A Validation Method using Fuzzy Simulation in an Object Oriented Physical Modeling Framework

Gyooseok Kim and Paul A. Fishwick

Computer and Information Science and Engineering Department
University of Florida

ABSTRACT

Object Oriented Physical Modeling (OOPM) is an object-oriented methodology for constructing physical models by emphasizing a clear framework to organize the geometry and dynamics of the models. An environment called MOOSE (Multimodeling Object Oriented Simulation Environment) is under development to explore this OOPM concept. MOOSE provides a solid connection between blackboard models and software models in an unambiguous way, capturing both static and dynamic semantics of objects. Even though this facility reinforces the relation of "model" to "program", an adequate validation technique for modeling processes has not yet been developed. In this paper, we propose a validation method for the modeling process in MOOSE. This method utilizes a fuzzy simulation approach to encode uncertainty arisen from human reasoning process into computer simulation components.

Keywords: Physical Modeling, Validation, Fuzzy Simulation

1. INTRODUCTION

MOOSE (Multimodel Object Oriented Simulation Environment)[1,2] is an enabling environment under the development at University of Florida for modeling and simulation based on OOPM. OOPM extends object-oriented program design with visualization and reinforces the relation of "model" to "program". This permits a tight couple between a model author and the modeling and simulation process through an interactive HCI (Human Computer Interface). MOOSE consists of four major components[1]: **Modeler**, **Translator**, **Engine** and **Scenario**. **Modeler** interacts with a model author via a GUI (Graphical User Interface) in a way to help the author make the valid conceptual model of the system. **Translator** is a bridge between a model design and a model execution. It reads the output from **Modeler** and builds the corresponding structures of the conceptual model with C++ code automatically, therefore it ensures that the program is a valid representation of the conceptual model. **Engine** is a C++ program, composed of Translator output plus runtime support, compiled and linked once, then repeatedly activated for Model Execution. **Scenario** is a visualization-enabling GUI which interacts with **Engine**, and displays **Engine**'s output in a meaningful form so that the output of MOOSE can be validated against the author's expertise.

Even though this facility reinforces the relation of "model" to "program" in a natural way, any adequate validation technique for the modeling process, particularly between physical systems and conceptual models, and between conceptual models and computerized (programmed) models has not yet been developed. *Face validation*[4,5] by domain experts is known as one of the validation techniques for the conceptual models. In this paper, we assume that there is knowledge from a domain expert in the form of linguistic *if-then* rules. Our goal is to propose a validation method that performs an automatic consistency checking between the expert's rule-based model and various types of conceptual models in MOOSE.

We organize this paper as follows: in Section 2, we propose the fuzzy set theory which is relevant to this research followed by a comparison between a general modeling process and its counterpart of MOOSE. In Section 3, we propose a fuzzy simulation method that we can employ to validate the conceptual models of MOOSE.

Other author information: (Send correspondence to G. Kim)
G. K.: Email: kgs@cise.ufl.edu; Telephone: 352-392-1435; Fax: 352-392-1414.
P.A.F: Email: fishwick@cise.ufl.edu; Telephone: 352-392-1414; Fax: 352-392-1414

2. BACKGROUND

2.1. Fuzzy Set Theory

This section presents a review of the relevant aspects of fuzzy set theory which forms the basis of our fuzzy simulation. The theory of *fuzzy sets* can be found in Refs. 6–11. Fuzzy sets may be viewed as an attempt to deal with a type of imprecision that arises when the boundaries of classes are not sharply defined. A fuzzy set A of a universe of discourse X is characterized by a membership function $\mu_A : X \to [0, 1]$ which associates with each element x of X a number $\mu_A(x)$ in the interval [0, 1] which represents the grade of membership of x in A.

- *Definition 2.1:* If f is an n-ary crisp function which is a mapping from a Cartesian product $X_1 \times \cdots \times X_n$ to a space Y, and if A is a fuzzy set in $X_1 \times \cdots \times X_n$ which is characterized by a membership function $\mu_A(x_1, \ldots, x_n)$, with $x_i, i = 1, \ldots, n$, denoting a generic point in X_i, then *extension principle* [8] states that

$$\begin{aligned} f(A) &= f(\int_{X_1 \times \cdots \times X_n} \mu_A(x_1, \ldots, x_n)/(x_1, \ldots, x_n)) \\ &= \int_Y \mu_A(x_1, \ldots, x_n)/f(x_1, \ldots, x_n) \end{aligned} \tag{1}$$

  The membership function of A is expressed by

$$\mu_A(x_1, \ldots, x_n) = \mu_{A_1}(x_1) \wedge \mu_{A_2}(x_2) \wedge \cdots \wedge \mu_{A_n}(x_n) \tag{2}$$

  where $\mu_{A_i}, i = 1, \ldots, n$, is the membership function of A_i.

