SimBuild 2006
Second National IBPSA-USA Conference
Cambridge, MA
August 2-4, 2006

A LIBRARY OF HVAC COMPONENT MODELS FOR USE IN AUTOMATED DIAGNOSTICS

Peng Xu, Philip Haves, Dimitri Curtil
Lawrence Berkeley National Laboratory

ABSTRACT

The paper describes and documents a library of equipment reference models developed for automated fault detection and diagnosis of secondary HVAC system (air handling units and air distribution systems). The models are used to predict the performance that would be expected in the absence of faults. The paper includes a description of the use of automatic documentation methods in the library.

INTRODUCTION

The increasing complexity of building HVAC control and management systems heightens the need for the devolvement of tools to assist in monitoring the performance of these systems. The application of these tools is expected to lead to improved comfort, energy performance and reduced maintenance costs. The function of these tools may be limited to collecting raw data from sensors and control system outputs and displaying it for manual analysis by operators or engineers. Alternatively, the tools may analyze the data in order to determine whether the operation is correct or faulty (automated fault *detection*) and may also identify the location or nature of the physical cause of a problem (automated fault *diagnosis*).

The work reported here is part of an ongoing effort to develop model-based automated fault detection and diagnosis methods and tools for HVAC systems. The reference model is used to predict performance that would be expected in the absence of faults. A comparator is used to determine the significance of any differences between the predicted and measured performance and hence the level of confidence that a fault has been detected.

The model-based approach facilitates a building life cycle approach. In functional testing as part of initial commissioning, a baseline model of correct operations is first configured and calibrated against design information and manufacturers' data. Next, the model is fined-tuned to match the actual performance after any faults have been fixed and the model is then used as part of a performance monitoring diagnostic tool for operations.

The paper describes the library of equipment models, including cooling coil, control valve, mixing box and fan system models. The models are simple enough so that their parameters can be determined from information usually available to commissioning agents or building operators. Simple preprocessors were developed to calculate the model parameters from this information.

A method of generating model documentation automatically from meta-information embedded in comments in the source code is described.

REFERENCE MODELS

Simulation Software: SPARK

The simulation program SPARK (SPARK 2003) has been used to develop and implement the reference model library. SPARK is an object-based software system that can be used to simulate physical systems that can be modeled using sets of differential and algebraic equations. 'Object-based' means that components and subsystems are modeled as objects that can be interconnected to form a model of the entire system. SPARK differs from earlier object-based simulation programs such as HVACSIM+ (Clark 1985) and TRNSYS (Klein *et al.* 1976) in that the granularity extends down to the equation level, allowing the user to change which variables are inputs and which are outputs without changing the models and facilitating the use of graph-theoretic methods to improve simulation speed and robustness.

SPARK models have a hierarchical structure. The smallest programming element is a class consisting of an individual equation, called an atomic class. A macro class consists of several atomic classes (and possibly other macro classes) combined together into a higher level unit.

The process of describing a system in order to produce a SPARK system model begins by determining which components can be represented using existing models. New models can be created relatively easily from the atomic classes in the SPARK library that embody commonly used thermofluid relationships. Since there may be several components of the same type in a system, SPARK object models, i.e., equations or groups of equations, are defined in a generic manner, called classes. Classes serve as templates for creating any number of like objects that may be needed in a problem. The problem model is then completed by linking objects together, thus defining how they interact, specifying parameter values that specialize the system model to represent the actual problem to be solved and providing boundary values.

Cooling Coil

The challenges in cooling coil modeling are to treat the variation in surface resistance with flow rate and to treat partially wet operation. The most common fault in both heating or cooling coils is fouling of the heat exchange surface, either on the air or the water side. In order to detect loss of peak load capacity due to fouling when it occurs, it is only necessary to model full load operation. However, in order to be able to predict loss of capacity before it occurs, it is necessary to model part load operation as well. Another significant fault in coil subsystems is control valve leakage. Correct operation of the coil subsystem is modeled by combining the coil model described in this section with the valve model described in the next section.

