA GUI will be available as an AECL internal release candidate in February 2009. Release to external users of CATHENA will follow. Further information about CATHENA can be found at the CATHENA website www.cathena.aecl.ca.

Product Functionality

The primary purpose of the CATHENA GUI is to assist CATHENA code users in developing and working with CATHENA input files.

The GUI is designed to run as a single-user program primarily on the Windows platform; although the code was developed in a platform-independent manner. Prompt feedback from the detection of errors or the omission of details is a design goal to aid in the development of the CATHENA idealization along with providing warnings and modelling hints when appropriate. It also has the capability of performing a comprehensive check of the entire idealization to ensure global consistency prior to the creation of the CATHENA input file.

There are three main operations for the CATHENA GUI;

- creating new idealizations in the GUI,
- importing existing idealizations for the purpose of maintenance or modification,
- and creating the ASCII files for use with CATHENA.

The idealization, within the GUI, will be created graphically through a two-dimensional drawing user interface. The model structure is built from pre-defined CATHENA components that can be selected from a list of representative icons on a toolbar. Other simulation parameters, such as simulation control information, will be set through dialogs. The operations of cut, copy, and paste are available during the model building session along with support for undo and redo operations. Multiple views of a single idealization are possible or multiple idealization files can be open at the same time.

When building or modifying idealizations the toolbar, shown in Figure 2, is used to layout new CATHENA components such as pipes, volumes, tanks, t-junctions, reservoirs and GENHTP Wall models. Once the figures have been created on the drawing window, the Connection Tool allows for creation of the hydraulic links between these components. The Hydraulic Boundary Condition Tool allows the connections between the wall models and the hydraulic components to be defined. Both of these connection tools contain logic that will only allow legal connections (according to CATHENA's simulation design rules) to be created.

The Selection Tool is used to move existing objects around on the screen. When in this "edit" mode, double-clicking on an object (component, connection or wall model) will bring up its property sheet to allow for editing of the values that will be written to the CATHENA input file.

The menu items include functions such as align, ordering of object layers on the screen, and customizations including colour of the objects. These can assist with organizing the screen layout into different sub-systems. This can be found under the Align and Attribute menu items seen in Figure 2.

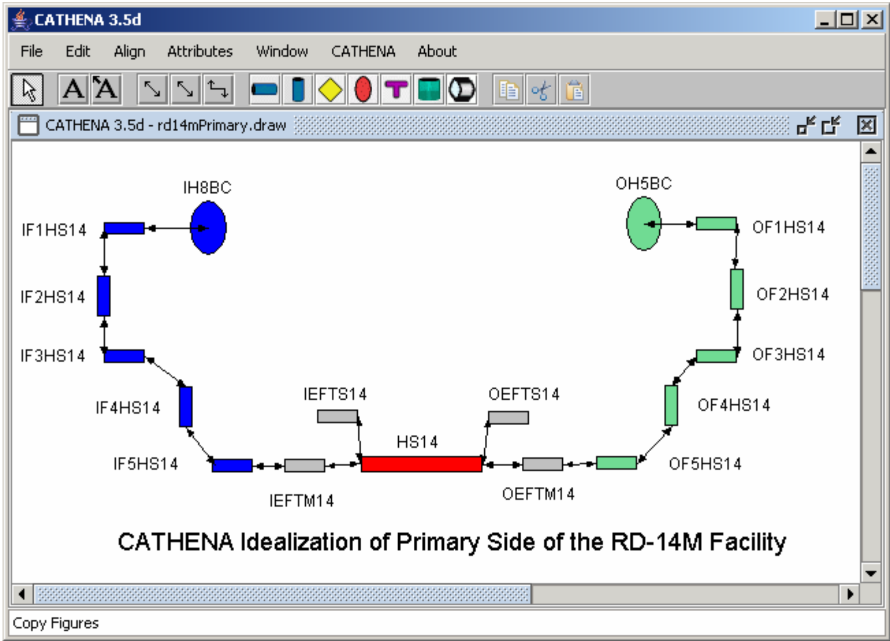

Figure 2 Main Screen of CATHENA GUI

The information for the various models will be entered within property sheets through standard user interface components such as text fields, checkboxes, and pre-populated drop-down lists. The CATHENA default values will be displayed where appropriate. Range checking for values will be enforced when possible. An example would be that a negative or zero value for a pipe diameter would not be permitted. Figure 3 contains an example of the property sheet for a pipe model.