- *Definition 2.2:* Let A and B represent two fuzzy numbers and let $\star$ denote any of the four basic arithmetic operations. Then, using the *extension principle* (1) under the assumption (2), we define a fuzzy set, $A \star B$ on $\mathcal{R}$, where $\mathcal{R}$ is the set of all real numbers, as

$$\mu_{A \star B}(z) = max_{z=x \star y}(\mu_A(x) \wedge \mu_B(y)), \tag{3}$$

  $\forall\ z \in \mathcal{R}$. Thus, for example, if $A, B \subseteq \mathcal{R}$ are two fuzzy numbers with respective membership functions $\mu_A(x)$ and $\mu_B(y)$, then the four basic arithmetic operations, (i.e., addition, subtraction, multiplication and division) give, for each $x, y, z \in \mathcal{R}$, the following results:

$$\mu_{A+B}(z) = max_{z=x+y}(\mu_A(x) \wedge \mu_B(y)). \tag{4}$$

$$\mu_{A-B}(z) = max_{z=x-y}(\mu_A(x) \wedge \mu_B(y)). \tag{5}$$

$$\mu_{A \times B}(z) = max_{z=x \times y}(\mu_A(x) \wedge \mu_B(y)). \tag{6}$$

$$\mu_{A \div B}(z) = max_{z=x \div y}(\mu_A(x) \wedge \mu_B(y)). \tag{7}$$

- *Definition 2.3:* Let P be a compound statement of the type, ($\mathcal{X}$ is A) $*$ ($\mathcal{Y}$ is B), where $\mathcal{X}$ and $\mathcal{Y}$ are fuzzy variables that take real numbers from some universal sets X, Y, respectively, A and B are fuzzy values on X, Y, respectively and $*$ is a conjunction (and) or a disjunction (or).

  - When $*$ is a conjunction, the *rule of conjunctive composition* [9] states that P can be expressed by a possibility distribution $\pi(x, y)$ which is defined by

$$\{\mu_{A \times B}(x, y)/(x, y) \mid x \in X, y \in Y\}, \tag{8}$$

    where $\mu_{A \times B}(x, y)$ denotes $min(\mu_A(x), \mu_B(y))$ and $\times$ is the cartesian product.

  - When $*$ is a disjunction, the *rule of disjunctive composition* [9] states that P can be expressed by a possibility distribution $\pi(x, y)$ which is defined by Equation (8), where $\mu_{A \times B}(x, y)$ denotes $max(\mu_A(x), \mu_B(y))$.

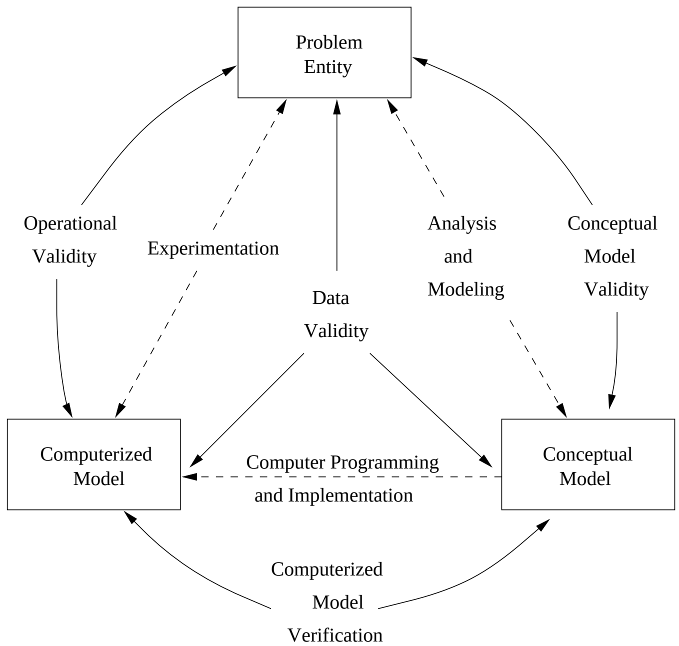


Figure 1. Modeling process and its relation to validation/verification

2.2. Modeling Process

Fig. 1[12,4,13] shows a general modeling process. The conceptual model represents the mathematical, logical or verbal representation of the problem entity developed for a particular study, and the computerized model represents the conceptual model implemented on a computer. The general purpose of the conceptual model validation depicted in this figure is to validate the underlying assumptions and theories. More specifically, the process is concerned with whether this specific model's representation of the problem entity being modeled and its structure, logic and mathematical and causal relationships are *reasonable* for the intended use of the model.[12] One of the primary validation techniques used for this evaluation is *face validation*.[4,5] Face validation involves having domain experts evaluate the conceptual model to determine if they believe it is correct and reasonable for its purpose. This usually means examining the flowchart or graphical model, or the set of model equations.

The counterpart of the above modeling process in MOOSE is depicted in Fig. 2. MOOSE supports many different types of models[1] including CODE, FSM (Finite State Machine), FBM (Functional Block Model), RBM (Rule Based Model) and EQN (EQuatioNal Constraint model) for the conceptual modeling process. Then, by translating the conceptual model into C++ code, it constructs the computerized model. MOOSE does not yet employ validation or verification techniques.

3. A PROPOSED FUZZY SIMULATION METHOD

Using the fuzzy simulation method introduced in this section, the *face validation* process discussed in the previous section can be automated, thereby contributing to validate the conceptual models in MOOSE. A prerequisite for this process is that there should exist an expert's validated rule set associated with appropriate membership functions for the system of interest. Given that this condition is satisfied, the fuzzy simulation method can perform *consistency check* between the conceptual model and computerized model as shown in Fig. 3. In the fuzzy simulation method, every vertex in the fuzzy number is issued independently to the simulation function, and the outputs of the simulation are mapped into the most closely matched fuzzy linguistic value by a linguistic approximation. In this way, we obtain rules from CODE, FSM, FBM, EQN and RBM by applying the fuzzy simulation, and through consistency checking against the expert's rules, we can identify any inconsistency due to an inadequate conceptual model of MOOSE or an improperly programmed or implemented conceptual model on the computer. To make this validation available, we have focused on the following tasks.

