WHAT NEXT FOR BUILDING ENERGY SIMULATION—A GLIMPSE OF THE FUTURE

D. B. Crawley
U.S. Department of Energy
Washington, DC 20585 USA

L. K. Lawrie
Construction Engineering Research Laboratories
Champaign, Illinois 61821 USA

F. C. Winkelmann, W. F. Buhl and A. E. Erdem
Lawrence Berkeley National Laboratory
Berkeley, California 94720 USA

C. O. Pedersen, R. J. Liesen, D. E. Fisher,
R. K. Strand and R. D. Taylor
University of Illinois
Urbana, Illinois 61801 USA

ABSTRACT

The U.S. Departments of Energy and Defense (DOE and DOD) are jointly developing EnergyBase, a new building energy simulation tool that builds on the capabilities of BLAST and DOE-2. EnergyBase will include innovative simulation features, including variable time steps, built-in template and external modular systems simulation modules integrated with a heat balance-based zone simulation, and input and output data structures tailored to facilitate third party interface development.

To provide input to future planning efforts, we sponsored workshops in August 1995 and June 1996 on next-generation building energy simulation tools. We first describe the methods used and results from the two workshops. We then give a brief overview of the organization and anticipated capabilities of EnergyBase.

INTRODUCTION

Many building energy simulation programs developed around the world are reaching maturity. Many use simulation methods (and even code) that originated in the 1960s. Without substantial redesign and recoding, expanding their capabilities has become difficult, time-consuming, and expensive. However, recent advances in analysis and computational methods and power have increased the opportunity for significant improvements in these tools.

In early 1996, DOE and DOD began developing a new building energy simulation tool that builds on their experience with two existing programs: DOE-2 (Winkelmann et al. 1993) developed by Lawrence Berkeley National Laboratory (LBNL) and BLAST (BLAST Support Office 1992) developed by U.S. Army Construction Engineering Research Laboratories (CERL) and University of Illinois (UI). The new program—EnergyBase—is expected to become available in 1998. As we begin testing EnergyBase, the team will begin planning development of next-generation building simulation tools that go substantially beyond the capabilities of simulation programs available today.

To inform the planning activities for both EnergyBase and the next-generation simulation tools, DOE and DOD held workshops in August 1995 and June 1996. Energy simulation developers and expert users were invited to the first workshop (developers workshop) held after Building Simulation '95 in Madison, Wisconsin. Energy simulation users and other professionals attended the second workshop (users workshop), held in Washington, D.C. This paper describes the structure and results of the two workshops and the current plans and structure for EnergyBase.

STRUCTURE OF THE WORKSHOPS

The goal of both workshops was to generate and prioritize ideas for next-generation simulation environments where the scope was simulation of building life-cycle processes that influence energy performance and environmental sustainability. The developers workshop focused on applications, capabilities, and methods and structures; the users workshop focused on applications, capabilities, and user interfaces. Participants were reminded that the workshops were not a forum to discuss pros and cons of any existing tool, or to decide who might perform any development work for any potential U.S. next-generation simulation tools.

Each workshop was organized in three breakout sessions: Applications, Capabilities, and Methods and Structures for the developers workshop; Applications, Capabilities, and User Interfaces for the users workshop. We divided the participants into groups each facilitated by a member of the EnergyBase team. The facilitators used a five-step process for each of the breakout sessions: brainwriting, grouping and eliminating duplicate ideas, brainstorming, prioritizing and Pareto voting, and summarizing.

At the beginning of each breakout session the facilitators described the general subject of the session. Then, the groups began brainwriting in which each workshop participant writes down one idea on a note card and passes that card to their right.

As cards are passed, each person reviews the idea and continues to generate their own new ideas. Brainwriting encourages idea generating through individual creativity and brainpower. After 10-15 minutes the groups organized the cards/ideas into general groups and eliminated duplicate ideas. To make sure no important ideas were missed, the groups then spent 10-15 minutes brainstorming—working as a group to generate new ideas. After brainstorming, each group counted the number of cards/ideas and multiplied by 0.2. This was the number of votes each participant had when selecting their top 20% of the ideas (Pareto voting). Votes (using dots) were applied to the cards once all participants in a group had selected their top 20%. The groups then rank-ordered the cards from highest priority (most votes) to lowest priorities (fewest votes). Voting provided a relative ordering of the ideas within each group—all of the ideas generated would be useful to the group. Last, each facilitator prepared a summary that they presented to the entire workshop at the end of each breakout session.

