

An Object-Oriented Architecture for Possibilistic Models

Cliff Joslyn

Mail Stop 522.3, NASA Goddard Space Flight Center
Greenbelt, MD 20771, USA
joslyn@kong.gsfc.nasa.gov
http://groucho.gsfc.nasa.gov/joslyn

Abstract. An architecture for the implementation of possibilistic models in an object-oriented programming environment (C++ in particular) is described. Fundamental classes for special and general random sets, their associated fuzzy measures, special and general distributions and fuzzy sets, and possibilistic processes are specified. Supplementary methods—including the fast Möbius transform, the maximum entropy and Bayesian approximations of random sets, distribution operators, compatibility measures, consonant approximations, frequency conversions, and possibilistic normalization and measurement methods—are also introduced. Empirical results to be investigated are also described.

1 Introduction

Possibility theory [4] is an alternative information theory to that based on probability. Although possibility theory is logically independent of probability theory, they are related: both arise in Dempster-Shafer evidence theory as fuzzy measures defined on random sets; and their distributions are both fuzzy sets. So possibility theory is a component of a broader Generalized Information Theory (GIT), which includes all of these fields [18].

Possibility theory was originally developed in the context of fuzzy systems theory [28]. More recently, possibility theory is being developed independently of both fuzzy sets and probability. In particular, the author is developing the mathematics and semantics of possibility theory [9, 12, 13]. These methods include possibilistic measurement procedures [8, 10] and possibilistic processes such as possibilistic automata [11]—generalizations of nondeterministic processes whose non-additive weights adhere to the laws of mathematical possibility theory.

This paper describes an architecture for implementing possibilistic models in an object-oriented environment. The approach is based on the mathematics of consistent random sets as the basis for the representation of measured possibility distributions and possibilistic processes.

2 Possibility Theory

Mathematical possibility theory can only be briefly introduced here. See [4, 12, 18] for details and proofs.

2.1 Mathematical Possibility in GIT

Given a finite universe $\Omega := \{\omega_i\}$, $1 \leq i \leq n$, the function $m: 2^\Omega \mapsto [0, 1]$ is an evidence function (otherwise known as a basic probability assignment) when $m(\emptyset) = 0$ and $\sum_{A \subseteq \Omega} m(A) = 1$. Denote a random set generated from an evidence function as $\mathcal{S} := \{(A_j, m_j) : m_j > 0\}$, where $\langle \cdot \rangle$ is a vector, $A_j \subseteq \Omega$, $m_j := m(A_j)$, and $1 \leq j \leq N := |\mathcal{S}| \leq 2^n - 1$. A random set \mathcal{S} is essentially a subset-valued random variable on 2^Ω , with m its “probability distribution”. Denote the focal set of \mathcal{S} as $\mathcal{F} := \{A_j : m_j > 0\}$ with core and support respectively $C(\mathcal{F}) := \bigcap_{A_j \in \mathcal{F}} A_j$, $U(\mathcal{F}) := \bigcup_{A_j \in \mathcal{F}} A_j$.

The plausibility and belief measures on $\forall A \subseteq \Omega$ are $Pl(A) := \sum_{A_j \cap A \neq \emptyset} m_j$ and $Bel(A) := \sum_{A_j \subseteq A} m_j$. These are non-additive fuzzy measures [27], and are dual, in that $\forall A \subseteq \Omega$, $Bel(A) = 1 - Pl(\bar{A})$. In general only plausibility will be considered below. Bel (and Pl as its dual) determines the evidence function according to the Möbius inversion

$$m(A) = \sum_{B \subseteq A} (-1)^{|B-A|} Bel(B). \quad (1)$$

The plausibility assignment (otherwise known as the one-point coverage function) of \mathcal{S} is $Pl = \langle Pl_i \rangle := \langle Pl(\{\omega_i\}) \rangle$, where

$$Pl_i := \sum_{A_j \ni \omega_i} m_j. \quad (2)$$

Random set inclusion is defined by the formula

$$\mathcal{S}_1 \subseteq \mathcal{S}_2 := \forall A \subseteq \Omega, Pl_1(A) \leq Pl_2(A). \quad (3)$$

The best justified formula for combining two random sets $\mathcal{S} := \mathcal{S}_1 \odot \mathcal{S}_2$, yielding a combined evidence function $m = m_1 \odot m_2$, is Dempster’s rule

$$\forall A \subseteq \Omega, \quad m(A) := \frac{\sum_{A_1 \cap A_2 = A} m_1(A_1) m_2(A_2)}{\sum_{A_1 \cap A_2 \neq \emptyset} m_1(A_1) m_2(A_2)}. \quad (4)$$

Since $\forall \omega_i, Pl_i \in [0, 1]$, therefore Pl is the membership function of a fuzzy subset of Ω , denoted $\tilde{P}l$. Conversely, any fuzzy subset $\tilde{F} \subseteq \Omega$ with membership function $\mu_{\tilde{F}}: \Omega \mapsto [0, 1]$ can be mapped to an equivalence class of one-point equivalent random sets on Ω . If $\sum_i \mu_{\tilde{F}}(\omega_i) \geq 1$, then \tilde{F} can be taken as a plausibility assignment of any of an equivalence class of random sets; similarly, if $\sum_i \mu_{\tilde{F}}(\omega_i) \leq 1$, then \tilde{F} can be taken as the one-point assignment of a belief function of any of a different equivalence class of random sets. Under some

conditions the evidence values m_j are determined by the plausibility assignment values Pl_i . Then $N \leq n$, and Pl is a **distribution** of \mathcal{S} .

