

THE DEVELOPMENT OF A TEST PROTOCOL FOR AN ON-LINE WHOLE-BUILDING ENERGY ANALYSIS TOOL FOR HOMEOWNERS

Julia Purdy (jpurdy@nrcan.gc.ca), Phylroy Lopez (plopez@nrcan.gc.ca), Kamel Haddad (khaddad@nrcan.gc.ca), Alex Ferguson (aferguso@nrcan.gc.ca), Stephanie Mombourquette (stepmomb@nrcan.gc.ca), Paul Wyndham-Wheeler (pwheeler@nrcan.gc.ca) and Shawn Henry (shenry@nrcan.gc.ca)

CANMET Energy Technology Centre - Natural Resources Canada
Ottawa Ontario

ABSTRACT

This paper presents the testing of a dynamic on-line whole-building energy calculator. The Home Energy Analyser is intended for use by the Canadian public as a first step in assessing the energy efficiency of their houses.

The tool consists of a web-based interface for user inputs; an application to create the simulation input files from these user inputs; a simulation engine; and an application for passing the simulation engine output back to the interface and user.

A detailed test protocol, composed of three parts, was developed as part of the software development process for quality assurance purposes. These three parts include:

- testing the functionality of the user interface in passing the user inputs to the application that generates the input files for the simulation engine;
- testing the functionality of the mapping between the user interface output and the simulation engine input files; and
- testing the mapping between the simulation engine output and the web interface;

This paper will investigate the three parts of the test plan as well as the task automation tools developed to facilitate the testing. The use of these tools proved very useful given the large number of combinations of user inputs at the web interface and input files to the simulation engine. The findings show the importance of having a detailed and comprehensive test protocol during the software development phase.

INTRODUCTION

The CANMET Energy Technology Centre (CETC) develops, distributes, and supports building simulation software for the Canadian construction industry. These software tools are used to optimize the energy performance of house and building designs as well as demonstrate compliance with energy rating programs such as EnerGuide for

Houses¹ and performance-based compliance programs such as R-2000², and the Commercial Building Incentive Program³.

One of CETC's principal software tools is HOT2000 (2003), a residential energy analysis program. HOT2000 has evolved over the past 20 years by incorporating more complex and detailed calculation methods. The software has been thoroughly validated, and its user-friendly interface is designed around the needs of practicing building professionals.

The engine of our next-generation version of HOT2000 – coined ESP-r/HOT3000 (Haltrecht et al. 1997) – is based on the comprehensive and extensively validated simulation program ESP-r (ESRU 2000). It incorporates a time-step simulation, as well as many models of interest to the building industry, including a residential fuel cell model and a comprehensive air-to-air heat recovery (HRV) model.

HOT2XP (2003) is a member of the HOT2000 family of energy analysis software and serves as a quick and easy tool for analyzing energy use in residential buildings. While its graphical interface is simple enough to be used by homeowners, the underlying analysis engine is that of HOT2000.

HOT2XP is designed to speed up the task of characterizing a house by requiring only a small amount of critical information. The user may, however, edit many of the program's underlying rules

¹ An energy-efficiency rating programme that offers to help homeowners make home retrofit choices that improve the comfort and energy efficiency of their homes. Independent energy advisors visit the home to identify how the house uses energy and where it is being wasted
<http://oee.nrcan.gc.ca/houses-maisons/>

² A housing programme that encourages the building of energy-efficient houses that are both environmentally friendly and healthy to live in. <http://oee.nrcan.gc.ca/r-2000/>

³ CBIP offers a financial incentive to encourage building owners to reduce energy consumption of their buildings to 25% less than the National Energy Code for Buildings.
<http://oee.nrcan.gc.ca/newbuildings/>

and assumptions, which allows for a far greater control of the analysis than is suggested by the main interface.

For example, in HOT2XP:

- The geometry of the house is derived from only a handful of inputs, and
- The house characteristics are defaulted based on age and location - based on the values defined in Habitat Design + Consulting Ltd. SAR Engineering (1997).

