

# **THE CONCEPT OF THE CALCULUS OF FIRE SAFETY**

**Richard L. Smith**

**U.S. DEPARTMENT OF COMMERCE  
National Institute of Standards  
and Technology  
National Engineering Laboratory  
Center for Fire Research  
Galthersburg, MD 20899**

**U.S. DEPARTMENT OF COMMERCE  
Robert A. Mosbacher, Secretary  
NATIONAL INSTITUTE OF STANDARDS  
AND TECHNOLOGY  
John W. Lyons, Director**

**NIST**



# **THE CONCEPT OF THE CALCULUS OF FIRE SAFETY**

**Richard L. Smith**

**U.S. DEPARTMENT OF COMMERCE  
National Institute of Standards  
and Technology  
National Engineering Laboratory  
Center for Fire Research  
Gaithersburg, MD 20899**

**November 1990**



**U.S. DEPARTMENT OF COMMERCE  
Robert A. Mosbacher, Secretary  
NATIONAL INSTITUTE OF STANDARDS  
AND TECHNOLOGY  
John W. Lyons, Director**



## Table of Contents

Introduction . . . . .	1
Basic Fire Safety Problem . . . . .	3
The Calculus of Fire Safety . . . . .	5
Examples from the ASKBUDJr Problem . . . . .	7
General Examples . . . . .	8
Levels of abstractions . . . . .	9
Conclusions and . . . . .	9
References . . . . .	11
Appendix A: The Generations of Languages . . . . .	15
Appendix B: The AskBudJr Problem . . . . .	16

## Abstract

Many people (builders, owners, designers, occupants, etc.) have an interest in appraising the fire safety of existing or proposed buildings. Computers are playing an ever increasing role in fire safety analysis and in the technology transfer of fire science. In this report the concept of the Calculus of Fire Safety is developed as a declarative programming language. This calculus will enable users to specify what fire safety question is to be answered without specifying how the answer is to be obtained. Advances in Artificial Intelligence programming techniques will enable the implementation of the Calculus of Fire Safety on widely available workstations. This will provide a very powerful tool to anyone interested in determining the fire safety of buildings.

Key Words: Artificial intelligence, expert systems, computer programs, Fire Safety, Fire Risk

## Introduction

The purpose of this report is to describe the concept of the Calculus of Fire Safety (CFS) and its potential significance. What is presented is not intended to be the final form of the CFS, but just an illustration of the concept of this calculus. Also, no effort will be made to describe how this calculus will be implemented or realized.

One definition for calculus is "A method of analysis or calculation using a special symbolic notation." [1]. A Calculus of Fire Safety is, therefore, a method of reasoning, analyzing, or calculating fire safety questions using a special symbolic notation. Possibly the most powerful realization of the CFS will be its implementation in software for computers. However, it will also serve as powerful tool for persons to think about fire safety problems and for communication about such problems.

Computers and computer programs are playing an increasingly important role in moving modern fire science out of the laboratory into the main stream of fire protection. The use of computers promise to greatly improve the process of technology transfer and utilization [2]. Advances in Artificial Intelligence should enable the development of expert-system programs that can enable one to comply more readily with building codes [3] [4] [5]. Current examples of conventional programs that have the potential of having a significant impact on fire safety are FPETOOL [6] and Hazard I [7].

FPETOOL is a program containing a collection of engineering equations and models that are useful in estimating potential fire hazard and the response of the space and fire protection systems to the developing hazard. It is a compiled Basic program. The principal users of this program are expected to be practicing fire protection engineers. The author of FPETOOL recommends that the users understand the basic principles being applied by the program so that they can decide on the degree of confidence to be given to the results obtained. The user does not write computer code to operate this software. The individual procedures are called from a common menu and menus indicate the required inputs. The output is numerical and has to be interpreted by the user. This interpretation requires considerable skill in fire protection technology.