RESULTS OF THE WORKSHOPS

The following figures present summary grouping of the concepts and ideas generated in the two workshops. In total, the developers workshop generated 225 ideas for the Applications breakout session, 242 ideas for the Capabilities breakout session, and 201 ideas for the Methods and Structures breakout session. The users workshop (with more participants) generated 247 ideas for the Applications breakout session, 301 ideas for the Capabilities breakout session, and 213 ideas for the User Interface breakout session.

Figure 1 compares the application priorities of users and developers. The raw votes of software developers and users were normalized and plotted as percentages in the figure. Predictably, users disagreed with developers on the importance of research. The significance placed on design by the user community was also not surprising. But although the expected bias of the two groups is discernible, there is remarkable agreement on program application priorities. This indicates that, for the most part, researchers and developers are cognizant of the needs of the user community.

A similar trend can be seen in Figure 2, which compares the capability priorities of users and developers. For the most part, developers seem to be aware of user concerns and priorities. The most serious disconnect occurs on the issue of input and output capabilities. This category was clearly a high priority for users but a lower priority for developers.

As shown in Figure 3, users' top priorities for software program interfaces were interoperability and integration with other building tools such as CAD

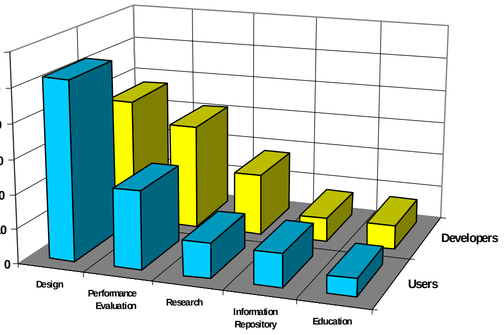

Figure 1. Program Application Priorities of Developers and Users

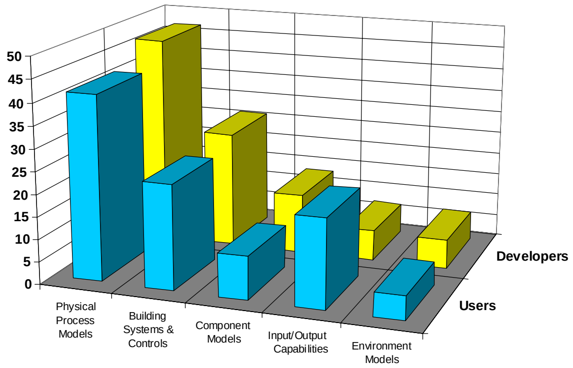

Figure 2. Program Capability Priorities of Developers and Users

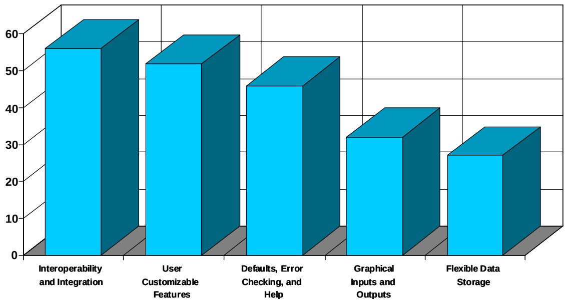

Figure 3. Program Interface Priorities of the Users Workshop

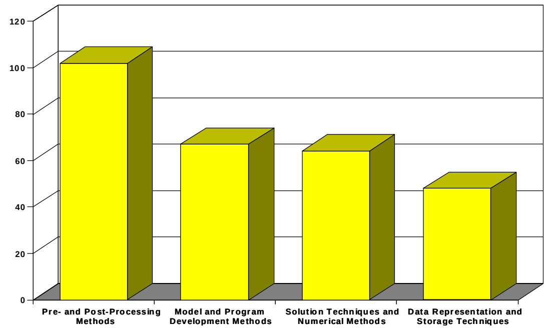

Figure 4. Program Methods and Structures Priorities of the Developers Workshop

and customizability. Still important but with less agreement as to relative importance was graphical input/output, defaults/error checking/help, and data storage. One 'fun' concept that came from one of the user teams was a TUI–similar to GUI (Graphic User Interface) but instead a Telepathic User Interface–at least some of the participants were thinking 'outside the box'.