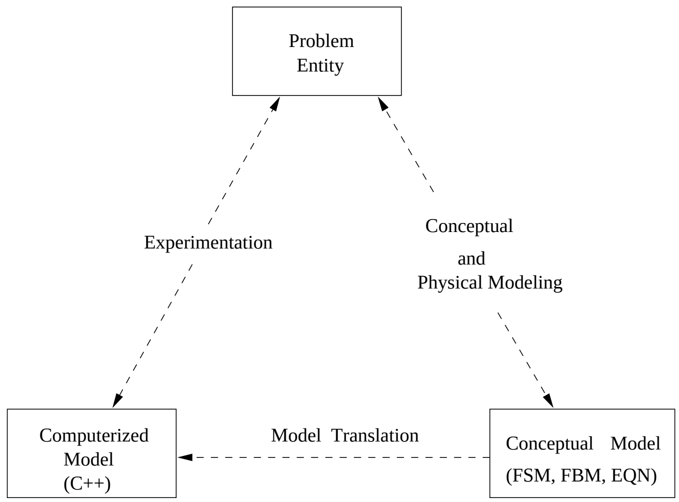


Figure 2. Modeling process in MOOSE

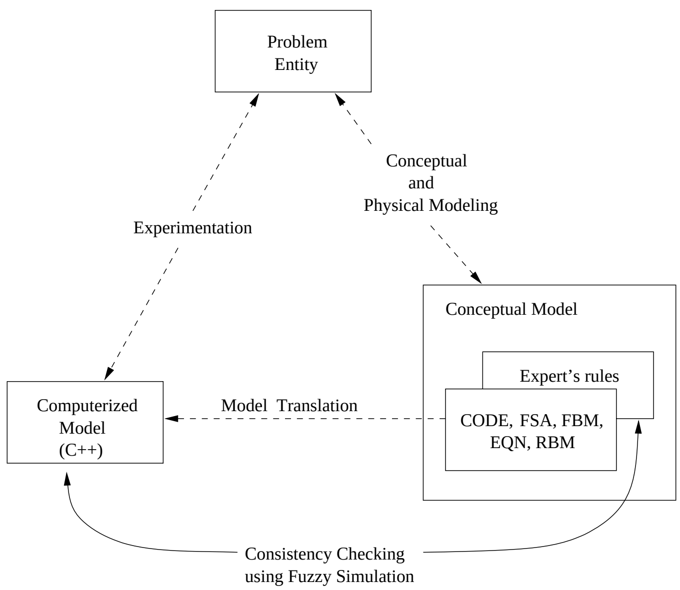


Figure 3. Consistency checking in MOOSE

1. Developing a user interface for accepting the expert's fuzzy rules as one of the conceptual models in MOOSE.
2. Making consistency-checking facility available between a computerized model and the expert's rules by using fuzzy simulation.
3. Developing a user interface via human intervention for resolving a inconsistency.

In this paper, we discuss mainly the fuzzy simulation approach that we employed for checking consistency in Task 2. In the fuzzy simulation approach, we are able to calculate a confidence factor for each rule by carrying membership degrees of fuzzy sets in the expert's rule premise and issuing them to simulation components. Then we compare against the confidence factor from the expert. This quantitative measurement provides us with useful information such as the most inconsistent rule and the amount of knowledge discrepancy as a whole. This facility serves to construct a convenient environment for resolving inconsistency in Task 3.

Table 1. Notation

Notation	Usage
MF_{fuzzy}	*Fuzzy Membership Functions* generated by *fuzzy* simulation
$MF_{premise}$	*Membership Functions* of fuzzy value in rule *premise*
MF_{conseq}	*Membership Functions* of fuzzy value in rule *consequence*
CF_{expert}	*Confidence Factor* presented by an *expert*
CF_{fuzzy}	*Confidence Factor* calculated by *fuzzy simulation*

3.1. Formats of Expert Rules as Input

The input of fuzzy simulation is a collection of expert rules. In what follows, we assume that the three following canonical forms of rules are presented by the expert.

- IF $\mathcal{X}$ is A THEN $\mathcal{Y}$ is B (CF)
- IF $\mathcal{X}$ is $(A_1 \star A_2)$ THEN $\mathcal{Y}$ is B (CF)
- IF ($\mathcal{X}$ is A) $*$ ($\mathcal{Y}$ is B) THEN $\mathcal{Z}$ is C (CF),

where $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$are *fuzzy variables* that take real numbers from universal sets X, Y Z, respectively, A, A_1 and A_2 are *fuzzy values* on X, B and C are *fuzzy values* on Y and Z, respectively, CF is a *confidence factor* in the rule consequence given that the premise conditions are satisfied, $\star$ and $*$ are arithmetic (+, -, $\times$ or $\div$) and logic (*or* or *and*) operators.