HAZARD I is a program that can be used for fire hazard assessment as it relates to human incapacitation. It is written in Basic and FORTRAN. Menus are used to obtain the input from users. It is recommended by the authors that the user needs to have extensive familiarity with fire science and engineering and all types of related data to successfully use the program. Its output requires significant skill in fire technology to correctly interpret the

results. Also, as with FPETOOL, the user does not write computer code in a programming language to operate this software.

Computer languages can be described as covering a continuous spectrum from purely imperative at one extreme to purely declarative at the other extreme. We will use the terminology of Petre and Winder [8] to classify programming languages. Imperative languages express sequences of operations required to achieve a calculation. On the other hand, declarative languages focus upon what is to be calculated instead of how the calculation is to be done. How the calculation is accomplished depends on the implementation of the declarative language. The algorithms to do the calculation are contained in its implementation. One great advantage of a declarative language is that it can be used to minimize the distance in the user's mind between the expression of the problem and the input required by the software. This is accomplished by shifting the emphasis of programming from prescription of operations to be performed, to definition of the result to be computed.

Familiar languages can be placed on this spectrum. Going from the imperative end of the language spectrum to the declarative end there are the languages FORTRAN, C, LISP, and PROLOG. FORTRAN is about as close to an imperative language as a high level language can get. Although PROLOG is a significant distance from the declarative end of the spectrum, it is closer to the declarative end of the spectrum than any of the others listed.

Computer languages can also be described as belonging to a particular generation of languages (See Appendix A). For example FORTRAN is a third generation language and PROLOG can be considered as an embryonic fifth generation language.

One function of a computer language is to make a computer perform certain operations. However, it can also be a formal medium for representing ideas about some particular domain [10]. It can serve as a framework upon which one can organize or represent ideas. Modern computer science provides facilities for controlling complexity, such as found in many fire problems, by building abstractions that hide details when appropriate. Complex programs can be constructed by building computational objects of increasing complexity. In decomposing a complex problem it is critical that each subprocedure performs a specific task that can be used as a module in defining other procedures. A well designed program is deigned in a modular manner, so that the modules can be developed, replaced, and debugged independently.

High level programming languages or third generation languages are computer languages which are designed to fit particular problems domains such as FORTRAN for numerical computation. It is believed that one can improve on high level programming languages by developing a higher level language that provides a higher



degree of expressive power within the subject domain [11] [13]. In any domain there are likely to be a collection of underlying facilities for many programs in the domain [12]. Specifying and implementing those underlying facilities is an important, distinguishable part of the task of program development. This specifying and implementing can constitute the development of a language of a newer generation. This newer generation language may have a great deal of domain specific information built into it and it may be of limited scope. However, its limited scope can be more than compensated for by power to efficiently and effectively solve problems within the particular domain.

CFS can be viewed as a limited scope fifth generation language designed to address the problem of fire safety. A typical person interested in fire safety should be able to write programs that would run interactively using this language without any significant formal training in CFS and the results should be in a form which he can readily use. The user would be assured that the program would always use the most appropriate fire technology to solve the problem. In addition, CFS should greatly assist one to: organize one's thinking, discussions, and communications.

The next section will describe the basic fire safety problems. It is followed by a discussion of the CFS at the highest level of abstraction. Then in the following section some examples for the idealized simplistic ASKBUDJr problem domain (See Appendix B) will be discussed. Next, there is a discussion of various level of abstractions of CFS. Finally, the conclusions and recommendations for future work will be presented.

### Basic Fire Safety Problem

There are a large number of people who have similar questions about the fire safety of buildings or items in buildings. One category of questions relates to safety, another relates to whether one has complied with the fire safety laws or what liability one is subjecting oneself to. We will limit ourself to only questions that relate to safety. That is, whether there is a reasonable expectation that an unwanted fire will cause losses due to physical damage.

From one perspective, fire is a simple phenomena. A fire starts. It grows or burns until it goes out. As a consequence of the burning there are losses in the form of damages to inanimate or animate objects. The inanimate objects are normally the parts of the building or its contents. The animate objects of most interest are normally humans. The expected losses are highly dependent upon the attributes of the particular objects involved in the fire.