A significant number of coil models have been developed over the last few decades; none of the models that treat partly wet operation is entirely suitable for fault detection. In particular, there are two cooling coil models in the ASHRAE Secondary Toolkit (Brandemuehl *et al*., 1993). The simple model approximates partially wet operation as all wet or all dry. The detailed model treats the dry and wet regions separately and iterates to find the position of the boundary. Testing of this model performed as part of the work described here showed that the iterative scheme employed in the model sometimes fails to converge under conditions of high humidity. Recently, a dynamic cooling cool model has been developed (Zhou, 2005). The model can predict the dynamic performance of cooling coil within a reasonable range of accuracy. However, the model is too detailed and complicated for it to be used on site for fault diagnosis. For this reason, it was decided to develop a new model of partially wet coil operation.

In the new model, the coil is divided into discrete sections along the direction of fluid flow, as shown in Figure 1. In each section, heat and mass balance equations are established for each fluid, together with rate equations describing the heat and mass transfer. If the dew point temperature of the air is lower than the metal surface temperature, that section of the coil is treated as dry. If not, the water condensation rate is assumed to be proportional to the difference between the humidity ratio of the bulk air stream and the humidity ratio of saturated air at the temperature of the coil metal surface. The coefficient of proportionality is determined by assuming the value of the Lewis Number is unity. The sections that make up the coil are linked together by associating the fluid inlet conditions of one section with the outlet conditions for the adjacent upstream section. The resulting set of coupled equations is then solved by SPARK. Although the computational burden of the new coil model is significantly greater than that of the ASHRAE Toolkit models, the model is more robust, and it has the additional advantage of being a suitable starting point for a dynamic cooling coil model.

SPARK cannot simulate a model with a dynamically varying number of objects so the number of layers of the coil is hard-wired to 20 instead of being variable. Dividing the cooling coil into 20 layers provides sufficient accuracy because, under the common operation range of the cool coil, the driving temperature difference in each layer is one order of magnitude lower than the temperature change along the flow direction. The thermal conductances of the external and internal surfaces are modeled using the turbulent flow approximation:

$$UA_{ext} = C_{ext} \cdot v_{air}^{0.8}$$

where v is the fluid velocity.

$$UA_{\text{int}} = C_{\text{int}} \cdot v_{liq}^{0.8}$$

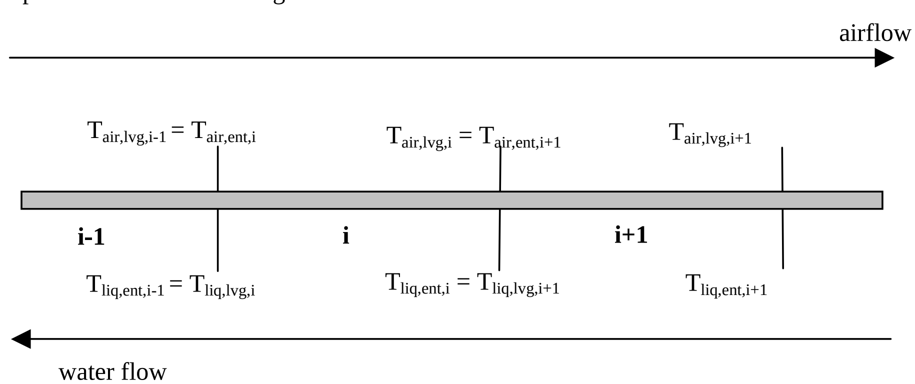


Figure 1: Discrete sections in the cooling coil model

For each layer of the cooling coil, the sensible and latent heat transfer rates between the air and cooling coil surface are given by:

$$q_{sen,layer} = (\frac{T_{air,ent} + T_{air,lvg}}{2} - T_{sur})UA_{ext}$$

$$q_{lat,layer} = \max\left\{0, (\frac{w_{air,ent} + w_{air,lvg}}{2} - w_{sur})h_{fg} \cdot h_{mass}\right\}$$

where T is the temperature, w is the humidity ratio, h_{fg} is the latent heat of vaporization of water and h_{mass} is the mass transfer coefficient.

Heat transfer between the cooling coil surface and the water is given by:

$$q_{tot,layer} = (T_{sur} - \frac{T_{liq,ent} + T_{liq,lvg}}{2})UA_{\text{int}}$$

Enthalpy and sensible heat balances yield:

$$\begin{aligned} q_{tot,layer} &= q_{sen,layer} + q_{lat,layer} \\ &= m_{liq} c_{liq} (T_{liq,lvg} - T_{liq,ent}) \\ &= m_{air,dry} (h_{air,ent} - h_{air,lvg}) \end{aligned}$$

$$q_{sen,layer} = m_{air,dry} c_p (T_{air,ent} - T_{air,lvg})$$

where h the specific enthalpy.