Due to the format of the CATHENA input file, definition information for a specific model may be found in multiple sections. Figure 3 shows the tabs that are used on the property sheets to group the information for a single model. For the pipe model shown, the component and model option information is found in the components section of the CATHENA input file, the attached models tab information is found in either the boundary conditions or the system model sections and the initial condition information is found in its own section at the end of the file. For a pipe model this can include Heat Boundary Conditions or a Heat Exchanger. Information is shared across the various tabs of an object's property sheets decreasing errors when creating the cross-references between them.

CATHENA models that do not easily lend themselves to visual representations on the drawing are provided via model lists and property screens. These models include some of the System Models and all of the System Control Models. Figure 4 contains a depiction of the main system control model screen that provides access for the creation, edit or deletion operations. These screens are available from the main menu item "CATHENA" as seen in Figure 2.

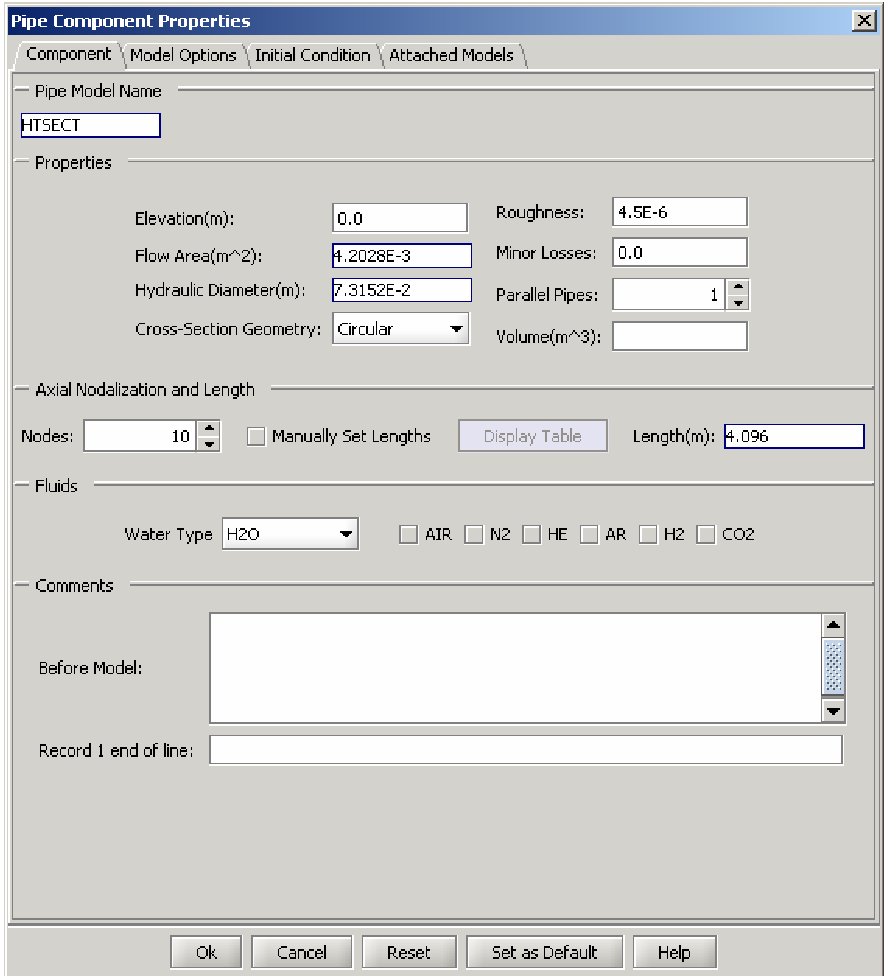

Figure 3 Pipe Property Sheet

Internal CATHENA GUI linkages are formed when cross-references are created to ensure all references remain consistent and up-to-date. However, this presents problems when the program elements are not created in order of their dependency tree. The CATHENA GUI allows for this issue by not enforcing that a model must be completely defined in order to be saved. This allows for the partial creation of the idealization in an arbitrary order. The final check for errors becomes very important to ensure that required parameters have not been left blank prior to the generation of the CATHENA input file.