The goal of this project is to develop an accurate Canadian on-line home energy evaluation tool to simulate a generalized model of the homeowner's house. The simulation is based on the house-as-a-system approach, where all the components in a house work together to form an integrated system. The performance of one component depends on its relationship with other components in the same system.

The tool under development incorporates a simple graphical user interface with the HOT2XP default house archetypes and the ESP-r/HOT3000 simulation engine. As the tool has not yet been released to the public, the URL address cannot be supplied to the reader.

Results and potential energy savings are presented to the user in graphical format and are determined by the input-information provided by the user, weather data for their city or region, and information about local construction standards and styles. The website also suggests generalized cost effective renovation information.

A detailed investigation of the tools currently available on-line is given in LBNL (2002). They evaluated 50 web-based residential calculators, of which 21 were considered to be whole-house tools. Of these,

- 13 provided energy calculations,
- 5 consolidated the results with actual energy bills, and
- 3 provided both options.

Purdy (2004) describes how the findings of several survey papers – including LBNL (2002) and SAIC (2001) – were used in the development of the Home Energy Analyser.

Validation of Simulation Models

While the test protocol does not directly address the accuracy of the simulation models within the simulation engine, the simulation environment used in this development project has been validated.

HOT2000 has been compared against other detailed simulation programs (Haltrecht and Fraser 1997) using the Home Energy Rating System (HERS)

BESTEST (Judkoff and Neymark, 1995a and b).

In addition, several aspects of the ESP-r/HOT3000 simulation environment have been validated using BESTEST tools – including the building envelope (Haddad and Beausoleil-Morrison 2001) and the following HVAC systems: the air-conditioning model (Haddad 2004a and b), the radiant heating and cooling system models (Haddad 2002), and fuel-fired furnace models (Purdy and Haddad 2002).

The following sections detail the structure of the software, the three parts of the test plan, as well as the automated testing tools developed.

STRUCTURE OF THE SOFTWARE

The software is composed of two main components: the simulation engine and the web-based interface (Figure 1). The simulation engine performs all the calculations to simulate the behaviour of the house and its HVAC system and the interface provides the user with a convenient method to enter data to describe their house.

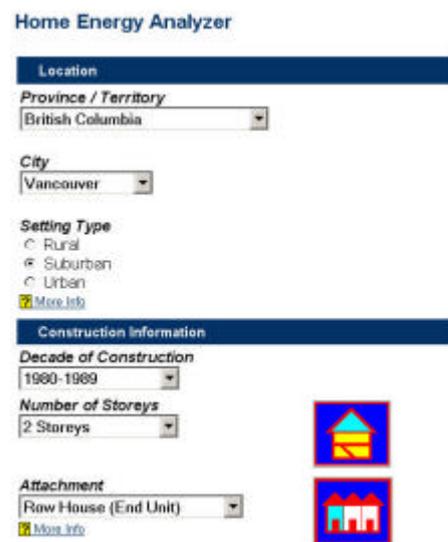


Figure 1: A Section of the Home Energy Analyzer User Interface

The on-line user of this calculation tool passes through a number of successive inputs to define all of the information required to complete the simulation of their home. The user interface communicates with the simulation engine through a series of ASCII text files - called the *simulator input files*. These files are created from the user-input data and from default data contained within databases.

The ESP-r/HOT3000 simulation engine reads the building and HVAC description from these ASCII text files and then performs the calculations to simulate the house and its HVAC system.

The simulation results are written to an ASCII text file, called the *simulator results file*. Once the

simulation is complete, the interface reads the results contained in this file and displays the results to the user (Figure 2).

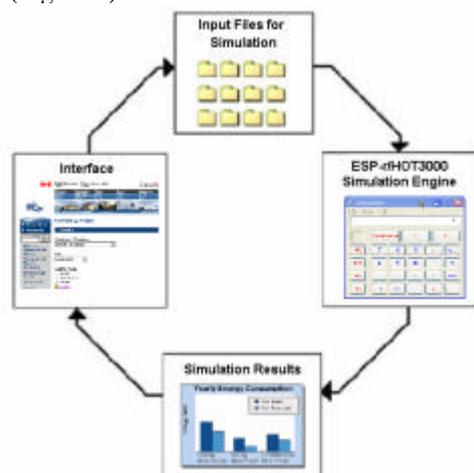


Figure 2: Data Flow from the Interface to the Simulation Engine and Back to the Interface

Calculated Results

Based on the input information – both the user inputs and the defaulted housing information – an annual simulation is performed. The calculated results of the simulation are presented to the user in graphical format (Figure 3).