Let us try to formulate a precise form for a typical fire safety question that a building owner/occupant might ask. But first let us look at some imprecise questions. Some imprecise questions are: What are the

- A. expected total losses per year,
- B. expected loss of life per year,
- C. expected injuries per year,
- D. expected structural losses,
- E. expected content losses, and
- F. expected loss due to inability to use the building?

A precisely formulated question would need to contain additional information, for example:

What is the expected loss of life during the next five years in the apartment building at 77 Wester Vista Ave, Rockville, MD if the present economic class of occupants remains the same, the present level of fire department service remains unchanged, the neighborhood does not change in any significant manner, ...

Let us examine the major components of this question. The first major part is "What is the expected loss," which characterizes the general nature of the question. The second major part "life" actually contains two pieces of information: (1) the degree of damage, i.e. death; and (2) the object damaged, i.e., any human associated with the property of interest. The third major part is the time interval of interest. The final major part is a specification of the exact property of interest and any other factors that might have a bearing on the loss of life associated with this property. Therefore, we have identified the following parts of a precisely formulated question on fire safety:

- 1. General nature of the question,
- 2. Degree of damage,
- 3. Object(s) damaged,
- 4. Time interval of interest, and
- 5. The context of the question.

Another category of questions involves probability. Examples of these are:

- A. What is the probability of a fire causing me injury if I spend the night in this hotel?

B. What is the probability that a particular item will be damaged by fire in the next six weeks?, next five years?

An example of a precisely formulated question of probability is:

What is the probability of the loss of life as a function of time during the next five years in the apartment building at 77 Wester Vista Ave, Rockville, MD if the present economic class of occupants remains the same, the present level of fire department service remains unchanged, the neighborhood does not change in any significant manner, ...

We note that while the general nature of the question has changed, the parts of the question have remained basically the same.

In order to combine or compare different losses, we need a common metric for the measuring of the degree of damage for the different types of objects. One possible metric is dollars, but the particular one chosen does not concern us now. The fire losses could be reportable in various terms such as dollar loss, lives lost, serious injuries, minor injuries, room loss, work station loss, floor, etc. If all losses were converted into a dollar equivalent, then it would be possible to select the most cost efficient fire protection system to minimize the sum of fire losses and cost of fire safety.

### The Calculus of Fire Safety

We have, from the above discussion, two general questions we wish to answer. The first one relates to expected values and the other to the probability of some loss. We want a representation that will be readily understandable by the typical user, but also consistent with being able to stimulate a computer to do the right thing. We will not consider free form English as possible input. However, we could consider the following form as a viable candidate:

CFS><sup>1</sup> Expected-loss of-degree for-object in-time-interval with-context

In this expression "Expected-loss-of" identifies the type of question being asked. The remaining parts of this expression can

---

<sup>1</sup> Placing the expression CFS> in the left hand margin indicates that what follows is what is typed into the computer which has an implementation of the CFS running on it.

be considered as variables. Some typical values for each of these variables are:

of-degree	death, 2nd degree burns over 50% of the body, slight, serious, total loss
for-object	any occupant, joe, building, room 27, main frame computer
in-time-interval	next year, next five years, life of building
with-context	specification of property, environment, and any thing that impacts the expected losses

An examples of how this would appear for particular question is:

CFS> What is the expected loss of life in the next 5 years in the author's office?

=><sup>2</sup> The expected loss of life in the author's office during the next 5 years is 0.00001 lives.

For questions dealing with probability, we could write the general form as:

CFS> Probability-of-loss of-degree for-object at-time with-context.

In this expression "Probability-of-loss" identifies the type of question being asked. The remaining terms are variable which are the same as the ones above expected for the one referring to time. This variable "at-time" has as its value the instantaneous time.

An examples of how this would appear for particular question is:

CFS> What is the probability of the loss of life at noon of June 25, 1992 in the author's office?

