DEVELOPMENT OF A COMPONENT PROGRAM LIBRARY FOR BUILDING ENERGY SIMULATION - A JAPANESE EXPERIENCE -

by

Hisaya Ishino, Associate Professor, Department of Architectural Engineering, Tokyo Metropolitan University, 2-1-1 Fukazawa, Setagaya-ku, Tokyo 158, Japan

Masanori Shukuya*, Associate Professor, Department of Architecture, Musashi Institute of Technology, 1-28-1 Tamazutsumi, Setagaya-ku, Tokyo 158, Japan

Kimiko Kohri, Research Associate, Department of Architectural Engineering, Utsunomiya University, 2753 Ishii-machi, Utsunomiya-shi, Tochigi 321, Japan

ABSTRACT

This paper describes the concept of developing a component program library for building energy simulation and the current state of its development, which has been made by a working group of the Society of Heating, Air-Conditioning and Sanitary Engineers, Japan (SHASEJ) since 1985. The purpose of developing this component program library is to assist researchers and engineers in designing their own programs for their own objectives.

Since the early 1970s, quite a few building energy simulation programs have been developed: for example, NBSLD, DOE-2 and BLAST are internationally well known programs and, in Japan, programs called HASP and ACSS are also available. Although these programs are usually suitable for simulating a normal building, it is not often that they can automatically be used to simulate new energy saving strategies in air-conditioning systems, innovative window systems and so on. To accomplish such simulations, one usually must modify portions of the computer program being used. This necessarily causes a certain amount of tedious work and occasionally seems almost impossible. These problems may be resolved by the use of the component program library for building energy simulation.

Each component program in the library described in this paper has been written in FORTRAN 77 and designed so that it has the fewest number of lines as possible (the average is about 30 lines. A program having more than 100 lines is not allowed.). Every component program's documentation describes how to use it, an example of its use, fundamental equations involved, its limitations and the whole list. So far, this component program library contains approximately 180 programs.

During the course of development, we found that many component programs can be classified in a tree-like structure. This suggests that a component program library may have a potential for constructing a simulation program to conveniently and accurately model a room or building. In other words, this approach may help us make a program which will simulate a room or building as we assemble building elements and various heating/cooling system components to construct the room or building.

INTRODUCTION

Since the early 1970s, quite a few building energy simulation programs have been developed: for example, NBSLD[1], DOE-2[2] and BLAST[3] are internationally well known programs and, in Japan, programs called HASP[4] and ACSS[5] are also available. These programs were usually designed so that they can simulate ordinary buildings that have commonly available windows, walls, and heating/cooling systems; it is not often that they can automatically be used to simulate new energy saving strategies in air-conditioning systems[6,7,8], innovative window systems[9], and geometrically complicated buildings such as atria[10], and so on.

To do such simulations, one usually must modify portions of a computer program being used. This necessarily causes a certain amount of tedious work and occasionally seems almost impossible, since building energy simulation programs usually have large numbers of lines and complicated structure.

When we need an inverse matrix to do some calculation within a course of building simulation, we do not necessarily develop a program making the inverse matrix, but we can use a subroutine program in a mathematical library[11,12,13]. The same is true, when we need a statistical analysis or a graphic presentation of the results of calcula-

* Currently, Visiting Scientist, Windows and Daylighting Group, Lawrence Berkeley Laboratory, University of California, Berkeley CA 94720, U.S.A.

tion; we can usually use a subroutine library for statistical analysis[14,15], or a subroutine library for graphics[16].

By analogy, if we have a subroutine library (component program library) for building simulation, it can help solve the above-mentioned problems. This paper describes the current status of such a subroutine library, which has been under development by a working group of the Society of Heating, Air-Conditioning and Sanitary Engineers, Japan (SHASEJ) since 1985. We have found that many component programs can be classified in a tree-like structure and that such a component program library can have a great potential for constructing a simulation program as we assemble building elements and various heating/cooling system components to construct a room or building.

ITEMS INVOLVED IN THE BUILT ENVIRONMENT

A building consists of several rooms. A room consists of several building elements: walls, windows, ceiling and floor. The walls comprise several layers. A HVAC system is composed of pipes and equipments: all pipes and equipments can be regarded as kinds of heat exchangers from the view point of heat transfer. Accordingly, we can separate all building and HVAC system elements into fundamental elements. Therefore, we may regard one building and its HVAC system as a combination of these fundamental elements.