Information for different operating points is required to estimate the coefficients C_{int} and C_{ext}. Typically, only performance information at design conditions is available to commissioning agents and building operators; this is sufficient to estimate the total UA but not the air-side and water-side components. However, empirical correlations developed by Holmes (1982) can be used to estimate the ratio of the air-side and water-side conductances at specified conditions and hence determine the coefficients C_{int} and C_{ext} when combined with the design condition information usually provided on the mechanical drawings.

Control valve

A control valve varies the fluid flow rate in a circuit by varying its flow resistance. An external actuator is used to move a plug connected to the valve stem that restricts the flow to varying degrees depending on its position. There are three distinct valve flow types based on the geometry of the plug: quick opening, linear, and equal percentage. The equal percentage characteristic is used to compensate for the non-linear characteristic of heating and cooling coils and the effect of the series resistance of the coil.

The most common faults associated with control valves are: leakage, stuck valve/actuator, actuator/valve range mismatch and unstable control. In order to detect these faults, it is more important to model the valve behavior at each end of the operation than in the middle. However, as discussed in the Coil section, it is desirable to be able to predict the part load performance of coils in order to anticipate loss of peak capacity before it occurs. Since the water flow rate through a coil is not generally measured in HVAC systems, it is necessary to treat the behavior of the control valve at intermediate flow rates by modeling its inherent and installed characteristics in order to predict the water flow rate through the coil.

The water flow rate is a function of the valve position, the flow rate through the valve when fully open and the leakage. The flow characteristic is assumed to be parabolic (see below), which is an adequate and convenient approximation to the equal percentage characteristic.

In order to model the installed characteristic of the valve, it is necessary to treat the effect of the series resistance of the coil and other components in the branch. This is conventionally expressed in terms of the authority of the valve. Authority is the ratio of the pressure drop across the valve when it is fully open to the pressure drop across the whole of the branch when the valve is fully open. When the authority is equal to unity, the pressure drop across the valve dominates the pressure drop in the branch and there is no distortion of the valve flow characteristic. When the authority is equal to zero, the pressure drop across the valve is negligible unless it is fully closed and so the valve has essentially no effect on the flow rate except when it is fully closed. A more detailed description of valve authority is given in ASHRAE (2004). The fractional leakage is given by

$$l_{inher} = \frac{\dot{m}_{leak}}{\dot{m}_{open}}$$

where $\dot{m}$ is the mass flow rate through the control port at constant pressure drop.

The inherent flow characteristic is then

$$f_{inher} = l_{inher} + (1 - l_{inher}) \cdot s^2$$

where f_{inher} is the ratio of the flow rate at stem position s to the flow rate when the valve is fully open when the pressure drop across the valve is constant. (s is in the range 0-1.)

The installed flow characteristic is then

$$f_{install} = \frac{1}{\sqrt{\frac{A}{f_{inher}^2} + (1 - A)}}$$

where A is the authority of the valve.

In general, the leakage of a new control valve is negligibly small; however, there may be situations where it is necessary to model a significant amount of leakage, e.g. when monitoring an existing valve to determine whether the leakage gets worse over time.

Fan

The most commonly used fans in large air handling units are centrifugal fans. In VAV systems, fan capacity is controlled by varying either the rotation speed or, in older installations, the position of an inlet guide vane. There is a pressure sensor in the supply duct and a feedback control loop to maintain the air pressure in the duct constant by adjusting the supply fan capacity. The return fan capacity is varied in order to control the pressure of the part of the building served by the particular air handling unit (AHU). The model described here applies to VAV systems in which the capacity of each fan is controlled by varying the rotation speed. As with all the models presented here, only the behavior of the mechanical equipment is treated in the model. The most common faults associated with fans are: variable speed drive malfunction, incorrect rotation direction, and pressure sensor offset.