Home Energy Analyzer

Thank you for using our Energy Analyzer to get a picture of how a home like yours uses energy.

In Canada, much of the energy we use to run our homes comes from the burning of fossil fuels such as coal, oil and gas. This creates greenhouse gases that contribute to climate change. So saving energy not only makes sense for your family's budget, it also helps our environment.

Results

The way your house presently uses energy is illustrated by the column on the left of each of the graphs below.

EnerGuide for Houses data for your region and house type suggests that, if upgraded, your home has the potential to use energy as illustrated by the column on the right of each of the graphs below.

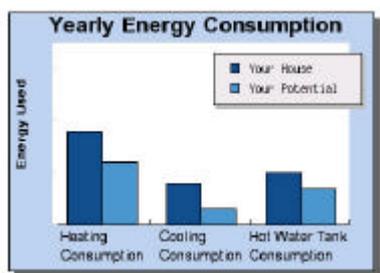


Figure 3: Results Output to the User

The simulator results file are brought into the interface and the user is provided with a graphical analysis of their current energy consumption as well as a comparison to a 'typical' energy efficient house.

These comparison values were derived from HOT2XP simulations based on a standard set of upgrades. These simulations were run for each

location and age of construction.

Additional information on the structure of the software - including the user inputs, the streaming application and the results output – is available from Mombourquette and Wyndham-Wheeler (2004).

SIMULATOR INPUT FILES AND THE STREAMING APPLICATION

The ESP-r/HOT3000 calculation engine requires a set of ASCII text files – called the *simulator input files* – that define the system to be simulated. The facility designed to create these simulator inputs files is called the *streaming application* and is described here.

The web interface PHP Hypertext Pre-processor (PHP) script writes the user inputs to an ASCII file and launches the streaming application, passing it the name of the input file as a command line parameter.

The ESP-r/HOT3000 streaming application, reads this file, line by line. The format of each line of input is *token + value*. In this way, the input data structure is flexible to the addition of future data items without disrupting the parsing of the input data stream.

All independent input data items are mapped immediately to the ESP-r/HOT3000 data structure. If an input data item is dependent on another input data item, then its mapping is left until the input data stream parsing is completed.

The ESP-r/HOT3000 data-mapping layer expands an input value to its corresponding ESP-r/HOT3000 data structure. This usually results in one input value generating multiple data entries in the ESP-r/HOT3000 data structure. For example, many simulation input file entries are dependent on the age of construction.

The data-mapping layer is structured as a logical progression of defining the building and its environment. The building's origin and dimensions are mapped first, then each zone with all its surfaces are defined and mapped – including the mapping of material properties, controls, and loads, followed by the HVAC and DHW mapping.

Once the mapping is complete, the streaming application writes all ESP-r/HOT3000 data to the corresponding data files in the typical ESP-r/HOT3000 data structure.

TESTING THE GRAPHICAL USER INTERFACE (GUI)

The graphical user interface (GUI) is designed to aid the user in inputting their house data. There are a number of dependencies inherent in this data input, and as such the GUI has a large amount of logic built into it. Many of the possible user selections available

in the interface are based on inputs already provided by the user, i.e., form fields are greyed-out or selection lists added or removed.

For example, if the user selects their province as “British Columbia” the list of cities changes, and only cities in that province are displayed in the list. This prevents the user from being able to select a combination of inputs that are not logically possible, e.g. Ottawa, British Columbia.

The testing of the GUI was twofold: testing the input dependencies, as well as testing the streaming of these inputs to the simulation engine.

By having input values dependent on previously selected inputs, the testing of the interface was performed by doing visual checks – by selecting each possible input, for example selecting each province, then ensuring that the dependent inputs (in this case, “City”) changes to allow only the possible values.

