INTELLIGENT INFORMATION FUSION IN BATTLEFIELD SENSOR NETWORKS

RESEARCH UPDATE

STEVE HENDERSON

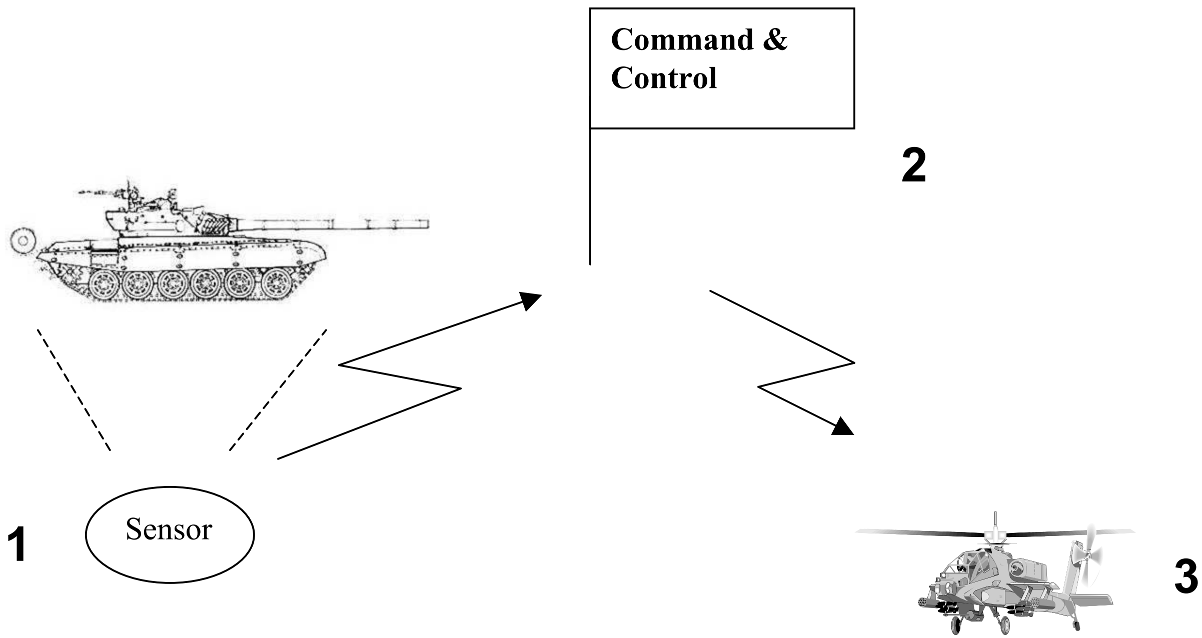
7 FEBRUARY, 2003

AGENDA

- Problem Background (3 min)
- Sensor Information Fusion Meta-Model (5 min)
- Modal Logic Applied to Meta-Model (15 min)
- Bayesian Belief Nets Applied to Meta-Model (15 min)
- Simulation Demo and Results from BBN Application (10 min)
- Future Research Objectives (2 min)
- Summary
- Questions
- Conclusion

PROBLEM BACKGROUND

<u>Overall Problem Statement</u>: Shorten the target acquisition to engagement life cycle.