=> The probability of the loss of life in the author's office at noon of June 25, 1992 in the author's office is  $6.34 \times 10^{-14}$ .

We see that both these questions divide into two major parts and the object of concern. One major part identifies the question and

---

<sup>2</sup> This indicates the response of the program to a question or input.

its specifications. The other is the context. We note that there is some duplication in the information in the item "for-object" and the context because the "for-object" is also in the context. If it is not, the answers are easy to come by.

In a more compact notation we could write<sup>3</sup>

```
CFS> Qi Oi Ci
```

or

```
CFS> (Qi Oi Ci)
```

or

```
CFS> {Question-i Object-i Context-i}.
```

Where Qi and Question-i stands for a particular question with its variables specified, Oi and Object-i the object the question is concerned with ("for-object"), and Ci and Context-i stand for the corresponding context. We will try to make this clearer by using examples from ASKBUDJr.

#### Examples from the ASKBUDJr Problem

The problem treated by the expert system ASKBUDJr is described in Appendix B. We will look at possible questions users of this program might ask.

First, as we have with all problem domains in fire safety, there are two primary questions. These are listed in Table 1 with possible values for the other quantities. Examples of context are given in Tables 2 and 3. One can see how potentially complex the context can become and how a seemingly little change can create a new context. From these tables we can select combinations of a question, variables, objects, and context to form precise CFS statements. Examples are:

```
CFS> Expected-loss death occupant next-five-years context1
```

or

```
CFS> What is the expected loss of life during the next five years  
in context1.
```

```
CFS> Expected-loss total chair next-12-months context2
```

or

---

<sup>3</sup> This compact notation was selected by the author primarily because of the limitations of the software used to write this report. If another program was used, a more expressive representation might have been used.

CFS> What is the expected loss due to the total loss of the chair during the next 12 months if the chair is in context2.

CFS> Probability-of-loss third-degree-burn-over-60%-of-the-body occupant time context1

CFS> Probability-of-loss slight bed life-of-building context2

The program responses are not given because they would be similar in form to the responses give previously.

## General Examples

We now turn from the application of the CFS to a particular class of contexts of ASKBUDJr to how it might be developed in general. Let us choose a particular question with appropriate variables Q1 and let focus on one object O1. Then we can write for contexts C1 and C2

CFS> Q1 O1 C1

CFS> Q1 O1 C2

It is reasonable to suppose that there will be occasions when one would want to determine which had the greater value or by how much do they differ. We could express this as

CFS> Greater? (Q1 O1 C1) (Q1 O1 C2)

This expression can be read as saying "Is (Q1 O1 C1) greater than (Q1 O1 C2)?" Similarly we can write for the difference

CFS> Difference (Q1 O1 C1) (Q1 O1 C2)

or

CFS> (Q1 O1 C1) - (Q1 O1 C2)

This leads naturally to the question as to what are the differences in the contexts. This could be expressed as

CFS> Difference C1 C2

The response would be to identify the differences between the two contexts (e.g. the door in C1 is open and closed in C2, The floor is concrete in C1 and steel in C2 etc.). Sometimes there may be numerous differences and most of them are not significant for the question Q1 and object O1. In these cases it would be desirable to have the ability to ask for only significant differences. This could be written as

CFS> Significant-Difference Q1 O1 C1 C2

The response to this would be to list only the differences that were significant.

There is much more we could say about the general usage of the CFS, but that will be left to later reports.

### Levels of abstractions

We have been focusing on the highest level of abstraction of the CFS so far in this report. As mentioned earlier, in the analyses of a problem in the creation of a program, one starts at a very high level of abstraction and works systematically to lower and lower level of abstraction. The CFS would explicitly contain these levels. They will be accessible by the user as desired. At the highest level of abstraction, the top level, the CFS would appear as described above. At lower levels there would be such questions as:

CFS> What is the criteria for damage that was met resulting in the death of the occupant in room 26 in context9?

CFS> What was the value of the failing parameter of the last question?

CFS> What is the criteria that was not met but is close to being met?