The modification and maintenance of existing CATHENA input files requires importing them into the CATHENA GUI and arranging the figures on the screen. The various figures are created and laid out on the screen in a linear manner, following their order of appearance in the input file. The user can then re-arrange these objects into a representative layout of the actual hydraulic system being modelled.

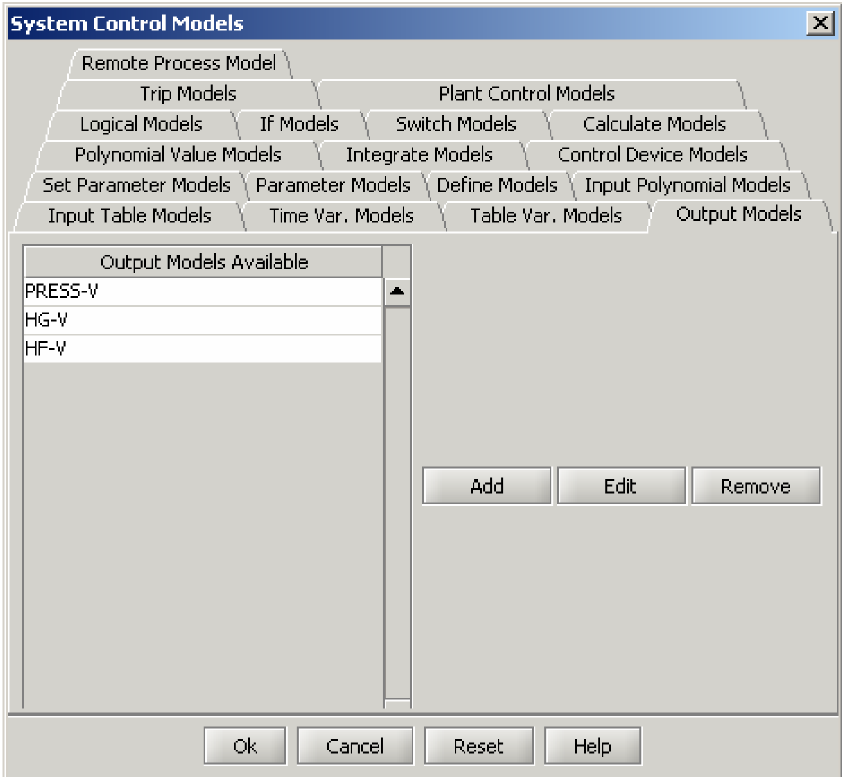

Figure 4 System Control Model Property Sheet

The CATHENA GUI does not contain an algorithm to layout the components on the screen. The reasons for this are three-fold; the CATHENA input file does not contain enough information to allow for the calculation of (X, Y) screen co-ordinates, it is an exceptionally difficult optimization problem to solve, and based on the information in the CATHENA input file there will probably not be a single "best" possible arrangement of the components. The creation of a layout in the GUI is further complicated by the cyclical nature of a CATHENA idealization.

The lack of sufficient layout information in the CATHENA input file is a function of the consequence of the CATHENA simulation requirements. The input file is an abstraction of a physical 3-D piping network. It seldom contains information about every junction and bend in the system as this increases the idealization modelling and simulation run time. The abstracted CATHENA idealization is then displayed in a 2-D layout, further limiting the usefulness of the available data with regards to layout of the various components[1].

[1] Calculating a layout dynamically as the input file is read is a problem related to the well-known Travelling Salesman problem [2] or Hamiltonian cycle in combinational optimization. It is considered to be in the mathematical domain of NP-Hard. An NP-Hard problem is a nondeterministic polynomial time problem where the solution algorithm can take many simultaneous computational paths. A brute force solution to calculating the idealization layout is (n-1)!, and no algorithm that can solve it in less than $n^2 \bullet 2^n$ time has been found. Attempting to calculate a layout becomes extremely expensive computationally (commonly solved in CPU years), and a human can provide a sufficiently useful solution in a more reasonable amount of time. The addition of one or more approximate solution algorithms for this problem will be considered for a future release of the GUI.

To ensure that the effort in creating the layout is not lost, an intermediate file format is provided to save the drawing. This file contains the information required for the CATHENA GUI to re-create the drawing. This input file contains data such as the screen coordinates of the models, and can be reloaded faster than re-parsing the original input file.