The model treats either the supply fan or the return fan, together with the appropriate section of the distribution system, as shown in Figure 2. The flow rate in a fan system is determined by the characteristics of the fan and the characteristics of the system, as shown in Figure 3. Fan performance is modeled by using the fan similarity laws to normalize the flow rate, pressure rise and power in terms of rotation speed and diameter. Over the limited range of normalized flow rate used in normal operation, the relationship between fan pressure rise and flow rate at constant speed (the 'head curve') can be approximated using a constant term and a squared term:

$$\Delta p_{fan} = k_{fan0} n^2 - C_{fan} \cdot m|m|$$

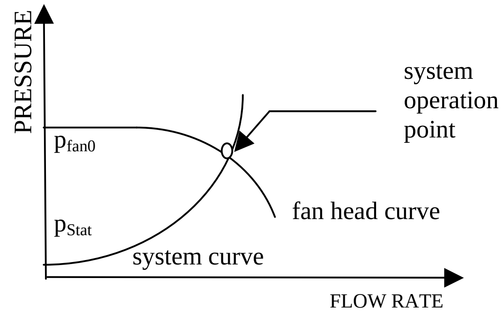


Figure 3 Schematic of the fan model

(Note that the range of normalized flow rates encountered in VAV systems is much less than the range of un-normalized flow rates.) The constant term is the pressure rise extrapolated to zero flow, which is proportional to the square of the rotation speed, n, and the squared term corresponds to the internal pressure drop inside the fan. ($m|m|$ is used rather than m^2 to yield the correct behavior at negative flows.) The model is written in terms of total pressure (i.e. static pressure plus velocity pressure) since the energy losses are directly related to changes in total pressure.

The system curve, which represents the pressure drop through all the air handling unit and distribution system components, also consists of a constant term and a squared term. For the supply fan subsystem, the constant term is the static pressure set-point. The squared term represents the pressure drop through the AHU and distribution system components and the velocity pressure at the static pressure sensor, both of which are proportional to the square of the air mass flow rate.
where ρ_{air} is the density of air and A is the cross sectional area of the duct at the pressure sensor.

$$\Delta p_{sys} = p_{stat} + \left(\frac{1}{2\rho_{air} A^2} + C_{sys}\right) \cdot m|m|$$

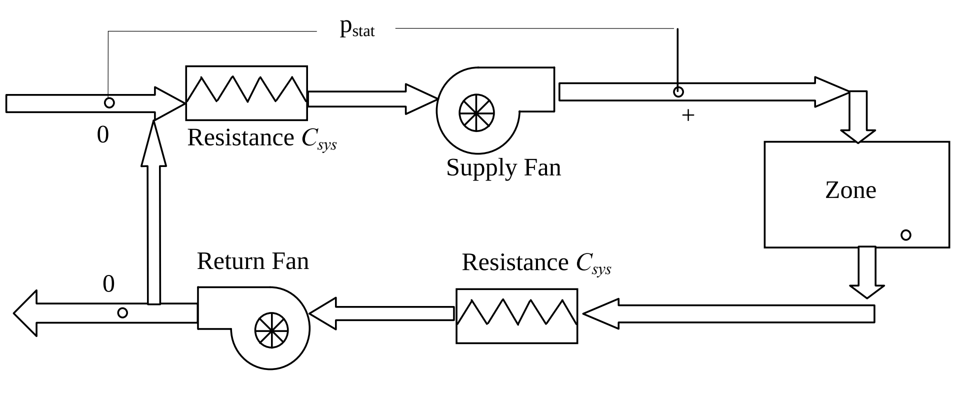


Figure 2: System diagram of the fan-air system simulated in the model

For the return fan subsystem, p_{stat} is the measured or assumed pressure in the occupied space and appears as a negative term, since a positive pressure in the space reduces the fan pressure rise required. The correction for the velocity pressure in the room is very small and can be ignored.

$$p_{sys} = -p_{stat} + C_{sys} m|m|$$

The fan operating point is where the pressure drop across the system equals the pressure increase across the fan, as shown in Figure 3.

The air flowing through the fan increases in temperature because of the heat added to the air stream due to fan inefficiency and due to motor inefficiency, if the fan is in the air stream. Because air is a compressible fluid and can be treated as a perfect gas, it can be shown that the fluid work performed by the fan results in the same temperature increase that would be obtained if the fluid work were completely converted to heat. (The opposite is true for incompressible fluids, such as water. In the case of incompressible fluids, the fluid work only appears as heat when the fluid passes through a dissipative element.)