CFS> Can room 99 flashover?

The reader is cautioned not to take the above examples of lower level question content or form too seriously. Significant work will have to be accomplished before these lower level questions can be properly and precisely formulated.

The development of the appropriate data and procedure abstraction will provide a flexible structure into which all fire data and fire process computational models can readily fit. This would expedite the use of new data and fire models. It also would make explicit the entire logical structure of the analysis of fire safety questions.

### Conclusions and Expectations

This report demonstrated a method of reasoning or analyzing fire safety questions using a special symbolic notation, the CFS. It

allows a person interested in fire safety to use the most appropriate fire technology to solve a fire safety problem and obtain results he can understand. It has also been demonstrated that the CFS will greatly assist the user to organize his thinking, discussions, and communications and allow him to write programs directly so they can run interactively. Productivity gains should be comparable or greater than those of forth generation languages compared to third generation languages (See Appendix A). It is hard to include in productivity gains the impact of CFS of allowing people who cannot now program to write programs that will solve important complex fire safety problems.

The interfaces for such programs as FPETOOL and HAZARD I probably contain some of the abstraction that would go into the CFS. However, this abstraction in these programs is implicit. Furthermore, the interface is inflexible in the sense it is customized for a particular limited fire problem set. On the other hand CFS holds the promise of being a flexible language that can address any fire safety question. Plus it can make explicit the entire logical structure for the analysis of fire safety questions.

We have shown how the use of the CFS enables the precise statement of top level questions relating to fire safety. While we have shown how to state precisely some possible top level questions, we have not shown how to obtain answers. In the final analysis, it is the worth of the answers that will determine the true value of the CFS.

The next step is to show how CFS can be implemented in a simple case. One possibility is to develop a CFS for the ASKBUDJr problem. The development of the CFS will progress from the top level to lower levels. Knowledge will be collected as we move from level to level. At first, the unrealistic problem of ASKBUDJr will be the problem addressed. After that problem has been completed, the problem definition and solution will be improved until we are dealing with a significant real world problem.

The implementation of CFS will show it to be a dynamic mechanism into which data and fire models can be placed to be used, tested, or evaluated. CFS will be able to incorporate models regardless of type, i.e., physical models, correlation models, or empirical models. If we can obtain the knowledge of domain experts to fill the system with knowledge, the calculus will provide answers comparable or superior to an expert fire protection engineer.

The development of the CFS would provide a means to allow the typical building owner/occupant and fire safety professionals to apply the best fire safety knowledge to fire safety problems. This would result in the cost of fire losses and fire prevention being greatly reduced.



## References

- [1] The American Heritage Dictionary of the English Language, Editor W. Morris, American Heritage Publishing Co. 1969
- [2] Wright, Richard N.; Lyons, John W. Machine Representation of Standards. ASTM Standardization News pp 44 - 48; August 1986
- [3] Rosenman, Michael. A.; Gero, John. S. Design Codes as Expert Systems. Computer Aided Design, vol 17, no. 9 November 1985 pp 481-494
- [4] Smith, Richard L. EXPOSURE: An Expert System Fire Code. Natl. Inst. of Stand. and Technol. NISTIR-4373; 1990
- [5] Buis, M., Hamer, J., Hosking, J. G., and Mugridge, W. B. An Expert Advisory System for a Fire Safety Code. in "Applications of Expert Systems Edited by J. R. Quilan, Turing Institute Press, 1987
- [6] H. Nelson "FPETOOL: Fire Protection Tools for Hazard Estimation" Natl. Inst. of Stand. and Technol. NISTIR-4380; 1990
- [7] Peacock, R. D., and Bukowski, R. W., A Prototype Methodology for Fire Hazard Analysis. Fire Technology vol 26, No.1 p15; February 1990
- [8] Marian Petre and R. Winder "On Languages, Models and Programming Styles" The Computer Journal nol 33, no. 2, 1990, p173
- [9] Smith, Richard L. ASKBUDJr: A Precursor of an Expert System for the Evaluation of Fire Hazard. Fire Technology Vol. 23, No. 1, p. 5; February 1987
- [10] Abelson, H. and Sussman, G.J. with J. Sussman "Structure and Interpretation of Computer Programs" The MIT Press 1986
- [11] Hawely, R. Editor "Artificial Intelligence Programming Environments" Ellis Horwood Limited 1987
- [12] Ramsay, A. Embedding very high level languages, p61 in [11]
- [13] Martin, James "Fourth-Generation Languages" Prentice-Hall, Inc. 1985