When $\forall A_j \in \mathcal{F}, |A_j| = 1$, then \mathcal{S} is **specific**, and $\Pr(A) := Pl(A) = Bel(A)$ is a **probability measure** which is additive in the traditional way

$$\forall A, B \subseteq \Omega, \quad \Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B). \quad (5)$$

Then $\mathbf{p} = \langle p_i \rangle := Pl$ is a **probability distribution** with additive normalization and operator

$$\sum_i p_i = 1, \quad \Pr(A) = \sum_{\omega_i \in A} p_i. \quad (6)$$

\mathcal{S} is **consonant** (\mathcal{F} is a **nest**) when (without loss of generality for ordering, and letting $A_0 := \emptyset, A_{j-1} \subseteq A_j$). Now $\Pi(A) := Pl(A)$ is a **possibility measure** and $\eta(A) := Bel(A)$ is a **necessity measure**. Since results for necessity are dual to those of possibility, only possibility will be discussed in the sequel.

As \Pr is additive, so Π is **maximal**, in that $\forall A, B \subseteq \Omega, \Pi(A \cup B) = \Pi(A) \vee \Pi(B)$, where \vee is the maximum operator. As long as $C(\mathcal{F}) \neq \emptyset$ (this is required if \mathcal{F} is a nest), then $\pi = \langle \pi_i \rangle := Pl$ is a **possibility distribution** with maximal normalization $\vee_i \pi_i = 1$ and operator

$$\Pi(A) = \bigvee_{\omega_i \in A} \pi_i. \quad (7)$$

However, **consistency** $C(\mathcal{S}) \neq \emptyset$, not consonance, is all that is necessary for Pl to be a maximally normalized possibility distribution π , even though the plausibility *measure* of a consistent, non-consonant random set is not a possibility measure. But given a possibility distribution π , there is a unique consonant one-point equivalent random set, denoted \mathcal{S}^π and called the **consonant approximation**. It is determined by taking the ω_i in order of descending π_i , and letting $A_i = \{\omega_1, \omega_2, \dots, \omega_i\}$ and $m_i = \pi_i - \pi_{i+1}$, where $\pi_{n+1} := 0$.

Nonspecificity N and **strife** S are two **uncertainty measures** which are defined on random sets, respectively

$$N(\mathcal{S}) := \sum_j m_j \log_2 |A_j|, \quad S(\mathcal{S}) := - \sum_j m_j \log_2 \left[\sum_{k=1}^n m_k \frac{|A_j \cap A_k|}{|A_j|} \right]. \quad (8)$$

They measure respectively the possibilistic and probabilistic aspects of the uncertainty or information represented in the random set, and together form the **total uncertainty** $T := N + S$. They have special forms for distributions, and in the probabilistic case the uncertainty collapses to stochastic entropy $H(\mathbf{p})$, while in the possibilistic case the strife is bounded above by a small number.

Generalized **processes** are defined on distributions when a generalized **disjunction operator** \oplus and a generalized **conjunction operator** \otimes on $[0, 1]$ are available such that:

- (\oplus, \otimes) form a semiring (\otimes distributes over \oplus),

- $\oplus = \sqcup$ and $\otimes = \sqcap$ are a **triangular conorm** and **norm**: monotonic, associative, commutative operators with identity 0 and 1 respectively [4], so that $(\oplus, \otimes) = (\sqcup, \sqcap)$ is a **conorm semiring**,
- and \oplus is the operator of the distributions in question.

For $x, y \in [0, 1]$, the operation $x \sqcup_m y := (x + y) \wedge 1$ is a conorm, where \wedge is the minimum operator. So stochastic normalization (6) forces \oplus , when acting on a probability distribution and restricted to $[0, 1]$, to be a conorm $\oplus = \sqcup_m = \oplus$. Then (\oplus, \times) is the unique semiring for stochastic processes. For possibility, $\oplus = \vee$ is the unique conorm operator, but there are many semirings of the form (\vee, \sqcap) for a generic norm \sqcap . \wedge and \times are two of the more popular norms, as is $x \sqcap_m y := 0 \vee (x + y - 1)$.

Given an appropriate semiring, then **joint**, **marginal**, and **conditional distributions** are available, with novel possibilistic forms. In particular, **conditional possibility** is parameterized by the choice of norm operator \sqcap , and even once a norm is fixed, $\pi(y|x)$ is not always unique.

Given a particular **current state distribution** (probabilistic \mathbf{p} or possibilistic π), and a table of conditional probabilities or possibilities, then the **next state distribution** \mathbf{p}' or π' is derived from a generalized linear matrix composition operation (the familiar form in the probabilistic case).