In what follows, we call the first type of rule *simplex rules*, and the other two types of rule *compound rules*. These two types of rule are handled in different way by the fuzzy simulation method discussed in next section. For simplicity, the notation in Table 1 will be used. The premise parts of the last two canonical types of rules can be combined to make a more complex rule such as IF ($\mathcal{X}$ is $(A_1 + A_2)$) or ($\mathcal{Y}$ is $(B_1 + B_2)$) THEN $\mathcal{Z}$ is C.

3.2. Fuzzy Simulation

The Fuzzy simulation method is capable of simulating the expert rules using quantitative models. For each expert rule, this method takes the premise part and its $MF_{premise}$, and through simulation it generates a conclusion associated with CF_{fuzzy}. With the intention of comparing this result against the expert's counterpart, the fuzzy simulation method is forced to derive the same conclusion that the expert presented, but with possibly different CF_{fuzzy} from CF_{expert}. When the expert rule is *simplex*, fuzzy simulation involves one simulation by taking each element within the $MF_{premise}$. In contrast, when the rule is *compound*, we obtain an intermediate fuzzy set by applying the *extension principle*[8] or the *rule of conjunctive* or *disjunctive composition* [9] prior to sampling.

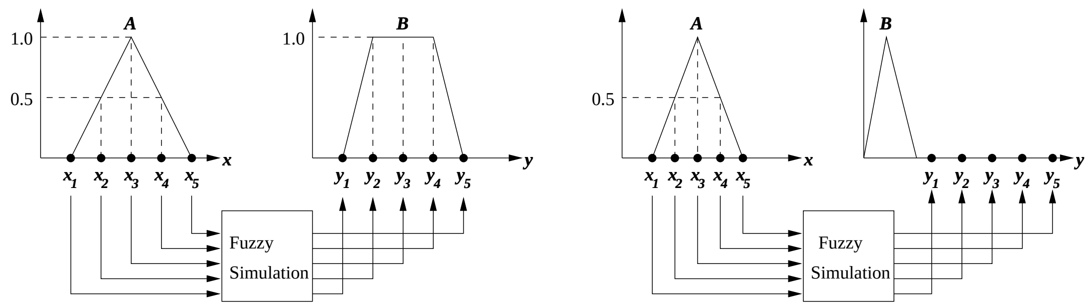
3.2.1. Simplex Rules

- Fuzzy Simulation Algorithm
  Consider a simplex rule of the type, IF $\mathcal{X}$ is A THEN $\mathcal{Y}$ is B. Then the algorithm for fuzzy simulation is:

  1. Let a fuzzy simulation component such as a parameter $\mathbf{p}$ be defined as a fuzzy set A, where

     $$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \ldots + \mu_A(x_n)/x_n.$$

     Assume the element of A is identified by brackets (i.e., $A[2] = x_2$).
  2. For $j \in 1, 2, \ldots, n$:
     (a) Let $\mathbf{p}[j] = A[j]$.
     (b) SIMULATE REAL
     (c) obtain $(\mu_B(y_j)/y_j)(t_e)$



(a) All members of ***A*** support the conclusion ***B*** with full confidence

(b) None of the members of ***A*** support the conclusion ***B***

Figure 4. All members or none of members support the conclusion

3. calculate CF_{fuzzy},

where SIMULATE REAL denotes simulation using real arithmetic, $y_j, j = 1, \ldots, n$ denotes real values on Y, and and t_e is the end time for the simulation.

During SIMULATE REAL, the correlated uncertainty method requires that when we replace $\mathbf{p}$ with a real number whose membership degree is d, we should replace other fuzzy simulation components with real numbers whose membership degrees are also d. This procedure involves a two-step process of searching membership degree of $\mathbf{p}$ and then using this degree to drive the elements of other fuzzy sets. In what follows, SIMULATE REAL involves this operation.

- Calculation of CF_{fuzzy}
  Just as CF_{expert} is presented by expert, we need a way to obtain CF_{fuzzy} from fuzzy simulation. By doing this, we benefit from the comparison of the two rules in terms of their CF values. However, since the derivation of the CF_{expert} involves a subjective opinion as well as certain amount of uncertainty, there is no theoretical formulation to calculate the CF_{fuzzy} whose derivation process is exactly the same as that of the CF_{expert}. Our solution is to define an equation in such a way that its result agrees with *human intuition* as much as possible. We used a *weighted average method* to create such an intuition. Given a simplex rule, we define the CF_{fuzzy} by using the weighted average method

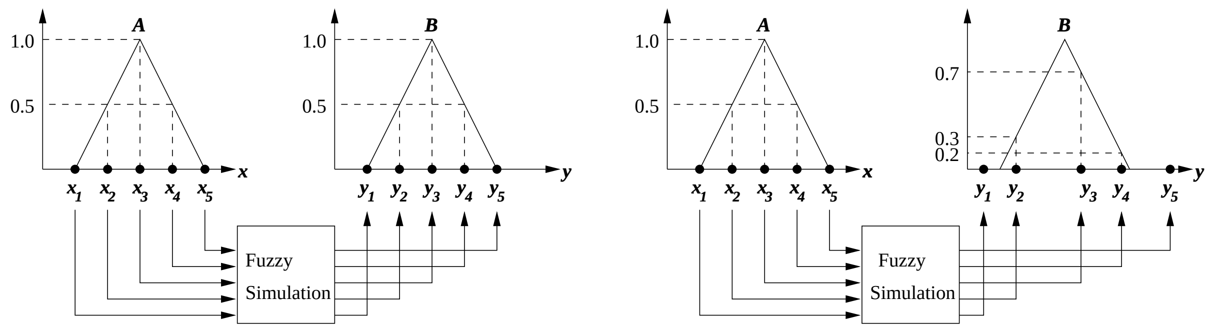
$$CF_{fuzzy} = \frac{\sum_{j=1}^{n} (\mu_A(x_j) \times \mu_B(y_j))}{\sum_{j=1}^{n} \mu_A(x_j)}, \tag{9}$$