Once an idealization has been created in the CATHENA GUI, and all required modifications have been completed, the input file for CATHENA can be created. The expectation is that this will be a two-part process. The "final check" process can be run on the idealization and a list will be created of any outstanding issues ordered by model name. The issues can be dealt with, and a CATHENA input file in ASCII format can be created. This precludes the iterative debugging process of submitting the input file to CATHENA until all syntax errors have been resolved. This check process can be performed at any time during the creation/modification of the idealization.

Verification

Due to the graphical nature of the CATHENA GUI, a large portion of the testing effort was devoted to the user interface. Software industry best practices were followed and the testing effort focused on three main areas:

- the underlying code relating to the CATHENA models,
- input screen functionality,
- and user acceptance testing of the product as a whole.

Unit tests were developed for all non-graphical portions of the code, based on the automated JUnit Framework [3]. These tests were designed on a per function basis with the expectation that each individual test would either pass or fail. The framework was then used to aggregate the thousands of tests together, provide an overview of each run (show the pass/fail rate) and indicate the failed tests.

The Clover [4] code coverage tool was run in concert with the unit tests. Code coverage analysis provides a description of the lines of code exercised by the tests and the logical paths that the tests covered. Using Clover and JUnit in concert allowed for an objective analysis of the usefulness of the existing tests and suggested areas where more testing was required.

Functional testing, or User Interface testing, was performed from the perspective of the user of the application. The automated testing tool TestComplete [5] was used to create scripted tests that exercised the various elements of each application screen. Test results were logged to determine the types and reasons for test failures.

The unit tests along with the code coverage tool and the automated user interface test scripts were used to perform regression testing as development proceeded. Code that was shown to contain errors when tested was fixed before further development occurred.

Version control software is used for the development of the CATHENA GUI. A version control repository tracks all file changes by date and time, allows comments to be saved with each change and keeps tracks of the developer who made the change.

Integration Testing was performed each time the changes to the source code were added to the source code repository. The CruiseControl [6] framework for a continuous build process was used to control the integration testing. The program polls the code repository on a predefined basis waiting for a notification that a file has been changed. Once a change has been detected and the repository has had no changes for a predefined timeframe (e.g. 10 minutes) the build process will be triggered. The build process consisted of checking out all the source code from the repository into an empty directory, compilation of the source code, execution of the unit tests, creation of the executable, and running the TestComplete tests against the new executable. A web-based tool was used to track the builds and their results.

SUMMARY

The design goal for the CATHENA GUI is to assist the user of the CATHENA code by simplifying the effort to create or modify CATHENA input files, provide prompt feedback for error conditions and assist with error reduction and consistency of the models.

The current development version of the CATHENA GUI will include all but 2 of the CATHENA control parameter models, the component models, the connection models, the boundary condition models, the system models with the exception of the DELAY LINE, ECI ACCUMULATOR MODEL, and REACTOR KINETICS models, all system control models with the exception of the ITERATE model, the initial conditions and the GENHTP models with the exception of the 4 Auxiliary models. At this time the ability to perform uncertainty analysis for a simulation is not part of the CATHENA GUI.

The CATHENA GUI will be available as an AECL internal release candidate in February 2009. Release to external users of CATHENA will follow. Further information about CATHENA can be found at the CATHENA website www.cathena.aecl.ca.

REFERENCES

1. Hanna, B.N., "CATHENA: A thermalhydraulic code for CANDU analysis", Nuclear Engineering and Design, Vol. 180 pp. 113-131, 1998.

2. Applegate, D. L., et al., The travelling salesman problem: a computational study, Princeton (NJ): Princeton University Press, 2007.

3. Beck, K., Test driven development by example, Boston:Addison-Wesley, 2003.

4. Clover [database on the Internet]. San Francisco (CA): Atlassian Pty. Ltd. ©2008. Available from: http://www.atlassian.com.

5. TestComplete [database on the Internet]. Beverly (MA): Automated QA Corp. ©1999-2008. Available from: http://www.automatedqa.com.

6. Clark, M., Pragmatic project automation: how to build, deploy and monitor java applications, Raleigh (NC): Pragmatic Bookshelf, 2004.