A **possibilistic process** is a system $\mathcal{Z}_\pi := \langle \Omega, \sqcap, \Pi, \pi^0 \rangle$ for some \sqcap , where:

- $\Pi = [\Pi_{ij}] = \langle \Pi^{(j)} \rangle$ is a matrix of conditional possibilities, so that for $1 \leq j \leq n, \Pi^{(j)}$ is the vector representation of a conditional possibility distribution function $\pi(\cdot|\omega_j): \Omega \mapsto [0, 1]$, Π is a vector of such conditional distributions, and $\Pi_{ij} := \pi(\omega_i|\omega_j)$;
- π^t is a possibility distribution on Ω with π^0 given, and $\forall t > 0$,

$$\pi^t := \pi^{t-1} \circ \Pi, \quad \pi_i^t := \bigvee_{j=1}^N \pi_j^{t-1} \sqcap \pi(\omega_i|\omega_j), \quad (9)$$

where \circ is (\vee, \sqcap) matrix composition.

$\pi_i^t = \pi^t(\omega_i)$ is the possibility of being in state ω_i at time t , and $\Pi_{ij} = \pi(\omega_i|\omega_j)$ is the possibility of transiting from state ω_j to ω_i .

It has been demonstrated [12, 11] that while both stochastic and possibilistic processes generalize deterministic processes, only possibilistic processes are generalizations of *nondeterministic* processes. When possibility values are restricted to $\{0, 1\}$, then the strict nondeterministic case is recovered.

Possibilistic versions of some of the standard stochastic systems theoretical forms have been defined [12]. In particular, **possibilistic Markov processes** and **Monte Carlo methods** are available, and **possibilistic automata** are constructed by extending possibilistic processes to include possibilistically distributed input and output alphabets and functions.

Finally, it should be noted that both probabilistic and possibilistic processes are specializations of fuzzy processes using fuzzy matrix composition [18]. In each

special case the conorm semiring (\sqcup, \sqcap) is restricted to special forms, resulting in the various normalization and other conditions mentioned above, which would not otherwise be required.

2.2 Possibilistic Models

A possibilistic model requires possibilistic measurement and prediction procedures. Possibilistic prediction is based on the possibilistic processes briefly outlined above. **Possibilistic measurement** methods have also been developed by the author [8, 10]. The essential requirement is the collection of the frequency of occurrence of subsets or intervals which are partially overlapping. If their global intersection is nonempty (the empirical random set is consistent), then (2) yields an empirical possibility distribution. Otherwise, possibilistic normalization procedures [9] would be required.

3 CAST Implementations of Possibilistic Models

The Computer-Aided Systems Theory (CAST) movement is predicated on the idea that computer implementations are at least useful, and sometimes even necessary, for the development and application of systems theoretical methods. This has been remarked on by Klir.

Systems knowledge can also be obtained experimentally. Although systems (knowledge structures) are not objects of reality, they can be simulated on computers and in this sense made real. We can then experiment with the simulated systems for the purpose of discovering or validating various hypotheses in the same way as other scientists do with objects of their interest in their laboratories. In this sense, computers, may be viewed as laboratories of systems science. Experimentation with systems on computers is not merely possible, but it may give us knowledge that is otherwise unobtainable. [16, p. 102]

Horgan notes [6] that as the complexity of problems increases, this situation is becoming common generally in mathematics. The result is the growth of so-called "experimental" or "computer-aided" mathematics, where computer-based tools are used to empirically investigate the properties of mathematical systems.

CAST implementations of possibilistic systems in particular are crucial not only as platforms for the application of possibilistic qualitative modeling, but also for the empirical investigation of the properties of possibilistic processes. There are still many open questions, as described more fully elsewhere [12].

Existing CAST implementations (for example Pichler's [22] and Zeigler's [29]) are deterministic. The extension of these implementations, and the development of new environments, to include representations of indeterminism, uncertainty, and information is crucial. For example, existing systems could not implement neural networks with (stochastic) noise.

It is also clear that the fundamental categories for the representation of uncertainty and information should be included in the foundational, primitive levels of CAST systems, from which implementations of more complex and specific systems theoretical methods should then be constructed. This includes the entire repertoire of GIT, allowing, for example, the use of methods from probability, statistics, and fuzzy theory, as well as random sets and possibility theory.

GIT-based CAST implementations should allow the handling of hybrid sources and representations of uncertainty, the integration of multiple sources of information, and the transformation between representational forms of information. For example, Klir's General Systems Problem Solver (GSPS) [15] was designed specifically to accommodate both probabilistic and possibilistic representations of information, and is best implemented in a GIT-based general systems theoretical CAST environment.

There has recently been an explosion of fuzzy systems implementations for both the commercial and academic markets ([7, 23] are examples). The same is not the case for general GIT methods, however, and certainly not for possibility theory. One exception is the work of Galway [5], who has implemented a system for manipulating random subsets of \mathbb{R}^2 .

4 Object-Oriented Environments

One of the most successful programming paradigms of recent years is the **object-oriented** approach [3]. This methodology is based on the concepts of **objects**, which are complex data elements, and **classes**, or "intelligent data types" for objects, which isolate type-specific procedures within type-specific levels. Logical relations among classes allow for the **inheritance** of procedures from more general classes to their specialized cases. Classes and objects have **attributes**, either other objects (data attributes) or **methods** (procedural attributes). A class **invariant** is a logical condition which must always be true of every object of the class in order for it to be existing in a legal state.

The target language for the proposed architecture below is C++ [24]. It was selected for its popularity and efficiency, and because of the availability of standard, inexpensive compilation environments and software support libraries.