  where $x_j, j = 1, 2, \ldots, n$, denote real values on X in the fuzzy set A,
  and $y_j, j = 1, 2, \ldots, n$, denote real values on Y obtained from simulation using x_j. The validity of calculating CF_{fuzzy} using weighted average method can be easily shown in Fig. 4 and Fig. 5. CF_{fuzzy} using Equation (9) is 1.0 and 0.0 for Fig. 4(a) and Fig. 4(b), respectively. The results exactly match our intuition. When the CF falls into some range between the above two extreme cases (i.e., 0.0 and 1.0) as shown in Fig. 5, we can intuitively say that the greater CF we get, each member in A supports the conclusion B with a higher confidence. Using (9), the CF_{fuzzy} for Fig. 5(a) is 0.75 and the CF_{fuzzy} for Fig. 5(b) is 0.475.

3.2.2. Compound Rules with Arithmetic Operations

- Fuzzy Simulation Algorithm
  Consider a compound rule of the type, IF $\mathcal{X}$ is $(A_1 \star A_2)$ THEN $\mathcal{Y}$ is B, where $\star$ is one of the four basic arithmetic operators. Then the algorithm for fuzzy simulation is:

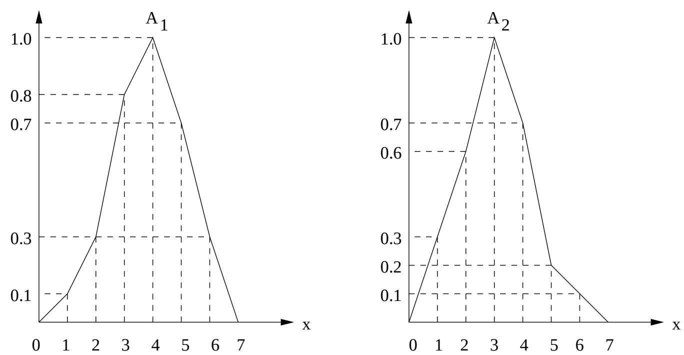
  1. Apply Equation (3) to the rule premise.



(a) Members of ***A*** support the conclusion ***B*** with higher confidence compared to the case of (b)

(b) Members of ***A*** support the conclusion ***B*** with less confidence compared to the case of (a)

Figure 5. Some members support conclusion



(a)

(b)

Figure 6. Two fuzzy sets for addition

2. Let Z be a resulting intermediate fuzzy set, and let a fuzzy simulation component such as a parameter $\mathbf{p}$ be defined as a fuzzy set Z, where

$$Z = \mu_Z(z_1)/z_1 + \mu_Z(z_2)/z_2 + \ldots + \mu_Z(z_n)/z_n.$$

   Assume the element of Z is identified by brackets (i.e., $Z[2] = z_2$).

3. For $j \in 1, 2, \ldots, n$:
   (a) Let $\mathbf{p}[j] = Z[j]$.
   (b) SIMULATE REAL
   (c) obtain $(\mu_B(y_j)/y_j)(t_e)$
4. calculate CF_{fuzzy}.

- Calculation of CF_{fuzzy}
  Given a compound rule with arithmetic operations, we define the CF_{fuzzy} by using the weighted average method

$$CF_{fuzzy} = \frac{\sum_{j=1}^{n}(\mu_{A_1 \star A_2}(z_j) \times \mu_B(y_j))}{\sum_{j=1}^{n}\mu_{A_1 \star A_2}(z_j)}, \tag{10}$$

where $z_j, j = 1, 2, \ldots, n$, denote real values on a fuzzy set resulted from arithmetic operation, $A_1 \star A_2$, $y_j, j = 1, 2, \ldots, n$, denote real values on Y obtained from simulation using z_j.