As shown in Figure 1, there are a lot of items involved in the built environment. They are interwoven with each other and then contribute to forming the built environment. We may consider these items as the fundamental elements for constructing a building energy simulation program. For example, the solar heat gain through a clear glass pane can be considered to be composed of several items: solar radiation, solar position, angle of incidence, solar optical property of the glass pane, and the inside/outside film heat transfer coefficients. These items are often common to various calculations; solar position is not only necessary to calculate the solar heat gain through windows, but also necessary to calculate daylight illuminance inside buildings, shadow areas due to overhangs or side fins and so on.

REQUIREMENTS IN DEVELOPING A COMPONENT PROGRAM LIBRARY

A subroutine program, which makes a calculation with respect to an item described above, can assist researchers and practitioners, who may not necessarily always need to know and use everything in a whole building energy simulation pro-

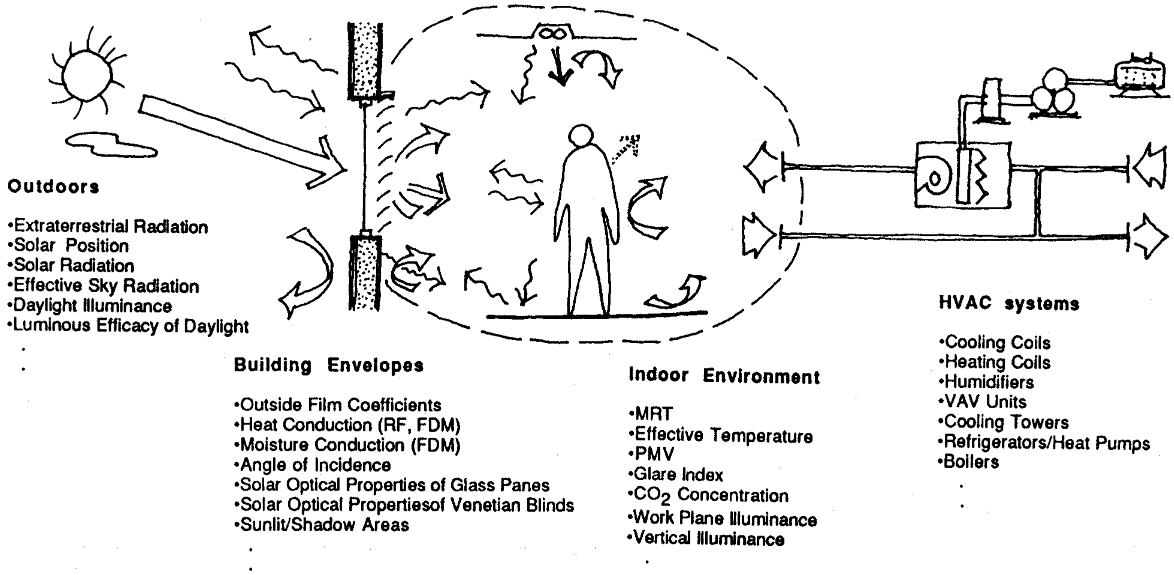


Fig. 1 Items involved in the built environment. They are interwoven with each other and contribute to forming the built environment.

gram. They could then design their own programs for their own purposes. In developing such a subroutine program and finally forming a subroutine library (component program library) for building energy simulation, we have taken into consideration the following.

Programming Language

There are several programming languages applicable for scientific and engineering calculation: FORTRAN, BASIC, PASCAL and C. We decided to use FORTRAN, because it has been used by numerous scientists and engineers for many years since it was first conceived in the 1950s. FORTRAN appears to be eligible for developing a component program library because of its well-defined concept of subroutine and function. These days it has also become available even for micro computers[17].

Length of a component program

We have been making component programs with the fewest lines as possible. A program having more than 100 lines is not allowed; this is to make each component program easier to understand. In general, we found that it is rarely necessary to develop a component program having more than 50 lines.

Arguments

It is desirable to make the arguments of a component program easy to use. What should also be kept in mind is that outputs and inputs of the component program required by a practitioner may differ from those required by a researcher.