The main results of this paper are summarized in Figs. 1-3, which show Entity-Relationship (ER) diagrams [3] of the proposed class hierarchies. These are a slight modification of the standard form presented by Coad and Yourdon [3]. Each node denotes a class (data type). Nodes are linked by labeled arcs, each indicating one of the following relations, where X and Y are classes:

- $X \xrightarrow{\text{is-a}} Y$: Y inherits from X , so that Y is a **specification** of X and has all the properties of X . X is called a **parent** and Y a **child**. For engineering reasons, or to capture efficiencies present in the special cases, some attributes may be implemented redundantly in child classes. For example, the formula for random set nonspecificity $N(\mathcal{S})$ is greatly simplified in the consonant case by calculating the nonspecificity of the possibility distribution $N(\pi)$.

- $X \xrightarrow{\text{collection}} Y$: Y objects are implemented as a **collection** (for example, a list, set, bag, or vector) of X objects.
- $X \xrightarrow{\text{has-a}} Y$: Y is a **component** of X , so that each X -object contains a Y sub-object. When Y is a sub-object of X , then Y may have access to X -specific information. For engineering or efficiency reasons, Y may be implemented separately from X , or a Y may be constructed from an X , copying the appropriate X -specific information into the Y object. In fact, at the strictly logical level this relation simply requires that each X object **determines** a unique Y object, or that a procedure exists to construct a Y object from an X object. This is the sense that will frequently be used below.

Note that while arrows always move from the more general to the more specific, this is not always mirrored by the English translations. For example, $X \xrightarrow{\text{is-a}} Y$ is read that " Y is an X ", while $X \xrightarrow{\text{has-a}} Y$ is read that " X has a Y ".

5 Fundamental Classes

Each of the ER diagrams below describes a different portion of the overall architecture, and is accompanied by a set of descriptions of the classes included in the figure. Only the most basic methods and procedures are included here; supplementary methods are described in Sec. 6.1.

The ER diagrams and class descriptions are written in a kind of non-C++-specific class design "pseudo-code". Only the *logical* relations among the classes are described. For example, in Fig. 1 it is not specified exactly how a plausibility assignment is determined from a plausibility measure. It will be presumed that if $X \xrightarrow{\text{has-a}} Y$, then Y will have access to X -specific information. The figures share the class *Poss_Dist*, for possibility distributions, in common.

5.1 Random Sets

Fig. 1 shows the class hierarchy including random sets and their fuzzy measures and distributions. The class *Random_Set* is the most general, and therefore one of the most heavily laden, classes in the proposed architecture.

Random_Set — A random set S .

Data Attributes	Universe	Integer $n = \Omega $
	Card	Integer $N = S $
Data		A list of pairs (a_j, m_j) , with floats m_j and integers a_j , where a_j is an n -long bit-mask determining the characteristic function χ_{A_j} of the focal element $A_j \subseteq \Omega$
	Invariants	$1 \leq n, \quad 0 \leq m_j \leq 1, \quad \sum_{j=1}^N m_j = 1$ $1 \leq a_j \leq 2^n - 1, \quad 1 \leq j \leq N \leq 2^n - 1$
Methods	Strife	$S(S)$
	Nonspec	$N(S)$
	Total	$T(S)$
	Core	$C(S)$

Random sets, continued.

Methods	Monte_Set	Select a focal element A_j by a possibilistic Monte Carlo method
	Monte	Select a universe element ω_i by a possibilistic Monte Carlo method
	+	The Dempster combination operator \odot (4) which combines this <i>Random_Set</i> with another, producing a combined <i>Random_Set</i>
	<=	The random set inclusion relation \subseteq (3), a boolean reporting whether this <i>Random_Set</i> is included within another
	Complete?	Boolean: is S complete?

Consistent_RS — A consistent random set.

Invariant $\|C(S) \neq \emptyset$

Consonant_RS — A consonant random set.

Data Attribute | Ordering | A list $\langle a_j \rangle$ permuting the a_j according to the inclusion relation among the A_j

Invariants | $\forall 1 \leq j_1 \leq j_2 \leq N, A_{j_1} \subseteq A_{j_2}, \quad 1 \leq N \leq n$

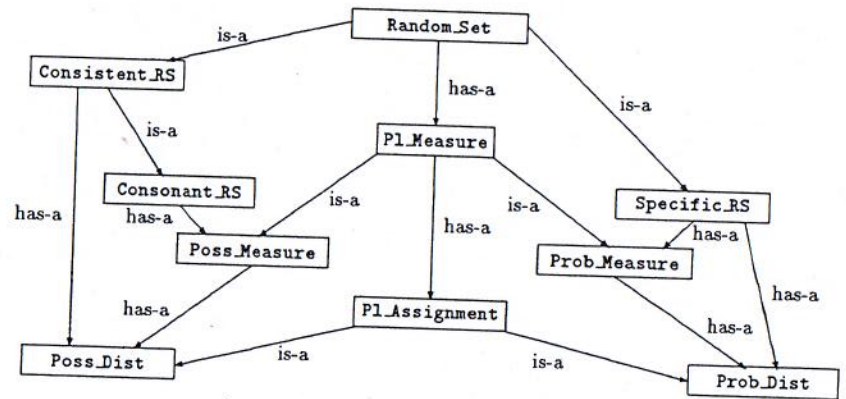


Fig. 1. Random sets, evidence measures, and distributions.

Specific_RS — A specific random set.