Because of the inherent input dependencies, the user-input data is also validated before it is sent to the mapping application. This eliminates the possibility of the user making errors in their selections and guides them through the difficult process of answering questions relating to their home’s thermal description.

Once it was ensured that all dependent inputs were being correctly displayed, the streaming of these inputs – the so-called tokens – to the simulation engine was confirmed. The token values were first streamed to an output file, and visually examined it to ensure all the tokens and entered values were being passed correctly. An additional check on the token values was that the tokens streamed from the interface had to match the input format required by the simulation engine, or the simulation would not initiate.

TESTING THE MAPPING OF SIMULATOR INPUT FILES

The user interface sends a HTTP request with all of the selected inputs to the streaming application that create the required ESP-r/HOT3000 simulation engine input files.

The creation of these input files were tested using three different techniques, including:

- Testing against the input files for standard archetype models,
- Testing against the streaming application specification, as well as
- Simulating an end-to-end run by generating a random, or partially random, set of input files.

The latter two testing approaches were enabled with the use of automated procedures.

Archetype Testing

A set of 45 house models – the so-called archetypes – were created using the ESP-r/HOT3000 interface for the purpose of testing. These 45 house models incorporated each of the systems available to the user and represented each geographical region across the country.

The specification for the creation of these ESP-r/HOT3000 models was then used to create 45 streaming application input files. These ASCII files defined the tokens + values as they would have been selected by the user.

The inputs available to the user include:

Basic Housing Type

- Size/Area
- Number of Floors/Storeys
- Attachment
- Location
- Age of House

Foundation type (basement, crawl space, walkout/half-basement or no basement)

Heating Equipment

- Type
- Fuel Type

Air-Conditioner Properties

Hot Water Heater

- Number and Volume of Tanks
- Fuel Type

Day and Night Thermostat Setpoints

Number of House Occupants

The simulator input files created by the streaming application were manually compared to those created for the archetypes in the ESP-r/HOT3000 interface.

This testing procedure was extremely useful as the first testing exercise of the streaming application. The manual comparison of the input files is time-consuming, so it was only used as an initial screen. To test a wider array of inputs, automated testing procedures were adopted.

HVAC Mapping

As defined in the previous section, the interface requires the user to select the type and age of heating system installed in their house. The choices are: furnace, boiler, wood-stove, electric baseboard, air-source heat pump, ground-source heat pump, and electric radiant floor heating. The user also has the option of detailing the existence of an air-conditioner.

All the information related to the HVAC system being modeled is contained within the HVAC *simulator input file* – an ASCII file that is an input to the ESP-r/HOT3000 engine. The construction of this file is based on user input information from the interface and appropriate defaults for each Canadian province.

The capacity of the heating and cooling equipment is determined based on a sizing procedure dependent on: the house construction, weather information for the house location, and the user-defined heating and cooling control set points. This capacity is a required input to the standard ESP-r/HOT3000 HVAC and control input files.

The calculated capacity of the heating and cooling equipment was manually compared with hand-calculations and the HVAC inputs to the simulator input files were also manually checked before the code was incorporated.

Testing Against the Streaming Application Specification

Sources of error in the simulation input files produced by the streaming application include: errors in the streaming application specification and errors in the implementation of the streaming applications. The streaming application output was thoroughly tested against the specification to ensure that it was properly implemented. Upon completion of this testing, the remaining errors in the simulation input files and the results they produced could be attributed to discrepancies in the streaming application.

Manually performing detailed comparisons between the simulation input files and the streaming specification would be a monumental task. Each ESP-r/HOT3000 model produced by the streaming applications was comprised of as many as 17 files, totaling 800 lines. Manual testing of the streaming applications would thus require the authors to compare 36000 lines with the streaming specification. Furthermore, these tests would have to be revisited with each set of revisions to the streaming applications, ensuring the required changes were properly implemented. Finally, manual testing of the streaming application is problematic as it risks introducing additional error into the test results; a tester might easily overlook a disparity between the specification and the simulation input files, or misinterpret the specification.