Documentation

Each component program must be well-documented. Its documentation contains the purpose of the program, how to use it, an example of its use, the fundamental equations involved, its limitations and the whole list.

EXAMPLE OF A COMPONENT PROGRAM

Figure 2 shows an example of a component program called FSB. This component program FSB can calculate the form factor subtended by a window with horizontal louvers, assuming that the surfaces of the louver slats are ideally diffuse[18,19]. A user of this program must input the slat width, slat spacing, slat angle and the lengths concerned with the geometry of the window. The outputs are the form factor subtended by the void spaces between the louver slats and the form factor subtended by the louver slats.

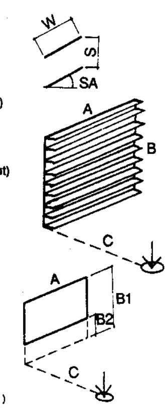
1. Program Name: FSB

2. Purpose

This porogram calculates the form factor subtended by a rectangular window with horizontal louvers.

3. Usage

- CALL FSB(W,S,SA,A,B,C,FS,FLB)
- Arguments
  - W: slat width [m] (input)
  - S: slat spacing [m] (input)
  - SA: slat angle [rad.] ($0 \leq SA < \pi/2$) (input)
  - A: see Figure. (input)
  - B: see Figure. (input)
  - C: see Figure. (input)
  - FS: form factor subtended by void spaces between the louver slats (output)
  - FLB: form factor subtended by the louver slats (output)



- Example

```
      PROGRAM TFSB
      DATA W/0.0125/,S/0.01/,
     $     A,B1,B2,C/2.5,2.0,0.3,2.0/,
     $     DTOR/0.0174533/
 10   READ(*,500)SA
500   FORMAT(F5.1)
      IF(SA.LT.0.)STOP
      SA=SA*DTOR
      CALL FSB(W,S,SA,A,B1,C,FS1,FLB1)
      CALL FSB(W,S,SA,A,B2,C,FS2,FLB2)
      FS=FS1-FS2
      FLB=FLB1-FLB2
      WRITE(*,600)FS,FLB
600   FORMAT(1H ,'FS=',F5.3,'   FLB=',F5.3)
      GO TO 10
      END
```

4. Fundamental Equations

If $\alpha \leq 0$ then $f_s = 0$, $f_{lb} = U_V$

If $0 < \alpha$ then $f_s = [(1-a)(1-\cos z)+b \sin z - b \ln\{\tan(\pi/4+z/2)\}]\, U_{Vo}/2$,

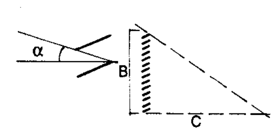
$f_{lb} = U_V - f_s$

where $\alpha = \tan^{-1}(1/b-a/b)$

$a = (W \sin SA)/S$

$b = (W \cos SA)/S$

$U_{Vo} = U_V/\{(1-\cos\theta)/2\}$



If $\alpha < \theta$ then $z = \alpha$

If $\theta \leq \alpha$ then $z = \theta$

where $\theta = \tan^{-1}(B/C)$

5. Limitations

This program uses a component program UHV. Note $0 \leq SA < \pi/2$.
This program can be used only for the slat angle inclined towards outside.

6. Program List

```
      SUBROUTINE FSB(W,S,SA,A,B,C,FS,FLB)
c     This is a program to calculate the form factor subtended
c     by a window with horizontal louvers.
      DATA P4/0.7853982/
      AT=ATAN(B/C)
      AA=W*SIN(SA)/S
      BB=W*COS(SA)/S
      AAL=ATAN(1./BB-AA/BB)
      UV=UHV(A,B,C,-1)
      U0=0.5*(1.-COS(AT))
      UV0=UV/U0
      IF(AAL.GT.0.)THEN
                AZ=AT
                IF(AAL.LT.AT)AZ=AAL
                FS=0.5*UV0*((1.-AA)*(1.-COS(AZ))+BB*SIN(AZ)
     $             -BB*ALOG(TAN(P4+0.5*AZ)))
                FLB=UV-FS
      ELSE
                FS=0.
                FLB=UV
      ENDIF
      RETURN
      END
```

Fig. 2 An example of component program. This program calculates the form factor subtended by a window with horizontal louvers.

This program FSB uses a function program called UHV, which can calculate the form factor subtended by a rectangular surface. The function program UHV is also a component program of the library.