Invariants $\forall 1 \leq j \leq N, |A_j| = 1, \quad 1 \leq N \leq n$

Pl_Measure — A plausibility measure PI on a random set.

Method | Value | Given $0 \leq a_j \leq 2^n - 1$, returns a float $PI^j := PI(A_j)$.

Poss_Measure — A possibility measure Π .

Invariant $\forall a_1, a_2, \text{Pl}(A_1 \cup A_2) = \text{Pl}^1 \vee \text{Pl}^2$.

Prob_Measure — A probability measure Pr .

Invariant $\forall a_1, a_2, \text{Pl}(A_1 \cup A_2) = \text{Pl}^1 + \text{Pl}^2 - \text{Pl}(A_1 \cap A_2)$.

Pl_Assignment — A plausibility assignment Pl from a random set.

Data Attribute	Data	A list of floats $\langle \text{Pl}_i \rangle$, where $\text{Pl}_i = \text{Pl}(\{\omega_i\})$
Invariant		$1 \leq i \leq n$

Poss_Dist — A possibility distribution π .

Invariant		$\bigvee_i \text{Pl}_i = 1$
Methods	Nonspec	$N(\pi)$
	Strife	$S(\pi)$
	Total	$T(\pi)$
	Core	$C(\pi)$

Prob_Dist — A probability distribution p .

Invariant		$\sum_i \text{Pl}_i = 1$
Method	Entropy	$H(p)$

5.2 General Distributions

Fig. 2 shows the class hierarchy of distributions and fuzzy sets.

Element — A generic element of a distribution or fuzzy set.

Data Attribute	Value	A floating-point "fit" (fuzzy digit), f .
Invariant		$0 \leq f \leq 1$

Dist_Elem — An element of a distribution.

Methods	+	The generalized disjunction \oplus which aggregates this Dist_Elem with another, producing an aggregated Dist_Elem .
	*	The generalized conjunction operator \otimes which combines this Dist_Elem with another, producing a combined Dist_Elem .

Fuzzy_Set — A collection of elements comprising μ .

Data Attributes	Universe	Integer $n = \Omega $
	Data	A list of Elements $\langle f_i \rangle$.
Invariant		$1 \leq i \leq n$

Poss_Elem — Element of a possibility distribution π_i .

Invariant $f_1 \sqcup f_2 = f_1 \vee f_2$.

Poss_Dist — A possibility distribution π .

Data Attributes	Universe	Integer $n = \Omega $
	Data	A list of Poss_Elems $\langle \pi_i \rangle$
Invariant		$1 \leq i \leq n, \bigvee_{i=1}^n \pi_i = 1$

Prob_Elem — Element of a probability distribution p_i .

Invariant $f_1 \sqcup f_2 = f_1 + f_2$.

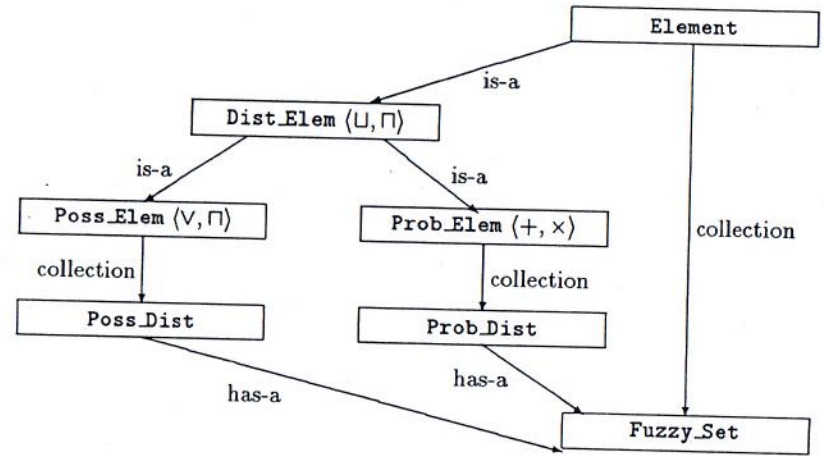


Fig. 2. Distributions and fuzzy sets.

Prob_Dist — A probability distribution p .

Data Attributes	Universe	Integer $n = \Omega $
	Data	A list of Prob_Elems $\langle p_i \rangle$
Invariant		$1 \leq i \leq n, \sum_{i=1}^n p_i = 1$

Note that **Poss_Dist** and **Prob_Dist** are repeated here as collections of their elements, inheriting from **Pl_Assignment** from Fig. 1.

5.3 Possibilistic Processes

Finally, Fig. 3 shows the class hierarchy of possibilistic processes. **Poss_Dist** has been specified above.

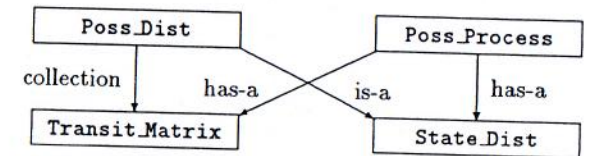


Fig. 3. Possibilistic processes.

Transit_Matrix — A possibilistic transition matrix Π .

Data Attributes	Universe	Integer $n = \Omega $
	Data	A list of Poss_Dists $\langle \Pi^{(j)} \rangle$.
Invariant		$1 \leq j \leq n$

State_Dist — The current state possibility distribution π^t .