Instead, a method was devised to extract data directly from the streaming specification and, using these data, a second set of simulation input files suitable for comparison with those produced by the streaming application was built. The streaming specification was comprised of a set of spreadsheets describing in detail how each record in the simulation input files was affected by the various options selected by the user. Each of these rules contained a tag, a set of conditions, and a value. The tag identified the record within the appropriate input file that the rule governed. The value described the data that the record should be set to whenever the set of conditions were true. This set of rules was exported from the

spreadsheets in comma separated value (csv) format.

A script was written to parse these csv files, and replace tags in a set of template files using values appropriate given the options chosen by the user. The resulting files were then compared to those produced by the streaming applications using standard file comparison software.

The advantages of this approach are twofold. The streaming applications were tested directly against the specification as opposed to a tester's interpretation of the specification, thus reducing the error in the test results. Additionally, this method lends itself to automation, permitting very rapid development and bug-fix cycles.

Whole System Testing Using Random Input Tokens

In order to compare a wider range of input values as well as a combination of inputs, a random input generator was created. This automated tool randomly selects from the available user inputs and creates the simulator input files using the streaming application.

These simulations were run using the ESP-r/HOT3000 engine. The automated tester determines if the run was successful or not. If the simulation was not successful, the problematic simulator input file can be tested using the automated specification tester and any differences in the input files can be resolved.

TESTING THE RESULTS MAPPING

Once the simulation is complete, the website script extracts the heating, cooling, and domestic hot water consumption information for each simulation. This information was passed to the output results page where they are formatted into a PHP graph for the user.

To ensure the produced simulation results are streamed to the output results page properly, a manual testing strategy was adopted. Before the result values are converted to a graph, they are printed to the screen and manually compared with the original simulation.

AUTOMATED TESTING TOOLS

Two automated tools were developed to test the application – a tool to validate the streaming applications and a tool to test the system as a whole based on a set of randomly generated inputs. These two automated tools are described in the following sections.

Streaming Application Validation

As discussed above, validation of the streaming application was accomplished using a script capable of parsing the csv files describing the specification and producing a set of simulation files suitable for

comparison with those output by the streaming applications. This process was further automated by revising the script to:

- read the configurations of the 45 archetypes,
- build simulation input files for each archetype using the csv files,
- call the streaming applications to produce a similar set of files, and
- compare the results produced from the csv files and the streaming applications.

With these modifications, the script permitted very rapid testing cycles (the streaming applications could be exercised through all 45 archetypes in less than five minutes), thus greatly reducing the time required to develop and debug the streaming applications, and greatly increasing confidence in their accuracy.

Whole System Validation

An automated script was developed to test the system as a whole – from user inputs to results generation – as well as to narrow problems to specific units.

The script was written in PHP to interface with existing PHP components and because of PHP's inherent HTTP and text-processing abilities.

The script simulates an end-to-end run by generating a random, or partially random, HTTP request to the server hosting the simulator. At the intersection of each component, the automated testing script copies each component's output and logging files (if they exist). At the end of each run, the scripts determines if the run was successful and organizes the results for further investigation by a developer.

The script generates its request in one of three ways. First, the script can read a token file in the same format as used by the mapping component. This is useful because the file is both human-readable and can be used to regenerate known failure cases.

Alternatively, the testing script can generate request for every possible permutation within pre-defined bounds. This approach is useful for testing within a narrow set of possibilities that are likely to cause problems, such as at the limits of input values. Of course, it's not feasible to test every possible permutation of input since the number of possible combination would run in to the billions.

As a final method, the script can generate random requests. These requests are check for legality based on the constraints defined by the interface and then made to the server. The method is useful to find new problem areas.

For the testing of the system, all three methods of testing were implemented. The random generation of input scenarios and the limits testing were both used to discover errors. Once the error was fixed in the

code, the token file that was used to discover the error is run through the script again to ensure that the error was successfully fixed.

CONCLUSIONS

This paper presents the testing of a dynamic on-line whole-building energy calculator. The tool consists of a web-based interface for user inputs; an application to create the simulation input files from these user inputs; a simulation engine; and an application for passing the simulation engine output back to the interface and user.