In Figure 4, the developers' topic priorities for program methods and structures are shown. By far the most important issue for the developers was pre- and post-processing methods–similar to the users' priorities of interoperability and integration. The other three categories were considered important but of lessor priority. The authors conjecture that this occurred because developers feel they have these issues under control.

Tables 1 through 4 show the votes by topic within each category from the users and developers workshops. Tables 3 and 4 (as with Figures 3 and 4) show information only for the users and developers workshops respectively.

ENERGYBASE, COMBINING BLAST AND DOE-2

For the past twenty years, the U.S. government has maintained and supported two building energy simulation programs, DOE-2 and BLAST. DOE-2, supported by DOE, has its origins in the Post Office program written in the late 1960s for the U.S. Post Office. BLAST, supported by DOD, has its origins in the NBSLD program developed at the U.S. National Bureau of Standards (now NIST) in the early 1970s. The primary difference between the programs is load calculation method—DOE-2 uses a room weighting factor approach while BLAST uses a heat balance approach.

The need for two separate government-supported programs has long been questioned. Discussions on merging the two programs began in earnest in April 1994 with a DOD-sponsored workshop in Illinois. No concrete plans came out of that workshop, but eventually, under the initiative of DOE, a merger project has begun. This new program, EnergyBase, will combine the best capabilities of DOE-2 and BLAST, and begin the restructuring process necessary to make it easier to modify and extend the merged program. The overall structure proposed for EnergyBase is shown in Figure 5.

The major concept behind the merger is to combine the heat balance engine of the IBLAST program (a version of BLAST with integrated building, system and plant simulation) with a generalized HVAC engine that includes system types from BLAST and DOE-2 and links to MODSIM (from HVACSIM+) and SPARK. The heat balance engine will also be restructured to accommodate the daylighting algorithms and WINDOW-4-based fenestration calculations used in DOE-2 as well as new ground heat transfer and interzone airflow models. Through a translator, EnergyBase will be able to read both DOE-2 building description language and BLAST input files. Depending on the progress made by the International Alliance for Interoperability (Bazjanac and Crawley 1997), a common object-oriented data store may eventually become the main interface to the program.

One of the main goals of the EnergyBase development effort is to create an organized, modular program structure that allows easy additions of features and links to other programs. New FORTRAN 90 code will be developed for all modules. Significant reengineering of concepts from BLAST and DOE-2 will be used to develop the new modules.

EnergyBase is an interim step along the path to a truly next-generation energy analysis program. The EnergyBase team includes CERL, UI, LBNL, and DOE. For more details on the design concepts and structure intended for EnergyBase, see Pedersen et al (1997). The merged program is scheduled to begin testing spring of 1998.

SUMMARY

A surprising outcome of the workshops (at least for the authors) was that not many new or unusual ideas were brought up—even with a group of international building energy simulation developers and users. The hundreds of ideas generated during the workshops showed instead that the field of building energy simulation has many fundamental issues that are being addressed. Even the developers were not willing to stretch the boundaries and capabilities of simulation (even in their own minds) until more of these basic issues are resolved.

We note that participants in both workshops identified similar topics of concern and priority. Using any simulation program for design is high on both lists (though naturally a stronger issue for users). The main differences appear in the areas where we split the focus of the workshops—Interface, and Methods and Structures. The interface priorities identified in the user workshop are crucial to the success of any next-generation tool in the building simulation area.