Traditional Target Acquisition Sequence

1- Sensors detects targets

2- Command gets information – determines what is really happening

3- Command tells shooters to move and kill

<u>Problems:</u>

Command and control processing delay

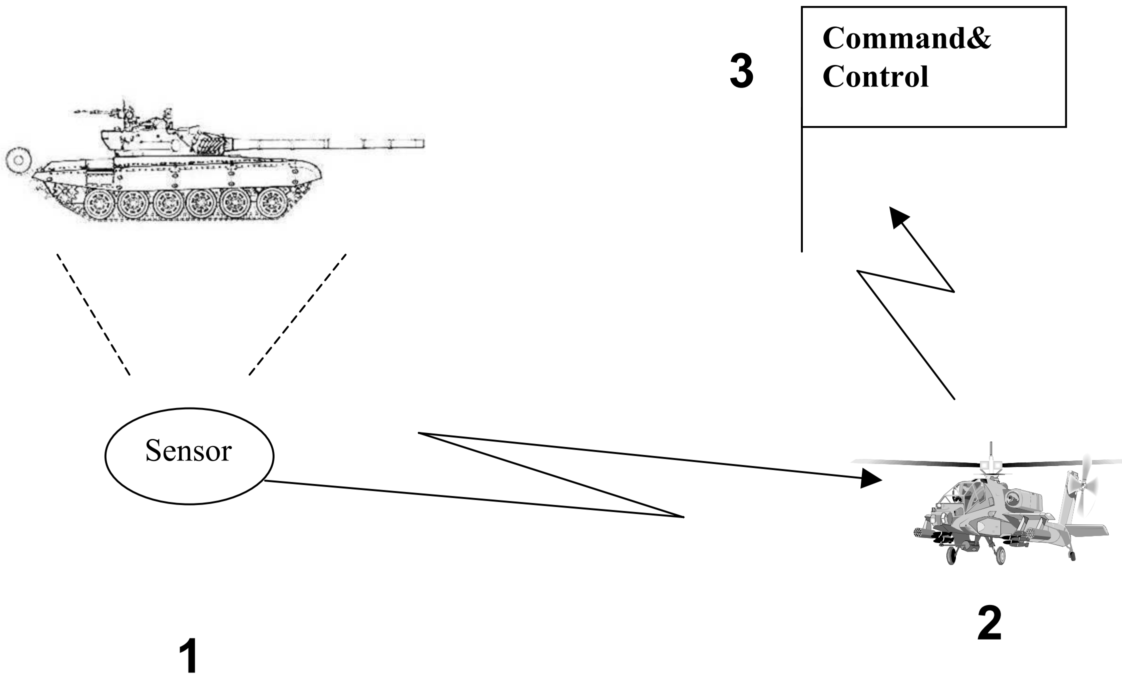
Command and control info overload

Fusion of information becomes difficult with more information

PROBLEM BACKGROUND

Solution: Move command and control into a supervisory role and let sensors communicate directly with shooters.

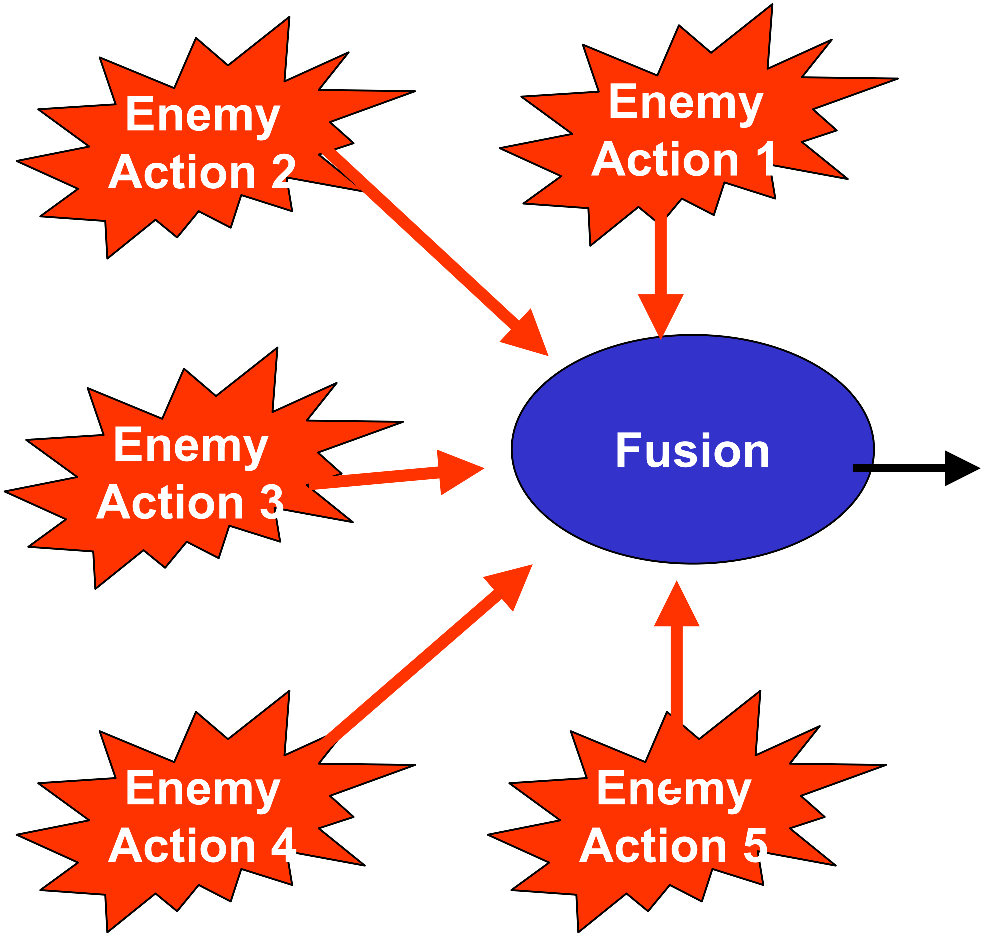
This approach is part of a large system known as the Future Combat System (FCS).



Revised Sequence

1. Sensors detects targets
2. Sensors request optimal shooter(s)
3. Command element monitors exchange

THE NEED TO AUTOMATE INFORMATION FUSION



Fusion Goal:

What is enemy really doing?

- *This question is key to directing shooters to the right place*

- *How can a semi-automated process perform this task?*

PROPOSED INFORMATION FUSION META-MODEL

- Fusion Model M is 5-tuple - (S, V, K, O, E)

S - Set of m ***sensors***, $\{s_1, s_2, \ldots, s_m\}$.

V - Set of n possible ***sensor values***, $\{v_1, v_2, .., v_n\}$.

When sensor s_i assumes the value v_j, we denote this as $s_i(v_j)$.