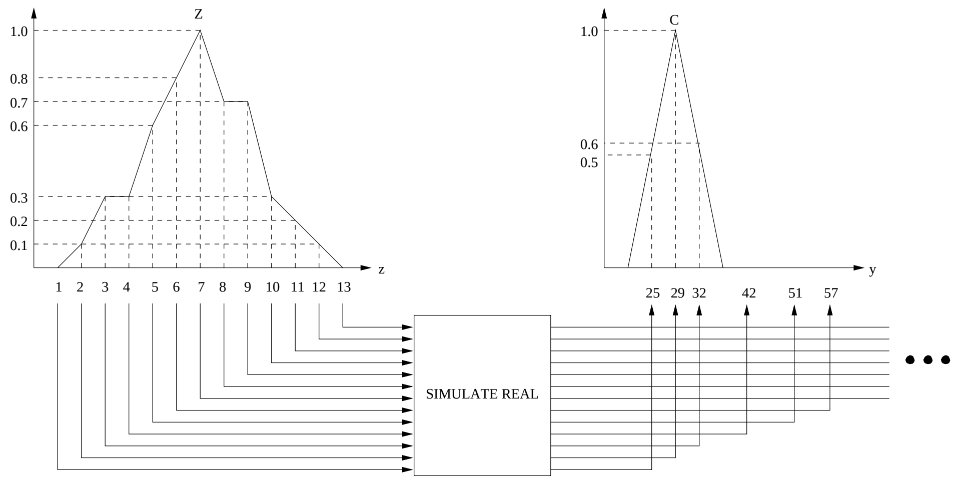


Figure 7. Fuzzy simulation using compound (addition) rule

- Example
  Let's assume that we want to perform fuzzy simulation using the following rule, IF $\mathcal{X}$ is $(A_1 + A_2)$ THEN $\mathcal{Y}$ is B, where A_1 and A_2 are defined by Fig. 6 (a) and Fig. 6 (b). Then by applying Equation (4) defined by

$$\mu_{A_1+A_2}(z) = max_{z=A_1+A_2}(\mu_{A_1}(x) \wedge \mu_{A_2}(x)),$$

  we can obtain the following set of equation for intermediate fuzzy set Z.

$$\begin{aligned}
\mu_Z(1) &= (0 \wedge 0.3) \vee (0 \wedge 0.1) = 0, \\
\mu_Z(2) &= (0 \wedge 0.6) \vee (0.1 \wedge 0.3) \vee (0.3 \wedge 0) = 0.1, \\
\mu_Z(3) &= (0 \wedge 1) \vee (0.1 \wedge 0.6) \vee (0.3 \wedge 0.3) \vee (0.8 \wedge 0) = 0.3, \\
\ldots\ldots &\quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\mu_Z(13) &= (0.3 \wedge 0) \vee (0 \wedge 0.1) = 0.
\end{aligned}$$

  Fig. 7 shows the fuzzy set Z and the result of fuzzy simulation using Z, where the result is arbitrarily made for illustration purposes. Using Equation (10), we can calculate CF_{fuzzy} by

$$CF_{fuzzy} = \frac{(0.1 \times 1.0) + (0.3 \times 0.6)}{0.1 + 0.3 + 0.3 + 0.6 + 0.8 + 1.0 + 0.7 + 0.7 + 0.3 + 0.2 + 0.1} = 0.06$$

3.2.3. Compound Rules with Logic Operations

- Fuzzy Simulation Algorithm
  Consider a compound rule of the type, IF ($\mathcal{X}$ is A) $*$ ($\mathcal{Y}$ is B) THEN $\mathcal{Z}$ is C, where $*$ denotes any logical operator. Then the algorithm for fuzzy simulation is:

  1. If $*$ is *and* operator, then apply the *rule of conjunctive composition* (*Definition 2.3*) to rule premise and calculate a possibility distribution $\pi(x, y)$. If $*$ is *or* operator, then apply the *rule of disjunctive composition* (*Definition 2.3*) to rule premise and calculate a possibility distribution $\pi(x, y)$.
  2. Let fuzzy simulation components such as **p** and **q** be defined as fuzzy sets A and B, respectively, where

$$\begin{aligned}
A &= \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \ldots + \mu_A(x_n)/x_n, \\
B &= \mu_B(y_1)/y_1 + \mu_B(y_2)/y_2 + \ldots + \mu_B(y_n)/y_n.
\end{aligned}$$

     Assume the elements of A and B are identified by brackets (i.e., $A[2] = x_2$ and $B[2] = y_2$).

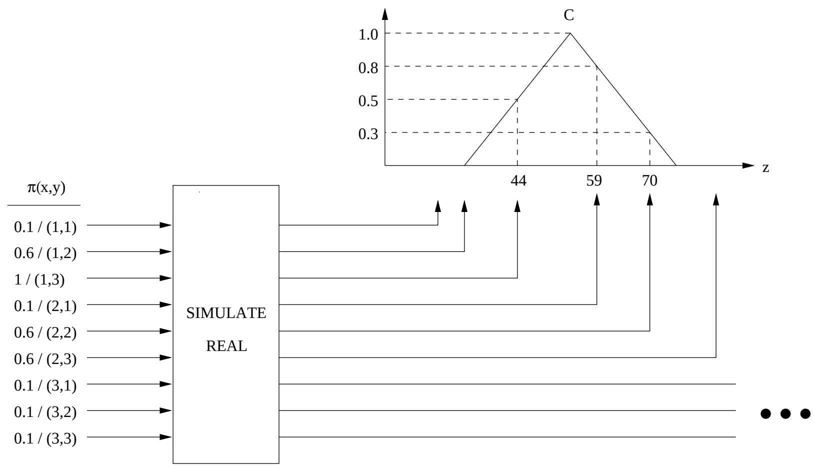


Figure 8. Fuzzy simulation using compound (conjunction) rule

3. For $i \in 1, 2, \ldots, m$
   For $j \in 1, 2, \ldots, n$
   Let $\mathbf{p}[i] = A[i]$.
   Let $\mathbf{q}[j] = B[j]$.
   SIMULATE REAL
   obtain $(\mu_C(z_{ij})/z_{ij})(t_e)$
4. calculate CF_{fuzzy},
   where m and n are the number of elements in A and B, respectively.