CURRENT STATE OF THE COMPONENT PROGRAM LIBRARY

Table 1 shows the current numbers of component programs in several categories related to the

Table 1
Current State of the Component Program Library

Category	No. of Programs	Examples
Outdoors	12	SOLR0: extraterrestrial solar radiation SOPOS: solar position EFL: luminous efficacy of daylight EFR: effective sky radiation ⋮
Building Envelopes	67	RF: response factors of walls WALLFD: finite difference calculations of heat transmission through walls RFFUR: response factorsof furnishings GLASS: solar optical properties of glass panes ⋮
Indoor Environment	9	FHM: form factors between a man and walls PMV: calculation of PMV NEWET: calculation of New Effective Temperature ⋮
HVAC systems	36	ACONC1: calculation of cooling coils ACONH: calculation of heating coils THEX: efficiency of heat recovery units ⋮
Miscel-laneous	58	CVP: convective heat transfer coefficients of walls PSY: psychrometrics ⋮
Total	182	

built environment. In the category of building envelopes, there have been fairly large number of programs developed so far; on the other hand, in the category of indoor environment, only nine programs have been developed. There has been no component program developed on some of the items shown in Fig. 1. Since the current availability of component programs is not yet sufficient to make it possible to fully assist researchers and practitioners in designing their own simulation programs, further development is required to get closer to the ideal state of the component program library.

TREE-LIKE STRUCTURE OF COMPONENT PROGRAM LIBRARY

Many programs can be classified in a tree-like structure. The relation between one component program as a stem and others as branches can be found in several ways.

Let us suppose a very simple case of calculating the form factor subtended by multiple windows as shown in Fig. 3 (a). Although it is possible to construct a program, which can directly calculate this form factor, the program would quickly become too involved, because of the complicated geometry of the multiple windows.

Instead, it is better to first construct a few programs to calculate the form factors for a couple

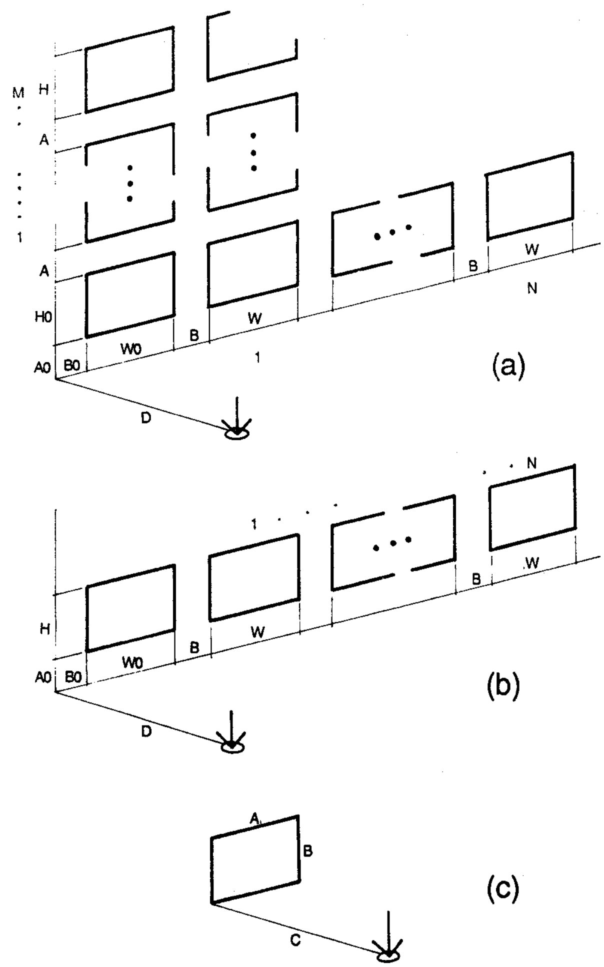


Fig. 3 Geometry of multiple windows (a). These multiple windows can be regarded as a combination of fundamental geometries of the windows (b) and (c).

of fundamental geometries and then combine these for calculating the final form factor.

Figure 4 shows an example of this step. A function program UHV can only calculate the form factor subtended by a single rectangular window shown in Fig. 3 (c). However, a subroutine program UHVM can calculate the form factor subtended by successive rectangular windows horizontally located shown in Fig. 3 (b). Finally, a subroutine program UHVMM can calculate the form factor subtended by the multiple windows shown in Fig. 3 (a).