Equating the fan pressure rise and system pressure drop yields:

$$k_{fan} n_{fan}^2 = p_{stat} + (\frac{1}{2\rho_{air} A^2} + C_{sys} + C_{fan}) \cdot m|m|$$

for the supply fan and a similar relationship for the return fan.

The fan shaft power is:

$$P_{shaft} = \frac{m \cdot p_{fan}}{\eta_{fan} \rho_{air}}$$

where η_{fan} is the efficiency of the fan.

The fan efficiency as a function of normalized airflow rate is assumed to vary quadratically about its maximum value:

$$\eta_{fan} = \eta_{fan,\max} - C_{\eta,fan} (\frac{m}{n_{fan}} - \frac{m_{\max}}{n_{fan}})^2$$

The total electric power is:

$$P_{tot} = \frac{P_{shaft}}{\eta_{VFD} \eta_{motor} \eta_{belt}} = \frac{m \cdot p_{fan}}{\eta_T \rho_{air}}$$

where η_{tot} is the combined efficiency of the variable frequency drive, motor, belt and fan.

The heat gain to the air stream is:

$$q_{gain}=P_{Tot}[\eta_T+(1- \eta_{motor}\, \eta_{belt}) f_{motor\&belt}]$$

where $f_{motor\&belt}$ indicates whether the motor is in the airstream (1) or out of the airstream (0).

The temperature rise across the fan is determined from:

$$(T_{air,out} - T_{air,in}) = q_{loss} / c_p \cdot m$$

Mixing box

A mixing box is the section of an air-handling unit used to mix the return air stream with the outside air stream. It consists of three sets of dampers whose operation is coordinated to control the fraction of the outside air in the supply air while maintaining the supply airflow rate approximately constant. Figure 4 is a simplified diagram of the mixing box simulated in the model. A variant of this design has a separate outside air damper that is adjusted to provide the minimum outside airflow required during occupancy.

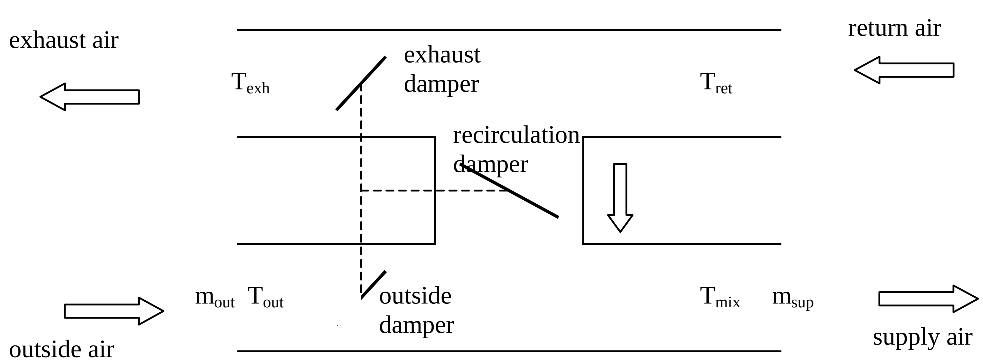


Figure 4: Flow directions and temperatures in a mixing box

Figure 5 shows ideal behavior and the range of acceptable behavior of a mixing box. The vertical axis is the outside air fraction, defined as:

$$OAF = \frac{m_{out}}{m_{\sup}}$$

To a good approximation, the temperatures in the mixing box are related to the outside air fraction by:

$$OAF = \frac{t_{ret} - t_{mix}}{t_{ret} - t_{out}}$$

Under the ideal conditions, the outside air fraction should range from 0 to 1 when the damper position varies from 0 to 100%. However, in general, there is leakage of both the outside air and the return air dampers; the outside air fraction then ranges between a minimum value that is greater than 0 and a maximum value that is less than 1. In addition, the air-flow rate is not necessarily linearly related to the damper position and therefore the mixed air temperature and humidity ratio are not linearly related to damper position. Theoretically, it is possible to predict the airflow rates of both the outside air and recirculation air streams as a function of damper position. The airflow rates could then be used to determine the outside air fraction and hence the mixed air temperature and humidity ratio. However, it is impractical to simulate the mixed air temperature accurately in this way because the pressure boundary conditions change with fan speed as a result of wind effects and because of the difficulty of estimating the authority of the dampers. This said, the behavior in the middle of the operating range is relatively unimportant compared to the behavior at the ends of the operating range.