Data Attribute	Time	The current time, an integer t
Invariant		$0 \leq t$

Poss_Process — A possibilistic process Z_π .

Method|Advance|Determine next state function $\pi^t = \pi^{t-1} \circ \Pi$.

6 Extensions to the Basic Architecture

Of course, the architecture described above is merely the core for a broader implementation of possibilistic models, which must also involve a variety of measurement methods and links into CAST implementations of other aspects of GIT, let alone the input/output routines necessary for any software system.

6.1 Supplementary Methods

There are a number of other procedures [12] which are special or supplementary to the basic procedures, but which are still beneficial to implement explicitly. Most of these are transformations of one of the classes to another. The following relations can be appended to the basic diagrams above as appropriate.

Möbius Transform: The Möbius transform [14] or fast Möbius transform [26] is an algorithm utilizing the Möbius inversion formula (1) to calculate among belief measures and evidence functions. The fast Möbius transformation is extremely efficient, and will be used in implementing the relation $\text{Random_Set} \xrightarrow{\text{has-a}} \text{Pl_Measure}$ from Fig. 1. Let the $\omega_i \in \Omega$ be taken in an arbitrary order, and assume a random set S .

1. Assume the evidence function m of S , and let $m_0 := m$. Then $\forall A \subseteq \Omega$, and $1 \leq i \leq n$, determine m_i by the algorithm

$$m_i(A) := \begin{cases} m_{i-1}(A) + m_{i-1}(A - \{\omega_i\}), & \omega_i \in A \\ m_{i-1}(A), & \omega_i \notin A \end{cases} \quad (10)$$

Then $m_n = \text{Bel}$, where Bel is the belief function of S .

2. Assume the belief function Bel of S , and let $m_n := \text{Bel}$. Then $\forall A \subseteq \Omega$, and $n \geq i \geq 1$, determine m_i by the algorithm

$$m_{i-1}(A) := \begin{cases} m_i(A) - m_i(A - \{\omega_i\}), & \omega_i \in A \\ m_i(A), & \omega_i \notin A \end{cases} \quad (11)$$

Then $m_1 = m$, where m is the evidence function of S .

Distribution Operations:

$$\text{Poss_Dist} \xrightarrow{\text{has-a}} \text{Poss_Measure}, \quad \text{Prob_Dist} \xrightarrow{\text{has-a}} \text{Prob_Measure}. \quad (12)$$

From (7), given a possibility distribution π , a possibility measure Π can be constructed; similarly, from (6), given a probability distribution p , a probability measure Pr can be constructed.

Probabilistic Approximations:

$$\text{Random_Set} \xrightarrow{\text{has-a}} \text{Prob_Dist}. \quad (13)$$

Probability distribution approximations of random sets are available [12] either as the maximum entropy probability distribution

$$p^S(\omega) := \sum_{A_j \ni \omega} \frac{m_j}{|A_j|}, \quad (14)$$

or the Bayesian approximation \bar{p} determined from

$$\bar{p}(\omega_i) := \frac{\sum_{A_j \ni \omega_i} m_j}{\sum_{j=1}^N m_j |A_j|} = \frac{\text{Pl}_i}{\sum_{i=1}^n \text{Pl}_i}. \quad (15)$$

Compatibility Measures: Given a **Prob_Dist** p and **Poss_Dist** π , a **compatibility measure** $\gamma(\pi, p)$ [12] represents the degree of compatibility of consistency between p and π . These are very useful in possibilistic normalization and transformation procedures. The most prominent is the Zadeh-compatibility $\gamma_Z(\pi, p) = \sum_i p_i \pi_i$.

Consonant Approximation:

$$\text{Consistent_RS} \xrightarrow{\text{has-a}} \text{Consonant_RS}. \quad (16)$$

As mentioned in Sec. 2, the consonant approximation random set S^π is uniquely determined by the maximally normal plausibility assignment of a consistent random set.

Frequency Conversions:

$$\text{Prob_Dist} \xrightarrow{\text{has-a}} \text{Poss_Dist}, \quad \text{Poss_Dist} \xrightarrow{\text{has-a}} \text{Prob_Dist}. \quad (17)$$

Probability and possibility distributions are co-determining according to a wide variety of different methods [12].

Possibilistic Normalization:

$$\text{Random_Set} \xrightarrow{\text{has-a}} \text{Consistent_RS}, \quad \text{Pl_Assignment} \xrightarrow{\text{has-a}} \text{Poss_Dist}. \quad (18)$$

Even given an inconsistent random set, there are a variety of **possibilistic normalization methods** [9, 12] which allow for the construction of a consistent random set, and of a normal possibility distribution from a sub-normal plausibility assignment.

Measurement: The result of the possibilistic measurement methods developed by the author [8, 10], and described in Sec. 2.2, is the construction of a random set, hopefully consistent, from measured data. Therefore, while they require explicit implementation, they fall outside of the regular class hierarchy which has one root in the class **Random_Set**.

6.2 Other Extensions

There are further extensions to link the implementation of possibilistic methods with other GIT methods and CAST implementations. These extensions can be considered either as a part of this research program, or as extensions to the research programs which have been or may be launched by others.