Three parts of the test plan were presented: the testing of the graphical user interface (GUI); the testing of the mapping of the *simulator input files*, by comparing the simulator input files to a set of archetype models, testing against the streaming application specification, as well as comparing the *simulator input files* for a series of randomly generated inputs; and testing the mapping of the results from the *simulator results files* to the user-interface.

In addition, two automated testing tools were described. The use of these tools proved very useful given the large number of combinations of user inputs at the web interface and input files to the simulation engine.

REFERENCES

- CMHC (1992), *Environmental Impact Study: Phase 1 –Development of a Database on Housing Characteristics Representative of the Canadian Housing Stock. Final Report*, Canada Mortgage and Housing Corporation Report No. Cal MH110 92E53, Ottawa Canada.
- Habitat Design + Consulting Ltd. SAR Engineering (1997), *Final Report: Defining Selected Technical Defaults for the Canadian Home Energy Efficiency Rating System*, Prepared for Buildings Group CETC, Natural Resources Canada.
- Haddad K. (2002), *Application of RADTEST to ESP-r/H3K*, Modeler's Report included in the International Energy Agency RADTEST - Radiant Heating and Cooling Test Cases.
- Haddad K. (2004a), *Application of the IEA HVAC BESTEST Suite of Test Cases for the Validation of an Air-Conditioning Model*, A paper submitted to *eSim 2004*, IBPSA-Canada Conference, Vancouver Canada.
- Haddad K. (2004b), *An Air-Conditioning Model Validation and Implementation into a Building Energy Analysis Software*, Accepted for publication in the ASHRAE Transactions, ASHRAE Annual Meeting.

- Haddad K. and Beausoleil-Morrison I. (2001), *Results of the HERS BESTEST on a Building Energy Simulation Computer Program*, ASHRAE Transactions, ASHRAE Annual Meeting.
- Haltrecht D., Zmeureanu R., and Beausoleil-Morrison I. (1999), 'Defining the Methodology for the Next-Generation HOT2000 Simulator', Proc. *Building Simulation '99*, (1) 61-68, International Building Performance Simulation Association, Kyoto Japan.
- Haltrecht D. and Fraser K, "*Validation of HOT2000 using HERS BESTEST*", Proc. Building Simulation '97, Fifth International IBPSA Conference, pp. 63-69, 1997.
- HOT2000: A Comprehensive Tool for the Design of Energy Efficient Homes and Buildings, Version 9.1, Natural Resources Canada, 2003. www.buildingsgroup.nrcan.gc.ca/software/hot2000_e.html
- HOT2XP Version 2.5, Natural Resources Canada, 2003. www.buildingsgroup.nrcan.gc.ca/software/hot2xp_e.html
- LBNL (2002), 'Review & Comparison of Web- and Disk-based Tools for Residential Energy Analysis', Lawrence Berkeley National Laboratory Report No. 50950, Berkeley USA.
- Judkoff, R and Neymark, J. (1995a). Home Energy Rating System: Building Energy Simulation Test (HERS BESTEST). User's Manual, Volume 1, Tier 1 and Tier 2 Tests.
- Judkoff, R and Neymark, J. (1995b), Home Energy Rating System: Building Energy Simulation Test (HERS BESTEST). Reference Results, Volume 2, Tier 1 and Tier 2 Tests.
- Mombourquette S. and Wyndham-Wheeler P. (2004), 'Structure of Mapping Methods Used to Connect ESP-r to the Web and its Increased Usability Through a New Interface', A paper submitted to *eSim 2004*, IBPSA-Canada Conference, Vancouver Canada.
- Purdy J. and Haddad K. (2002), *The Development and Validation of a Furnace Model for ESP-r/HOT3000*, Proceedings of eSim 2002, IBPSA-Canada Conference, Montreal Canada.
- Purdy J. (2004), 'The Development of a Dynamic On-Line Whole-Building Energy Analysis Tool for Homeowners', A paper submitted to *eSim 2004*, IBPSA-Canada Conference, Vancouver Canada.
- SAIC (2001), 'Home Energy Analysis Software Study: Final Report', Prepared for California Energy Commission, Sacramento USA.