Questions

Expected-loss

Probability-of-loss

of-degree

death

third degree burns over

60% of the body

slight, serious, or total

in-time-interval

the next 12 months

next five years

life of building

for-object

occupant

bed, chair, chest, table,

wastebasket, curtains &

drapes

at-time

time

Table 1. Questions, variables, and objects for ASKBUDJr

furnishing	A peak	B h-of-comb	C LC50	D growth-rate	E will-ignite
1. Bed	1000	35	20	FAST	NIL
2. Chair	500	20	20	MODERATE	(TABLE CURTAINS-DRAPES)
3. Table	500	18	60	SLOW	(CHAIR)
4. Chest	1000	25	40	SLOW	NIL
5. Wastebasket	50	20	60	VERY-FAST	(BED)
6. Curtains	100	25	15	VERY-FAST	(CHAIR)

8. ROOM - dimensions in feet

A detector	B length	C width	D height	E door-height	F door-width
N	15'	12'	8'	7'	2.5'
G fraction-open	H window-width	I window-height	J in-big-bldg		
1	2.5'	2'	Y		

7. Occupant

A awake	B mobile	C drunk	D heart-lung	E external-aid	F time-to-es
N	Y	N	N	N	8.23s
G rescue-time					
1.0F+05s					

Table 2. First example of context for the ASKBUDJr problem.

furnishing	A peak	B h-of-comb	C LC50	D growth-rate	E will-ignite
1. Bed	1000	35	20	FAST	(TABLE CURTAINS-DRAPES)
2. Chair	500	20	20	MODERATE	NIL
3. Table	500	18	60	SLOW	(CHAIR)
4. Chest	1000	25	40	SLOW	NIL
5. Wastebasket	50	20	60	VERY-FAST	(BED)
6. Curtains	100	25	15	VERY-FAST	(CHAIR)

8. ROOM - dimensions in feet

A detector	B length	C width	D height	E door-height	F door-width
N	15'	12'	8'	7'	2.5'
G fraction-open	H window-width	I window-height	J in-big-bldg		
1	2.5'	2'	Y		

7. Occupant

A awake	B mobile	C drunk	D heart-lung	E external-aid	F time-to-es
N	Y	N	N	N	8.23s
G rescue-time					
1.0F+05s					

Table 3. Second example of context for the ASKBUDJr problem.

## Appendix A: The Generations of Languages

Computer languages can be classified into five generations [13].

### First generation:

Machine language which is a binary notation such as  
011010000111010100111100  
is the language first used to program computers.

### Second generation:

Symbolic assembly language used symbolic addresses rather than physical machine addresses. Examples of languages are SAP, AUTOCODER, and EASYCODER. This generation started in the mid-1950s.

### Third generation:

This generation is also called high-level languages and started in the 1960s. Examples languages are ALGOL, FORTRAN, COBOL, BASIC and LISP. The languages were the first ones that showed significant independence from the hardware. By using English words and mathematical notation they started moving towards the language of the user.

### Fourth generation:

This generation languages (4GL) are sometimes called high-productivity languages since they are meant to offer significant improvements over the third generation languages. Improvements sought were more rapid application-building process, reduced maintenance cost, minimize debugging, and to increase the ability of the end user to write his own code. Examples are FOCUS, RAMIS, NATURAL, ADS, and UFO.

Typical productivity gains of using fourth generation languages over third generation are 10 to 1 gain. Some studies have reported gains in the range of from 17 to 1 to 45 to 1.