Notice that in the rule defined above, the universal sets of the fuzzy variables A and B are not identical. Otherwise, instead of $\pi(x, y)$, we can get a more simplified fuzzy set as an intermediate set by *Definition 2.5* and *Definition 2.6* for disjunction and conjunction, respectively.

- Calculation of CF_{fuzzy}
  Given a compound rule with logic operations, CF_{fuzzy} is defined by using the weighted average method

$$CF_{fuzzy} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} (\mu_{A*B}(x_i, y_j) \times \mu_C(z_{ij}))}{\sum_{i=1}^{m} \sum_{j=1}^{n} \mu_{A*B}(x_i, y_j)}, \tag{11}$$

  where $*$ denotes an logical operator.

- Example
  Let's assume that we want to perform fuzzy simulation using the following compound rule, IF ($\mathcal{X}$ is A) and ($\mathcal{Y}$ is B) THEN $\mathcal{Z}$ is C, where A and B are defined as $A = small = 1/1 + 0.6/2 + 0.1/3$ and $B = large = 0.1/1 + 0.6/2 + 1/3$. Then by applying the *rule of conjunctive composition*, the predicate, ($\mathcal{X}$ is A) and ($\mathcal{Y}$ is B), yields the following possibility distribution:

$$\begin{aligned}
\pi(x,y) &= \{[\mu_{A\,and\,B}(x_1,y_1)/(x_1,y_1)], [\mu_{A\,and\,B}(x_1,y_2)/(x_1,y_2)], \\
&= [\mu_{A\,and\,B}(x_1,y_3)/(x_1,y_3)], [\mu_{A\,and\,B}(x_2,y_1)/(x_2,y_1)], \\
&\quad \ldots, [\mu_{A\,and\,B}(x_3,y_3)/(x_3,y_3)]\} \\
&= \{[0.1/(1,1)], [0.6/(1,2)], [1/(1,3)], [0.1/(2,1)], [0.6/(2,2)], [0.6/(2,3)], \\
&\quad [0.1/(3,1)], [0.1/(3,2)], [0.1/(3,3)]\}
\end{aligned}$$

  Let's assume that we have the result as shown in Fig. 8 by performing fuzzy simulation on this $\pi(x, y)$. Using Equation (11), we can calculate CF_{fuzzy} by

$$CF_{fuzzy} = \frac{(1.0 \times 0.5) + (0.1 \times 0.8) + (0.6 \times 0.3)}{1.0 + 0.1 + 0.6} = 0.44$$