- The CAST program originated with the implementations of deterministic finite state machines of Pichler and his colleagues [22]. There is a clear relation to the possibilistic approach described here, and the opportunity to generalize to a variety of different GIT-based representations of finite state machines with uncertainty, including nondeterministic and stochastic machines.
- There has been some work [2, 21] on the implementation of Klir's GSPS system [15]. It would certainly be very valuable to relate these efforts directly.
- In addition to the measurement methods mentioned above, **possibilistic clustering methods**, including **possibilistic c-means** [20] and the **mountain method** [1], can be integrated.
- While we have criticized the traditional dependence of possibility theory on fuzzy theory, their relation certainly cannot be ignored. And although it is not our specific focus, there is value in relating possibilistic implementations with those of the variety of fuzzy set operations and concepts. There is by now a huge literature on these methods (see Kosko [19] and Terano, Asai, and Sugeno [25] for just two examples), and many academic and corporate efforts to develop fuzzy theoretical systems. Hopefully other researchers are building CAST-based implementations of fuzzy systems methods, which could then be integrated with this specifically possibilistic system. In particular, as mentioned in Sec. 2, the action of possibilistic and stochastic processes is just a case of fuzzy relation composition, so a generalized fuzzy CAST system implementing fuzzy relations can be used as a base for possibilistic automata.
- Finally, possibility measures, as extreme plausibility measures, exist within the more general Dempster-Shafer evidence theory. The extension of possibility theoretical implementations to necessity measures (as the extreme belief measures dual to possibility measures), and general belief/plausibility pairs, may be very useful. Beyond that, both belief and plausibility measures are special fuzzy measures [27], and so the ultimate extension is to the construction of CAST-based systems for fuzzy measures in general.

7 Empirical Investigations

As noted in Sec. 3, it is common in systems theory that computer-based implementation and simulation are necessary in order to investigate the properties of the systems under consideration, and this is the case with possibilistic systems, processes, and models. There are a number of issues which it is desirable to investigate empirically.

Nonspecificity Calculations: Determination of informational properties, and in particular nonspecificity values and the changes in these values, is of great interest. This would include, for example:

- Calculation of $N(\pi^t)$ of the state vector of a possibilistic process as a function of t ;
- Calculation of $N(\pi)$ where π is a possibilistic histogram, and the dependence of $N(\pi)$ on the measurement method used;
- The change from $\mathbf{T}(\mathcal{S})$ to $\mathbf{T}(\pi)$ under possibilistic normalization (transforming an inconsistent random set \mathcal{S} to a consistent random set with possibility distribution π), and the dependence on both the general normalization method chosen and the various sub-choices required within some of the methods.
- Determination of $N(\pi)$ under the cases where π is a special fuzzy number [10], for example a parallelogram or triangle.

Possibilistic Processes: Aside from nonspecificity calculations, there are a number of other properties of possibilistic processes which require empirical investigation, for example the dependence of the form of possibilistic processes on the choice of norm operator \square and the choice of conditional possibility measure.

Measurement Methods: It is natural to explore the properties of the various measurement methods empirically, comparing the results of one data source using multiple methods.

Uncertainty Invariance Transformations: The Uncertainty Invariance Principle (UIP) [18] is a method for transformation among GIT structures on the basis of conservation of total uncertainty. While Klir and Parviz [17] have begun to empirically examine some results of frequency conversion methods, including the UIP, there are still many unanswered questions about the UIP. In the context of this work, it would be interesting to compare the time evolution of similar stochastic and possibilistic processes, and then compare those against their respective UIP transformations.

References

1. Barone, Joseph M; Filev, Dimitar P; and Yager, Ronald R: (1993) "Exponential Possibility Distributions and the Mountain Method", in: *Proc. NAFIPS 1993*, ed. M. McAllister, pp. 118-122, Allentown PA
2. Cellier, Francois: (1987) "Qualitative Simulation of Technical Systems Using the GSPS Framework", *Int. J. of General Systems*, v. 13:4, pp. 333-344
3. Coad, Peter and Yourdon, Edward: (1991) *Object-Oriented Analysis*, Prentice-Hall, Englewood Cliffs NJ
4. Dubois, Didier and Prade, Henri: (1988) *Possibility Theory*, Plenum Press, New York
5. Galway, L: (1986) "Workstation Based Environment for the Statistical Analysis of Set-Valued Data", in: *Computer Science and Statistics: Proceedings of the 18th Symposium on the Interface*, pp. 332-334
6. Horgan, John: (1993) "Death of Proof", *Scientific American*, v. 269:4, pp. 74-89