Figure 5 shows the forms of the models to be used during commissioning and during routine operation following commissioning. In the model to be used at the commissioning stage, when only design information is available, the range of acceptable behavior is modeled. A 3:1 gain variation is used by default; when the damper position is 50%, the upper limit of the outside air fraction is 25% lower than its maximum and the lower limit is 25% above its minimum:

$$OAF_{lower} = d^2(1 - l_{ret})$$
$$OAF_{upper} = l_{out} + (2d - d^2)(1 - l_{out})$$

where d is the damper position (0-1) and l is the maximum acceptable fractional leakage. The square law relationships between OAF and d are used as a convenient approximation to the likely limits of acceptable performance in the absence of any recognized standard.

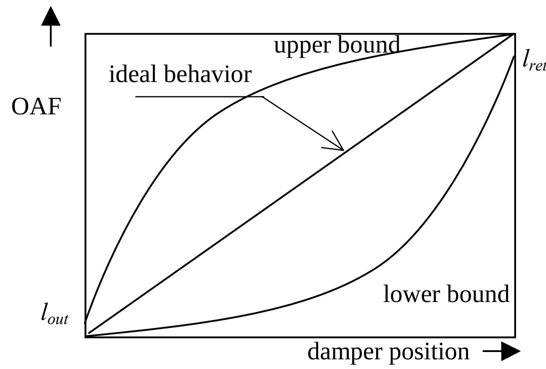


Figure 5: Mixing box performance

The maximum acceptable deviations from 0 and 100% outside air fraction at each end of the operating range should ideally be specified by the designer. Once the mixing box has been commissioned, the results of the functional test can be used to fit a polynomial to the measured variation of outside air fraction with the control signal:

$$OAF_{model} = (OAF_{\max} - OAF_{\min})\sum_{1}^{n} C_i d^i + OAF_{\min}$$

where the order, n, is constrained by $n<m$, where m is the number of measurements. The constraint ensures that the fitted curve passes through the measured value OAF_{max} at d=1 as well as through

$$\sum_{1}^{n} C_i = 1$$

OAF_{min} at d=0.

SELF-CONSISTENT DOCUMENTATION PROCESS

An integral part of developing a library of equipment reference models is the provision of detailed documentation of each model, including the equations, physical assumptions and numerical limitations, if applicable. Ideally, the documentation should always be up-to-date in the sense that it reflects the actual implementation of the model. It should also be clear and consistent, both for viewing and printing, and provide active cross-links to facilitate browsing, either within the document itself or to other external contextually-related documentation sources. Such requirements usually imply the need to generate documentation in different formats (e.g., MS Word, HTML and/or PDF) from the same meta-information. Such a task can easily become burdensome for the library writer, which typically results in an incomplete or inconsistent documentation effort.

In order to ensure self-consistency, and avoid possible discrepancies between different versions as well as likely duplication of effort, we propose an automated approach to generating documentation by embedding meta-information directly within the comments in the files containing the source code for the component models. The model writer specifies the documentation meta-information as part of the normal development and maintenance process. The comments can then be parsed so that the information is made available in customized formats for various outputs. Since the SPARK definition language essentially follows the same syntax as C/C++, with the addition of a few simulation-specific keywords, it is possible to use any automated documentation generation tool capable of parsing C/C++ source code to perform this task. We briefly describe how this has been accomplished using the widely-used doxygen tool (Doxygen 2006).

Doxygen, mostly written by Dimitri van Heesch, is an open-source documentation generator for C++, C, Java, Objective-C, Python, and IDL. Being highly portable, it runs on most Unix systems as well as on Windows and Mac OS X. Doxygen is a user-friendly tool that extracts specific source code comments and analyzes the declarations of the code, to create a comprehensive online documentation with active links. Doxygen also generates dependency graphs, inheritance diagrams, and collaboration diagrams to help in visualizing the relations between the various elements. If needed, it is possible to incorporate directed and undirected graphs in the documentation by directly embedding textual Graphviz (Graphviz 2006) declarations within the comments. By default, the output format is HTML but it can also be CHM (Microsoft compressed HTML), RTF (Microsoft Rich Text Format), PDF (Adobe Portable Document Format), LaTeX, PostScript or man pages. Because doxygen automatically takes care of generating the documentation in the different output formats from the same meta-description, the library writer does not need to be familiar with the formatting issues of each environment, thereby allowing attention to be focused on the content as opposed to the form.