We call the relationships between these three

```
      SUBROUTINE UHVMM(H,W,D,A0,B0,H0,W0,A,B,M,N,IC,F)
c     This is a program to calculate the form factor
c     subtended by multiple windows.
      CALL UHVM(H0,W,D,A0,B0,W0,B,N,IC,F)
      AA0=A0+H0+A
      DO 10 I=1,M
      CALL UHVM(H,W,D,AA0,B0,W0,B,N,IC,FF)
      F=F+FF
      AA0=AA0+H+A
   10 CONTINUE
      RETURN
      END


      SUBROUTINE UHVM(H,W,D,A0,B0,W0,B,N,IC,F)
c     This is a program to calculate the form factor
c     subtended by succesive multiple windows
c     horizontally located.
      F=UHV(W0+B0,H+A0,D,IC)
      IF(A0.EQ.0..AND.B0.EQ.0.)GO TO 10
      IF(A0.EQ.0.)GO TO 11
      IF(B0.EQ.0.)GO TO 12
      F=F-UHV(W0+B0,A0,D,IC)-
     *  UHV(B0,H+A0,D,IC)+UHV(B0,A0,D,IC)
      GO TO 10
   11 F=F-UHV(B0,H,D,IC)
      GO TO 10
   12 F=F-UHV(W0,A0,D,IC)
   10 IF(N.EQ.0)RETURN
      BB=B0+W0+B
      IF(A0.EQ.0.)GO TO 13
      DO 14 I=1,N
      F=F+UHV(BB+W,H+A0,D,IC)-UHV(BB,H+A0,D,IC)-
     *  UHV(BB+W,A0,D,IC)+
     *  UHV(BB,A0,D,IC)
      BB=BB+W+B
   14 CONTINUE
      RETURN
   13 DO 15 I=1,N
      F=F+UHV(BB+W,H,D,IC)-UHV(BB,H,D,IC)
      BB=BB+W+B
   15 CONTINUE
      RETURN
      END


      FUNCTION UHV(A,B,C,IC)
c     Form factor subtended by a rectangular surface
      DATA PAI/3.141593/
      R1=SQRT(B*B+C*C)
      IF(IC)10,10,11
   10 UHV=(ATAN(A/C)-C*ATAN(A/R1)/R1)*0.5/PAI
      GO TO 12
   11 R2=SQRT(A*A+C*C)
      UHV=(A*ATAN(B/R2)/R2+B*ATAN(A/R1)/R1)*0.5/PAI
   12 RETURN
      END
```

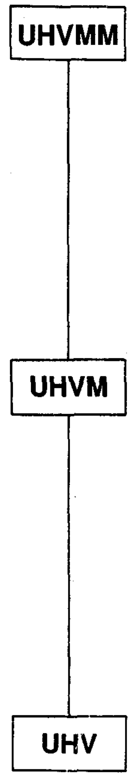


Fig. 4 An example of a tree-like structure of component programs. The program UHVMM uses the program UHVM and then the UHVM uses the UHV.

component programs UHV, UHVM and UHVMM shown in Fig. 4 as being a tree-like structure. A number of examples similar to this can be found in the items involved in the built environment.

As described above, researchers and practitioners may require different inputs and outputs for the same calculation. To overcome this problem, we may prepare two programs: one solves a fundamental problem and the other deals with simplified inputs and outputs, which suit the requirements of the practitioners. These two programs can also be classified within parts of the tree-like structure: the former as a stem and the latter as a branch.

Another feature, which seems very important, is that users can extend the content of the component program library, adding the component programs that they develop for their own purposes.

FUTURE PERSPECTIVES

There are several hurdles we have to jump over in order to make this component program library fully developed.

In general, any building energy simulation program requires a lot of inputs, from the geometry of a room or building to the schedules of building usage. Requirements in inputs differ as we go from residential buildings to office buildings with atria. Since preparing the inputs of a currently available building simulation program is usually time-consuming, it will be necessary to develop component programs, which can deal with inputs. The same is true for outputs.

It would also be necessary to develop a method for assembling the component programs to efficiently construct a simulation program, which exactly meets one's purpose, and to develop a method for validating the simulation program, even though each component program being used has been validated.