### Fifth generation:

This generation is not well defined yet. However, it will utilize developments from Artificial Intelligence such as knowledge-based systems, expert systems, inference engines, and natural language understanding. PROLOG can be considered as an embryonic language of this generation. Again the motivation for the creation of a new generation of languages is to increase the usefulness of computers.

## Appendix B: The AskBudJr Problem [9]

The program ASKBUDJr estimates the degree of hazard an occupant is exposed to while occupying a single room, e.g. a bedroom, a motel room, or a hospital room. It is assumed that:

- a fire starts in the room,
- there is only one occupant,
- the structure of the building does not become involved in the fire,
- the room has one window and one door,
- the room has the following furnishings: a bed, a chest, a chair, a table, a wastebasket, and a set of curtains or drapes.

In ASKBUDJr, the degree of hazard the occupant is exposed to depends upon how fast the hazard from the fire builds up versus how fast he can evacuate the room. Of all the many things that could influence these times, only a few are considered. However, there is enough to identify some interesting problems.

The user of ASKBUDJr will be asked to enter or to accept the default value for the following for each item of the room's furnishings:

- the peak burning rate in kW
- the effective heat of combustion,  $DH_c$ , in kJ/g
- LC50, the amount of burned material per unit volume that will kill 50% of a sample of rats in 30 minutes using a standard test procedure in mg/L or  $g/m^3$
- the growth rate of the burning rate as one of four t-squared curves, i.e., slow, moderate, fast, or very fast (or zero if it doesn't burn)
- what other items will be ignited if the one in question burns.

The t-squared burning curves mentioned above are defined by how long it takes the burning curve to reach 1MW for the thermal power being released. The growth rates are defined as follows:

- a slow growing curve leads to 1MW in 600 sec.
- a moderate growing curve leads to 1MW in 300 sec.
- a fast growing curve leads to 1MW in 150 sec.
- a very fast growing curve leads to 1MW in 75 sec.

The user will also either enter or accept the default values for the sizes of the room (length, width, and height), the size of

the door and window openings, and whether there is a fire detector/alarm or not.

Finally the user inputs whether the occupant is: awake, asleep, drunk, mobile, or nonmobile; the time the occupant would take to move out of the room after becoming aware of the fire; and whether the occupant has a heart or lung condition. Also the user will input whether there is a person outside the room who can aid the occupant and how long it would take this person to come in and remove the occupant after becoming aware of the fire.





1. PUBLICATION OR REPORT NUMBER NISTIR 4459
2. PERFORMING ORGANIZATION REPORT NUMBER
3. PUBLICATION DATE November 1990

# BIBLIOGRAPHIC DATA SHEET

4. TITLE AND SUBTITLE  
The Concept of the Calculus of Fire Safety

5. AUTHOR(S)  
Richard L. Smith

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS) U.S. DEPARTMENT OF COMMERCE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY GAITHERSBURG, MD 20899	7. CONTRACT/GRANT NUMBER
8. TYPE OF REPORT AND PERIOD COVERED	

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

Many people (builders, owners, designers, occupants, etc.) have an interest in appraising the fire safety of existing or proposed buildings. Computers are playing an ever increasing role in fire safety analysis and in the technology transfer of fire science. In this report the concept of the Calculus of Fire Safety is developed as a declarative programming language. This calculus will enable users to specify what fire safety question is to be answered without specifying how the answer is to be obtained. Advances in Artificial Intelligence programming techniques will enable the implementation of the Calculus of Fire Safety on widely available workstations. This will provide a very powerful tool to anyone interested in determining the fire safety of buildings.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)  
Artificial intelligence; expert systems; computer programs; Fire Safety; Fire Risk

13. AVAILABILITY	<input checked="" type="checkbox"/> UNLIMITED	14. NUMBER OF PRINTED PAGES 22
	<input type="checkbox"/> FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).	
	<input type="checkbox"/> ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.	15. PRICE A02
<input checked="" type="checkbox"/> ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.		