For users, recurrent themes throughout were design, environment, economics, and occupant comfort and safety. Designers need tools that provide answers to very specific questions during design. They are less concerned with the mechanics of the tools–although they want tools that provide the highest level of simulation accuracy and detail reasonably possible. The developers focused more on model and module development, and related issues. From the similar priorities identified, it is clear that the developers at least recognize the concerns of their users.

Although the workshops pointed up the critical nature of user interface for the success of any simulation tool, the EnergyBase team is first focusing on development of the heart of a new simulation tool—the calculation engine. In that area, we are consciously incorporating the priorities of the workshop participants in our development effort (many can be seen in Figure 5). In the near future, potential third-party developers of user interfaces will be invited to participate in EnergyBase—to bring the importance of interface issues into the project.

NEXT STEPS

In 1997, the EnergyBase team will begin formulating a plan to develop the next generation of building energy simulation tools in the United States. The plan will propose development of software that goes substantially beyond the capabilities of currently available tools and that has a broader scope in the building simulation arena. The results of the two workshops will be used to set priorities for applications, capabilities, methods and structures, and interface concepts in the next generation tools. It is our intent to structure development of the next generation tools as an open process so that a number of contributors from the United States and other countries can and will participate.

The authors hope that the information gathered in the workshops will be a starting point for encouraging simulation developers and users to talk more. The complete list of ideas generated during the workshops is available from the authors.

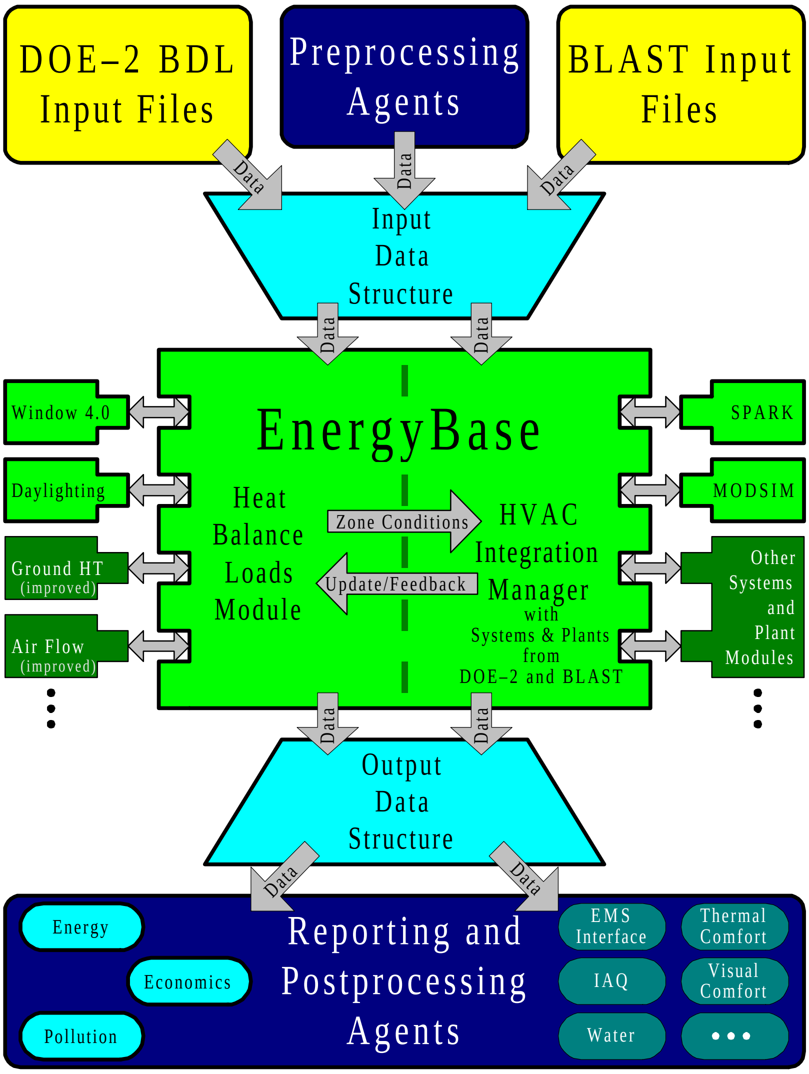

Figure 5. Proposed Structure of EnergyBase

ACKNOWLEDGMENTS

The authors participated as leaders and facilitators for the developers and users workshops. The authors wish to thank the participants for their contributions to the workshops and the long range planning efforts that will evolve from them. Participants in the developers workshop included:

Workshop Leader
Dru Crawley, U.S. Department of Energy

Facilitators
Linda Lawrie, U.S. Army CERL
Curt Pedersen, University of Illinois
Fred Winkelmann, Lawrence Berkeley National Lab.
Dan Fisher, University of Illinois
Rich Liesen, University of Illinois

Participants
Marlin Addison, Energy Simulation Specialists, Inc.
Doug Balcomb, National Renewable Energy Lab.
Chip Barnaby, WrightSoft
Vladimir Bazjanac, Lawrence Berkeley National Lab.
Ian Beausoleil-Morrison, Natural Resources Canada
Bill Beckman, University of Wisconsin
Nathan Blair, University of Wisconsin
Jean-Pascal Bourdouxhe, Universite de Liege, Belgium
Rob Briggs, Pacific Northwest National Laboratory
Axel Bring, Royal Institute of Technology, Sweden
Bill Carroll, Lawrence Berkeley National Laboratory
Joe Clarke, University of Strathclyde, Scotland
Sten de Wit, Tech. University Delft, The Netherlands
Mike Donn, School of Architecture, Victoria University, New Zealand
Helmut Feustel, Lawrence Berkeley National Lab.
Jon Hand, University of Strathclyde, Scotland
Philip Haves, Loughborough University of Technology, England
Doug Hittle, Colorado State University
Michael Holtz, Architectural Energy Corporation
Bob Howe, Carrier Corporation
Ron Judkoff, National Renewable Energy Laboratory
John Mitchell, University of Wisconsin
John Murphy, The Trane Company
Kosta Papamichael, Lawrence Berkeley National Lab.
Roger Pelletret, Centre Scientifique et Technique du Batiment, France
Per Sahlin, Royal Institute of Technology, Sweden
Peter Schwartz, Pacific Gas and Electric Company
John Seem, Johnson Controls, Inc.
Ed Sowell, California State University, Fullerton
Jeff Spitler, Oklahoma State University
Paul Strachan, University of Strathclyde, Scotland
George Walton, National Institute of Standards and Technology
Mike Witte, GARD Analytics
Gerhard Zweifel, EMPA, Switzerland

Participants in the users workshop included:

Workshop Leader
Dru Crawley, U.S. Department of Energy
Facilitators
Curt Pedersen, University of Illinois
Mike Case, U.S. Army CERL
Fred Winkelmann, Lawrence Berkeley National Lab.
Kosta Papamichael, Lawrence Berkeley National Lab.
Dan Fisher, University of Illinois
Bill Carroll, Lawrence Berkeley National Lab.
Rich Liesen, University of Illinois

Participants
Michael Andelman, Shooshanian Engineering Associates
Michael Anello, Florida Solar Energy Center
Doug Balcomb, National Renewable Energy Lab.
Vlado Bazjanac, Lawrence Berkeley National Lab.
Rob Briggs, Pacific Northwest National Laboratory
Martha Brook, California Energy Commission
John Castelvecchi, Virginia Power
Ted Collins, U.S. Department of Energy
Larry Degelman, Texas A&M University
Joe Deringer, The Deringer Group
K. Dean Devine, Engineering Analysis
Terry DuBois, U.S. Department of Energy
Gary Epstein, Energy & Resource Solutions
Jim Fireovid, SAIC
Rajinder Garg, U.S. Department of Veterans Affairs
Harry Gordon, Burt Hill Kosar Rittelmann Associates
Christopher Gribbs, American Institute of Architects
Debra Haltrecht, Natural Resources Canada
Mary-Margaret Jenior, U.S. Department of Energy
Theresa Jenne, Honeywell, Inc.
Barry Jones, HOK Architects, Inc.
Ron Judkoff, National Renewable Energy Laboratory
Michael Kintner-Meyer, SAIC
Satish Kumar, Carnegie Mellon University
Malcolm Lewis, RBFW & Associates
Gail Lindsey, Design Harmony
Robert Mackie, U.S. Department of Energy
Ardeshir Mahdavi, Carnegie Mellon University
Paul Mathew, Carnegie Mellon University
Joe McCarty, U.S. Army Corps of Engineers
Daniel Nall, Roger Preston + Partners
Fred Roberts, Solaequis Software
Thomas R. Rutherford, U.S. Department of Defense
Muthusamy Swami, Florida Solar Energy Center
Michael Tinkleman, Electric Power Research Institute
Adrian Tuluca, Steven Winter Associates
Gren Yuill, Pennsylvania State University