SUMMARY

The concept of developing a component program library for making building energy simulations flexible is described. Since many items, from the solar position to a HVAC system component, are involved in the built environment, we have been developing component programs, which make a calculation with respect to each of those items. We have developed 182 component programs for building energy simulations so far. During the course of development, we found that many of the component programs can be classified in a tree-like structure. This tree-like structure suggests that a component program library may have a great potential for constructing a simulation program to conveniently and accurately model a room or building.

ACKNOWLEDGMENT

We would like to express our sincere thanks to the members of the Component Program Library

Working Group of the Society of Heating, Air-Conditioning and Sanitary Engineers of Japan for their contribution to developing component programs. Our thanks are also due to Mr. H. Hisatomi of the Society for his devotion to the Working Group as its secretary.

REFERENCES

1. Kusuda, T. NBSLD, The computer program for Heating and Cooling Loads in Buildings, Building Science Series 69, National Bureau of Standards, Washington, D.C., 1976.

2. Lawrence Berkeley Laboratory and Los Alamos Scientific Laboratory, DOE-2.1B Reference Manual, Parts 1 and 2, 1981.

3. Hittle, D. C., The Building Loads Analysis and Systems Thermodynamics (BLAST) Program, Version 2.0, Report CERL-TR-E-153, U.S. Army Construction Engineering Research Laboratory (CERL), 1979.

4. Matsuo, Y.; Yokoyama, K.; Ishino H.; and Kawamoto S., Introduction to Dynamic Calculation of Air-Conditioning Load in Buildings, Japanese Institute for Building Services Engineers, 1980 (in Japanese).

5. HVAC System Simulation Standardization Committee, HASP/ACSS/8506---User's Manual, Japanese Institute for Building Services Engineers, 1985 (in Japanese).

6. Inoue, U.; Ishino, H.; and Kohri, K., Computer analysis of low temperature heating systems in residential buildings, Transactions of the Society of Heating, Air-Conditioning and Sanitary Engineers of Japan, No.38, 1988, pp.45-53 (in Japanese).

7. Shukuya, M. and Matsunawa, K., A nomograph for estimating annual cooling and heating energy requirements in buildings dominated by internal loads, Energy and Buildings, Elsevia Sequoia, Vol.12, No.3, 1988, pp.207-218.

8. Shukuya, M. and Saito, M., Simplified simulation of indoor temperature and humidity using control volume heat and moisture balance method, Proc. Annual Meeting, Architectural Institute of Japan, 1987, pp.921-922 (in Japanese).

9. Selkowitz, S., Thermal performance of insulating window systems, ASHRAE Transactions, Vol.85 Part 2, 1979, pp.669-685.

10. Bazjanac, V. and Winkelmann, F., Daylighting design for the Pacific Museum of Flight: energy impacts, LBL-Report-23617, September 1988 (to be published in Technical Proceedings of 1986-International Daylighting Conference)

11. Numerical Algorithm Group, The NAG Fortran Library Manual, The Numerical Algorithm Group Ltd., 1987.

12. Hitachi Co. Ltd., Subprogram Library for Numerical Calculations, 1979 (in Japanese).

13. IMSL, Inc., MATH/LIBRARY--- FORTRAN Subroutines for Mathematical Applications---User's Manual, 1987.

14. SPSS Inc., SPSS User's Guide 2nd Edition, McGraw-Hill , 1986.

15. IMSL, Inc., STAT/LIBRARY--- FORTRAN Subroutines for Statistical Analysis---User's Manual, 1987.

16. Computer Associates, CD-DISSPLA User's Manual Ver.11 , 1988.

17. Microsoft Corporation, Microsoft FORTRAN Optimizing Compiler for MS-DOS Operating System ---User's Guide and Language Reference, 1987.

18. Shukuya, M. and Kimura, K., Calculation of the work plane illuminance due to daylight including the effect of direct sunlight through windows with horizontal or vertical louvers, CIE 20th Session, 1983, paper D304.

19. Shukuya, M. and Ohta, M., Calculation of solar heat gain through windows with shading devices and its effect on occupant's thermal comfort, Proc. Annual Meeting, Architectural Institute of Japan, 1986, pp.763-764 (in Japanese).