With a few simple markups in the comments and manipulation of the options in the doxygen configuration files, it is possible to generate self-consistent and professional-looking documentation of the SPARK files automatically. Predefined style sheets can be used to further customize the look-and-feel and layout of the documentation in CHM (ideal for browsing on Windows platform), HTML (suitable for viewing in any browser) and PDF (ideal for printing). The code snippet in Appendix 1 demonstrates how a SPARK atomic class, valve.cc, can be made “doxygen-aware” by using doxygen keywords (shown with a bold typeface) within the C++-like comment blocks starting with the reserved token “///”. The result from applying doxygen to this SPARK atomic class is shown in Appendix 2 in the form of successive screenshots of the corresponding HTML output.

CONCLUSION

A set of models of the main components of HVAC air handling units and air distribution systems has been developed for use in model-based diagnostic tools. The models use simple first principles relationships, supplemented by empirical relationships, and have been designed to be configured using readily available performance data. The models are implemented in the freely distributed object-based simulation environment SPARK and documented using the doxygen automated documentation generator. The models will be made available at http://cbs.lbl.gov/diagnostics/model_library.

ACKNOWLEDGMENT

This work was supported by the California Energy Commission PIER Buildings Program through the California Institute for Energy and Environment and by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Federal Energy Management of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

REFERENCES

ASHRAE. 2004. HVAC Systems and Equipment Handbook, p41.7, American Society of Heating, Refrigerating and Air Conditioning Engineers, Atlanta, GA.

Brandemuehl, M. J., S. Gabel, I. Andersen. 1993. A toolkit for secondary HVAC system energy calculations, HVAC2 Toolkit. Prepared for The American Society of Heating, Refrigerating and Air Conditioning Engineers. TC 4.7 Energy Calculations. Atlanta, GA. ASHRAE.

Clark, D. R. 1985. HVACSIM+ Building Systems and Equipment Simulation Program Reference Manual, US National Institute for Standards and Technology, Gaithersburg, MD 20899.

Doxygen 2006. http://www.stack.nl/~dimitri/doxygen/ .

Graphviz 2006. http://www.graphviz.org/ .

Holmes, M. J. 1982. The simulation of heating and cooling coils for performance analysis, Proceedings of System Simulation in Buildings '82, Liège, Belgium, December.

Klein, S. A., Beckman, W. A. J. A. Duffie. 1976. TRNSYS - a transient simulation program, ASHRAE Transactions, Vol 82, Pt. 2

SPARK 2003. Simulation Problem Analysis and Research Kernel. Lawrence Berkeley National Laboratory and Ayres Sowell Associates, Inc. Downloadable from http://simulationresearch.lbl.gov/

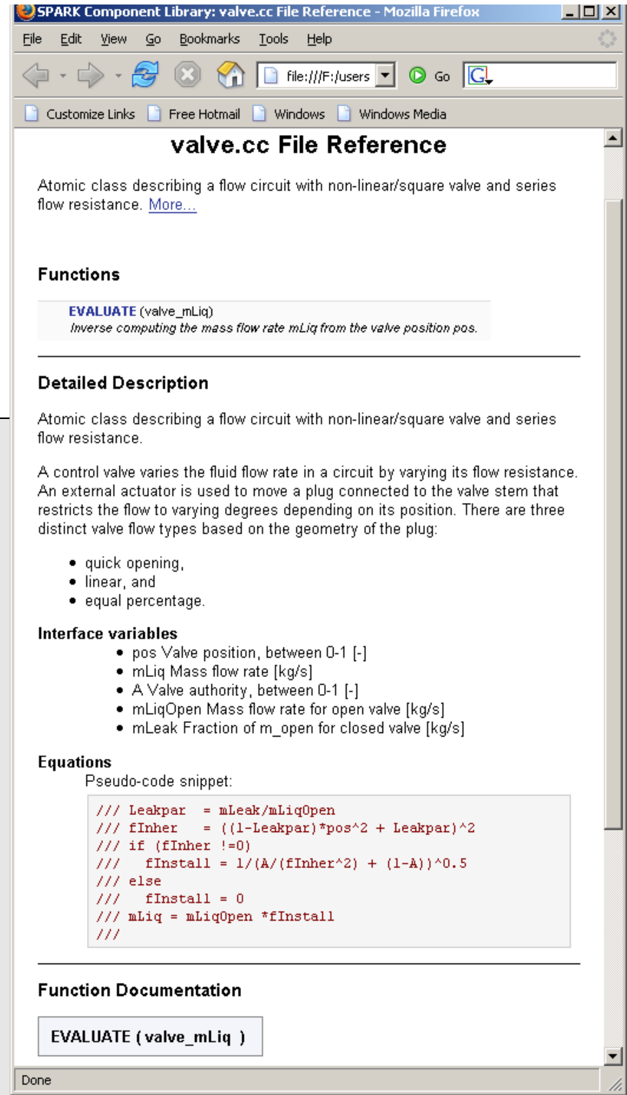
Sreedharan, P. and Haves, P. 2001. Comparison of Chiller Models for use in Model-Based Fault Detection, *Proc. International Conference for Enhancing Building Operations,* Austin, TX, July