REFERENCES

Bazjanac, V. and D. B. Crawley. 1997. “The Implementation of Industry Foundation Classes in Simulation Tools,” in *Building Simulation ’97, Fifth International Conference Proceedings*, Prague, Czech Republic, September 1997. IBPSA.

BLAST Support Office. 1992. *BLAST 3.0 Users Manual*. Urbana-Champaign, Illinois: BLAST Support Office, Department of Mechanical and Industrial Engineering, University of Illinois.

Mitchell, J. W., and W. A. Beckman, editors. 1995. *Building Simulation ‘95, Fourth International Conference Proceedings*, Madison, Wisconsin, 14-16 August 1995. IBPSA.

Pedersen, C. O., D. E. Fisher, R. J. Liesen, R. K. Strand, R. D. Taylor, W. F. Buhl, F. C. Winkelmann,

L. L. Lawrie, D. B. Crawley. 1997. “EnergyBase—The Merger of BLAST and DOE-2,” in *Building Simulation ’97, Fifth International Conference Proceedings,* Prague, Czech Republic, September 1997. IBPSA.

Winkelmann, F. C., B. E. Birdsall, W. F. Buhl, K. L. Ellington, A. E. Erdem, J. J. Hirsch, and S. Gates. 1993. *DOE-2 Supplement, Version 2.1E*, LBL-34947, November 1993, Lawrence Berkeley Laboratory. Springfield, Virginia: National Technical Information Service.

Table 1. Program Application Priorities of Developers and Users

Design			
Developers	**Votes**	**Users**	**Votes**
Collaborative, integrated, facilitated building design	39	Envelope design	37
Building code compliance—energy and environmental impact	18	Early analysis of design alternatives	25
System selection and equipment sizing wizards	16	Environmental impact and sustainability	24
Lighting/daylighting (selection of products, performance assessment)	7	Economic and cost analysis	15
Aid in selecting retrofit strategies	7	System design	14
		Occupant comfort and safety	11
		Retrofit design	3

Performance Evaluation			
Developers	**Votes**	**Users**	**Votes**
Comfort evaluation	21	Performance contracting	16
Economic, life cycle, and cost-benefit analysis	14	Code development and compliance	11
Optimal operation and control	14	Performance data acquisition and analysis	8
Control strategies/ optimization/ supervisory	13	Commissioning	7
Indoor air quality	12	Comfort- and energy-based controls	7
		Fault detection and diagnostics	7

Research			
Developers	**Votes**	**Users**	**Votes**
Policy formation code development	9	Emerging technologies and new processes	11
Solution of inverse problem to calibrate model for existing building	6	Occupant health and productivity	8
Basic research	5	Environmental impact	6
Sensitivity and error analysis	5		
Provide basis for simplified	4		

Information Repository			
Developers	**Votes**	**Users**	**Votes**
Electronic owner’s manual (building life cycle)	9	Performance databases and libraries	12
Feed intelligent database for future designs	5	Design databases and libraries	8
Need for structural libraries of models, object-oriented programming	3	Expert systems	4
No gap between description and behavior; i.e. performance data immediate after object selection	2		
Use of historical data files, previous work/buildings	2		

Education			
Developers	**Votes**	**Users**	**Votes**
Student and practitioner education	23	Student education	13
Make it fun	2		