7. Janzen, Thomas E: (1993) "C++ Classes for Fuzzy Logic, Sets, and Associative Memories", *C User's Journal*, v. 11/93, pp. 55-71
8. Joslyn, Cliff: (1992) "Possibilistic Measurement and Set Statistics", in: *Proc. NAFIPS 1992*, v. 2, pp. 458-467, Puerto Vallarta
9. Joslyn, Cliff: (1993) "Empirical Possibility and Minimal Information Distortion", in: *Fuzzy Logic: State of the Art*, ed. R Lowen and M Roubens, pp. 143-152, Kluwer, Dordrecht
10. Joslyn, Cliff: (1993) "Some New Results on Possibilistic Measurement", in: *Proc. NAFIPS 1993*, pp. 227-231, Allentown PA
11. Joslyn, Cliff: (1994) "On Possibilistic Automata", in: *Computer Aided Systems Theory—EUROCAST '93*, ed. F. Pichler and R. Moreno-Díaz, pp. 231-242, Springer-Verlag, Berlin
12. Joslyn, Cliff: (1994) *Possibilistic Processes for Complex Systems Modeling*, PhD dissertation, SUNY-Binghamton, UMI Disseration Services, Ann Arbor MI
13. Joslyn, Cliff: (1995) "Towards an Independent Possibility Theory with an Objective Semantics", in: *Proc. 1995 Int. Workshop on Foundations and Applications of Possibility Theory*, to appear
14. Kennes, Robert: (1992) "Computational Aspects of Möbius Transformations of Graphs", *IEEE Trans. on Systems, Man, and Cybernetics*, v. 22:2, pp. 201-223
15. Klir, George: (1985) *Architecture of Systems Problem Solving*, Plenum, New York
16. Klir, George: (1991) *Facets of Systems Science*, Plenum, New York
17. Klir, George and Parviz, Behvad: (1992) "Probability-Possibility Transformations: A Comparison", *Int. J. of General Systems*, v. 21:1, pp. 291-310
18. Klir, George and Yuan, Bo: (1995) *Fuzzy Sets and Fuzzy Logic*, Prentice-Hall, New York
19. Kosko, B, ed.: (1990) *Neural Networks and Fuzzy Systems*, Prentice-Hall, Englewood Cliffs NJ
20. Krishnapuram, Raghu and Keller, James: (1993) "Possibilistic Approach to Clustering", *IEEE Trans. on Fuzzy Systems*, v. 1:2, pp. 98-110
21. Orchard, Robert: (1990) "Knowledge Processing: Semantics for the Klir Hierarchy of General Systems", in: *Computer Aided Systems Theory*, ed. Franz Pichler, pp. 33-40, Springer-Verlag, Berlin
22. Pichler, Franz and Praehofer, Herbert: (1988) "CAST:FSM - Computer Aided Systems Theory: Finite State Machine", in: *Cybernetics and Systems '88*, ed. R. Trappl, pp. 737-742, Kluwer, Boston
23. Sosnowski, Zenon A and Pedrycz, W: (1992) "FLISP: Representing and Processing of Uncertain Information", in: *Fuzzy Logic for the Management of Uncertainty*, ed. L Zadeh and J Kacprzyk, pp. 495-521, Wiley, New York
24. Stroustrup, Bjarne: (1991) *C++ Programming Language*, Addison Wesley
25. Terano, Toshiro; Asai, Kiyoji; and Sugeno, Michio: (1987) *Fuzzy Systems Theory and Its Applications*, Academic Press, New York
26. Thoma, H Mathis: (1991) "Belief Function Computations", in: *Conditional Logic in Expert Systems*, ed. IR Goodman et al., pp. 269-308, North Holland
27. Wang, Zhenyuan and Klir, George J: (1992) *Fuzzy Measure Theory*, Plenum Press, New York
28. Zadeh, Lotfi A: (1978) "Fuzzy Sets as the Basis for a Theory of Possibility", *Fuzzy Sets and Systems*, v. 1, pp. 3-28
29. Zeigler, BP: (1990) *Object-Oriented Simulation with Hierarchical Modular Models*, Academic Press, San Diego

Fuzzy Expert System Technology

I.B. Türkşen

Principal Investigator
Intelligent Systems Laboratory
Department of Industrial Engineering
University of Toronto, Toronto
Ontario, M5S 1A4, Canada

Abstract. Fuzzy expert systems are the second generation expert systems. In real world activities, system states have various attributes that at best could be assessed as a matter of degree. If we express our knowledge of system states in terms of an all or nothing paradigm then much valuable information does not appear in system's models. However, if we express our knowledge of systems with fuzzy sets, much valuable information stays embedded in the rules of behaviour in system's models. Thus, the fuzzy set paradigm gives our knowledge of systems an effective representation.

For this purpose, four levels of knowledge representation are identified as: (i) linguistic, (ii) meta-linguistic, (iii) propositional, and (iv) computational. In particular, four classes of fuzzy rule base schema are identified within a unified framework of implication statements. That is we identify at least four classes of propositional expressions that represent linguistic expression of rules. Interval valued fuzzy sets are defined as particular Type II fuzzy sets in order to expose second order semantic uncertainty embedded in linguistic expressions.

1. Introduction

Generally, experts communicate their knowledge of a particular systems' behaviour via linguistic expressions of a natural language. Systems analysts, on the other hand, transform these linguistic expressions with their interpretive skills into certain "meta-linguistic" expressions, such as logical, mathematical, graphical, etc., in order to develop a particular analysis of a context dependent system's description. Next, such meta-linguistic expressions are transformed into certain "propositional expressions" again based on some interpretations and assumptions. Finally, propositional expressions are transformed into certain "computational expressions". At the end, after an appropriate analysis of computational expression, we usually need to return to linguistic expressions in order to interpret the "meaning" representation that is embedded in these computational expressions. This last step is needed in order to verify that an expert's knowledge is transformed properly to reflect a system's behaviour pattern. This cycle may be repeated a number of times until an expert's knowledge is properly encoded in computational expression through the transformation between the four levels of knowledge representation.

In this paper, we discuss a particular subset of these four levels of knowledge representation through fuzzy sets and logics paradigm.