Zhou, X.T. 2005. Dynamic modeling of chilled water cooling coils. Ph.D. Thesis. Purdue University.

APPENDIX I: ATOMIC CLASS VALVE.CC

```
/// \file  valve.cc
/// \brief Atomic class describing a flow circuit
/// with non-linear/square valve and series flow
/// resistance.
/// A control valve varies the fluid flow rate in a
/// circuit by varying its flow resistance. An
/// external actuator is used to move a plug
/// connected to the valve stem
/// that restricts the flow to varying degrees
///////////////////////////////////////////////////
/// \par  Interface variables
/// - pos             Valve position, between 0-1
/// - mLiq            Mass flow rate
/// - A               Valve authority, between 0-1
/// - mLiqOpen        Mass flow rate for open valve
/// - mLeak           Fraction of m_open for closed
///////////////////////////////////////////////////
/// \par  Equations
/// Pseudo-code snippet:
/// \code
/// Leakpar  = mLeak/mLiqOpen
/// fInher   = ((1-Leakpar)*pos^2 + Leakpar)^2
/// if (fInher !=0) fInstall = 1/(A/(fInher^2) +
(1-A))^0.5
/// else            fInstall = 0
/// mLiq     = mLiqOpen *fInstall
/// \endcode
///////////////////////////////////////////////////
…
/// \brief     Inverse computing the mass flow rate
///  mLiq from the valve position pos.
/// \return    Mass flow rate [kg/s]
/// \param     pos  Valve position, between 0-1
/// \param     A    Valve authority, between 0-1
/// \param     mLiqOpen Mass flow rate for open
EVALUATE(valve_mLiq)
{…}
```

APPENDIX 2: HTML DOCUMENTATION FOR VALVE.CC



valve.cc File Reference

Atomic class describing a flow circuit with non-linear/square valve and series flow resistance. More...

Functions

EVALUATE (valve_mLiq)
Inverse computing the mass flow rate mLiq from the valve position pos.

Detailed Description

Atomic class describing a flow circuit with non-linear/square valve and series flow resistance.

A control valve varies the fluid flow rate in a circuit by varying its flow resistance. An external actuator is used to move a plug connected to the valve stem that restricts the flow to varying degrees depending on its position. There are three distinct valve flow types based on the geometry of the plug:

- quick opening,
- linear, and
- equal percentage.

Interface variables

- pos Valve position, between 0-1 [-]
- mLiq Mass flow rate [kg/s]
- A Valve authority, between 0-1 [-]
- mLiqOpen Mass flow rate for open valve [kg/s]
- mLeak Fraction of m_open for closed valve [kg/s]

Equations

Pseudo-code snippet:

```
/// Leakpar  = mLeak/mLiqOpen
/// fInher   = ((1-Leakpar)*pos^2 + Leakpar)^2
/// if (fInher !=0)
///    fInstall = 1/(A/(fInher^2) + (1-A))^0.5
/// else
///    fInstall = 0
/// mLiq = mLiqOpen *fInstall
///
```

Function Documentation

EVALUATE (valve_mLiq)