Table 2. Program Capability Priorities of Developers and Users

Physical Process Models			
Developers	**Votes**	**Users**	**Votes**
Air flow modeling	25	Envelope/environment interaction	47
Moisture absorption/desorption in building materials	17	Heat transfer models	37
1-, 2-, and 3-D transient conduction	15	Air infiltration and movement within spaces	22
Daylighting	14	Realistic simulation time steps	7
Full generality 3-dimension shading, lighting, and solar geometry	14	Moisture	7
		Indoor air quality	5

Building Systems and Controls			
Developers	**Votes**	**Users**	**Votes**
Flexible system and plant modeling	18	Integrated systems with modular component models	21
First principles system and plant models	14	Realistic building and HVAC simulation	18
Imperfect mixing of zone air	13	Process (e.g. moisture, daylighting) and component controls	12
Zones, systems, plants coupling	8	Performance, compliance and validation	10
Passive and active solar	6	Multiple building systems	7
		Human interaction models	3

Component Models			
Developers	**Votes**	**Users**	**Votes**
Advanced fenestration	11	Air delivery system component models	10
Energy storage in buildings including phase change	8	Central plant equipment models	10
Advanced lighting system modeling	4	Building envelope component models	7
Dynamic coil models	3	Multilevel component models	2
Duct losses	3		

Input and Output Capabilities			
Developers	**Votes**	**Users**	**Votes**
Variable time step	5	Flexible inputs and outputs	26
Uncertainty analysis	4	Life-cycle and real time cost analysis	11
Economic Analysis	3	Expert systems	7
Costs based on utility rate schedules modular interchangeable features	2	Optimization	7
Shell to facilitate the combining of components into a system	2	Access library and database information	4
		Design support	3
		Multi-platform, parallel processing	2

Environment Models			
Developers	**Votes**	**Users**	**Votes**
Occupant comfort	9	Pollution models and environmental impact	6
Typical, extreme and site-specific weather	5	Daylighting	6
Wind pressure distribution	4	Micro and macro weather data	4
Modeling of terrain and surrounding obstructions	2		
Long-term climates with special peak conditions and micro-climates	1		

Table 3. Program Interface Priorities of Users

Interoperability and Integration	
Users	**Votes**
Interoperable with other tools	22
Interoperable with CAD programs	20
Integration of components and analysis modules	10
Multi-platform applicability	4

User Customizable Features	
Users	**Votes**
Multilevel inputs	13
Simple input options	13
Clear separation of interface and computational engine	10
Customizable output and reports	7
Customizable interface	6
Adaptable to multiple uses	3

Defaults, Error Checking, and Help	
Users	**Votes**
Context sensitive and "smart" help	17
Knowledge-based analysis of inputs and output	10
Automated error and range checking	7
Tutorials and documentation	7
Online support	5

Graphical Input and Output	
Users	**Votes**
Graphical representation of inputs	12
Graphical output of results	10
Three dimensional spatial displays	10

Flexible Data Storage	
Users	**Votes**
Component libraries	16
External databases and manufacturer's catalogs	11

Table 4. Program Methods and Structures Priorities of Developers

Pre and Post Processing Methods	
Developers	**Votes**
Adaptable interface according to user type and stage of design process	21
Knowledge-based front end with intelligent defaults	15
Visualization of complex outputs, including virtual reality display	10
CAD integration	7
Validation by empirical, analytical, and comparative techniques	7

Model and Program Development Methods	
Developers	**Votes**
Object-oriented representation	12
Model reduction	6
Modularity of components	6
Equation-based models—NMF format	5
Tool able to be used by a team (concurrency)	5

Solution Techniques and Numerical Methods	
Developers	**Votes**
Simultaneous solution of loads plant and controls	5
Stochastic methods	5
Macroscopic air-flow modeling (non-CFD)	4
Numeric nodal approach for maximum future flexibility	4
Powerful differential-algebraic equation solvers	4

Data Representation and Storage	
Developers	**Votes**
Extensive and extensible libraries of building components and systems	13
Online documentation, structuring information	6
Flexible structure to allow quick change in systems configuration	5
Standardized data structures	5
Case studies database for decision-making	4