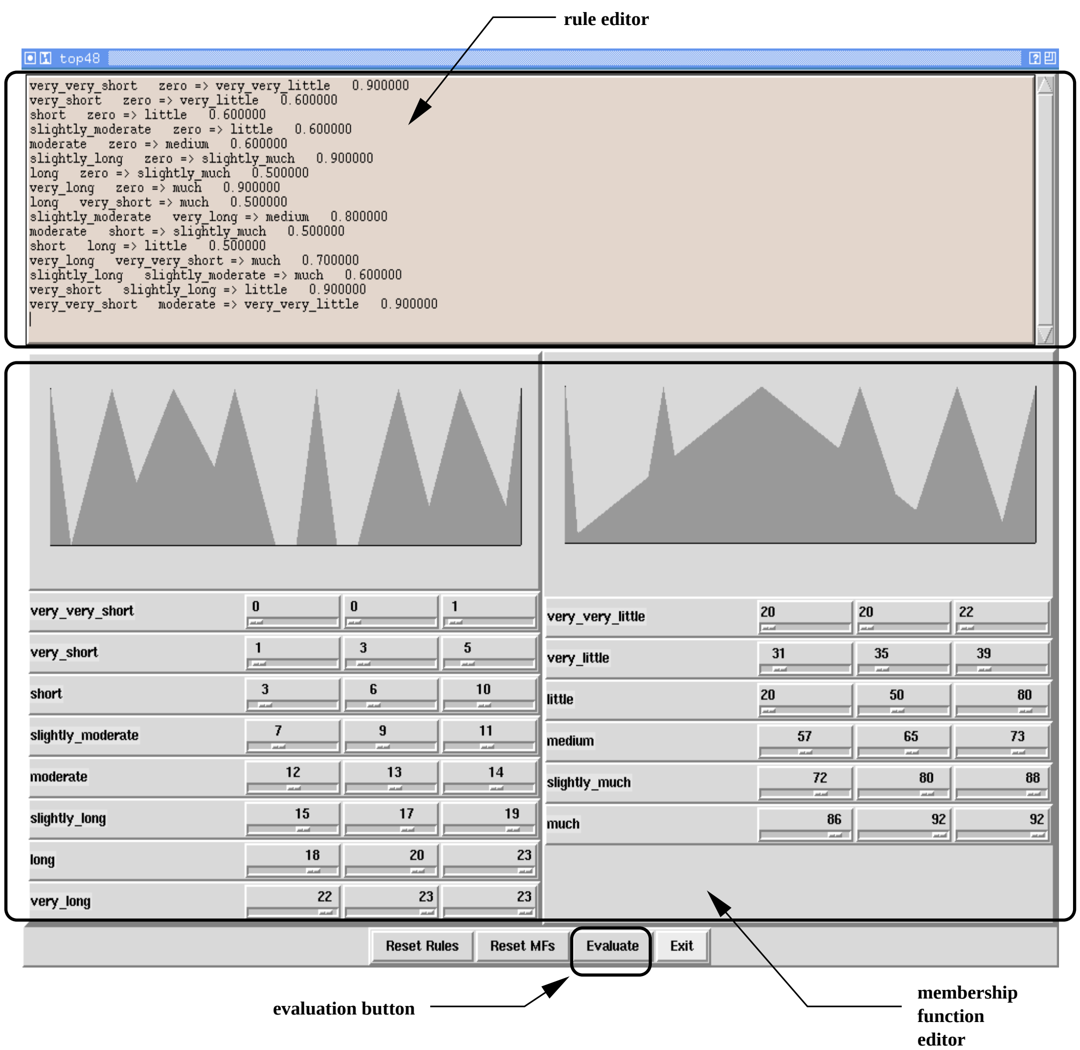


Figure 9. A GUI for human intervention for resolving inconsistency

3.3. Resolving Inconsistency

For the case where the amount of inconsistency is out of range after executing fuzzy simulations, we have developed an advising facility which suggests all expected rules from fuzzy simulations. With this information, the model author starts to resolve the inconsistency. For this process, either the expert rules (including CF_{expert} and membership function definitions) or the simulation parameters can be modified interactively. Every time these modification happens, the fuzzy simulation is reinvoked with visual aids so that the user can easily recognize the effect of the modification. Fig. 9 shows the GUI we developed for the human intervention to resolve the inconsistency.

4. CONCLUSION

As we discussed in the previous sections, the consistency between two types of models (expert's rules and conceptual models in MOOSE) can be measured by the difference between the CF presented by an expert and the CF calculated from fuzzy simulation on each rule. This gives the model author of MOOSE useful information, such as which components of the conceptual models should be further investigated. Whenever the inconsistency is detected, the quantitative measure mentioned above helps the human author identify and revise the most inconsistent component rapidly and analyze the effectiveness of that modification, thereby allowing the two models to gradually reach a consensus with high resolution. Consequently, by incorporating the proposed method into MOOSE, we can obtain a benefit from validating the simulation models against the expert's knowledge.

ACKNOWLEDGEMENTS

We would like to thank the following funding sources that have contributed towards our study of modeling and implementation of the MOOSE multimodeling simulation environment: GRCI Incorporated (Gregg Liming) and Rome Laboratory (Steve Farr) for web-based simulation and modeling, as well as Rome Laboratory (Al Sisti) for multimodeling and model abstraction. We also thank the Department of the Interior under a contract under the ATLSS Project (Don DeAngelis, University of Miami). Without their help and encouragement, our research would not be possible.

REFERENCES

1. T. G. R. M. Cubert and P. A. Fishwick, "MOOSE: Architecture of an Object-Oriented Multimodeling Simulation System," in *SPIE AeroSense*, , ed., *Proceedings of Enabling Technology for Simulation Science* **April**, 1997.
2. R. M. Cubert and P. A. Fishwick, "MOOSE: An Object-Oriented Multimodeling and Simulation Application Framework," *submitted to Simulation* **June**, 1997.
3. P. A. Fishwick, *Simulation Model Design and Execution: Building Digital Worlds*, Prentice Hall, New Jersey, U.S.A., 1995.
4. R. G. Sargent, "Simulation Model Verification and Validation," in *WSC'91*, B. L. Nelson, W. D. Kelton, and G. M. Clark, ed., *1991 Winter Simulation Conference Proceedings* **8-11 December**, pp. 37–47, 1991.
5. O. B. R. O'Keefe and E. P. Smith, "Validating expert system performance," *IEEE Expert* **Winter**, pp. 81–89, 1987.
6. L. A. Zadeh, "Fuzzy Sets," *Information and Control* **8**, pp. 328–353, 1965.
7. L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Trans. Systems, Mans, and Cybernetics* **SMC-3**, pp. 28–44, 1973.
8. L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning I," *Information Sciences* **8**, pp. 199–251, 1975.
9. H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*, Kluwer-Nijhoff Publishing, MA, U.S.A., 1985.
10. D. Dubois, H. Prade, and R. R. Yager, "Basic Notions in Fuzzy Set Theory," in *Fuzzy Sets for Intelligent Systems*, D. Dubois, H. Prade, and R. R. Yager, ed., pp. 27–64, Morgan Kaufmann, 1993.
11. G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, New Jersey, U.S.A., 1995.
12. R. G. Sargent, "An Exploration of Possibilities for Expert Aids in Model Validation," in *Modeling and Simulation Methodology in the Artificial Intelligence Era*, M. S. Elzas, T. I. Ören, and B. P. Zeigler, ed., pp. 279–297, Elsevier Science, 1986.
13. L. G. Brita and F. N. Ozmizrak, "A Knowledge-Based Approach for the Validation of Simulation Models: The Foundation," *ACM Transaction on Modeling and Computer Simulation* **6(1)**, pp. 76–98, 1996.