K - Finite set of q ***key descriptors***, $\{kd_1, kd_2, \ldots, kd_q\}$.

kd_i is defined as a function $f = \{ (x, p) : x \subseteq S \times V,\ p \in \mathcal{R} \}$ that maps subsets of possible sensor-value pairs to a real value p.

The real value p is certainty/truth-value of key descriptor based on the presence of its associated sensor values.

PROPOSED INFORMATION FUSION META-MODEL

- Fusion Model M is 5-tuple - (S, V, K, O, E) (continued)

S - Finite set of m ***sensors***, $\{s_1, s_2, \ldots, s_m\}$.
V - Finite number of n possible ***sensor values***, $\{v_1, v_2, .., v_n\}$.
K - Finite set of q ***key descriptors***, $\{kd_1, kd_2, \ldots, kd_q\}$.

O - Finite set of t mutually exclusive ***operational states***.

An operational state represents the force's overall behavior

Each operational state is a function $o_i = \{ (y, p) : y \subseteq K, \ , \ p \in \mathcal{R} \}$, that maps subsets of key descriptor values to a value p in $\mathcal{R}$.

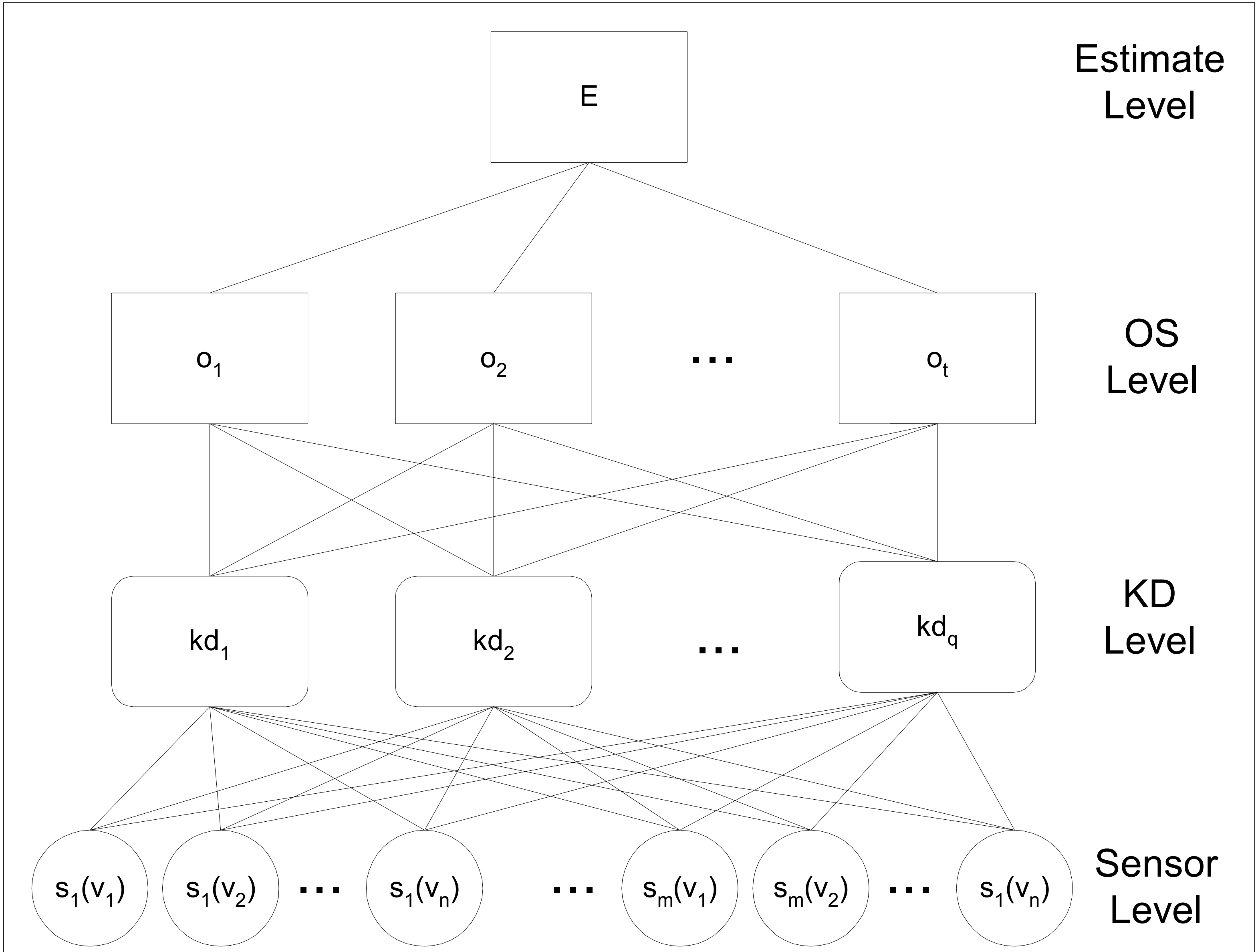
The value p represents the certainty associated with an operational state.

E - Operational state ***estimation*** function

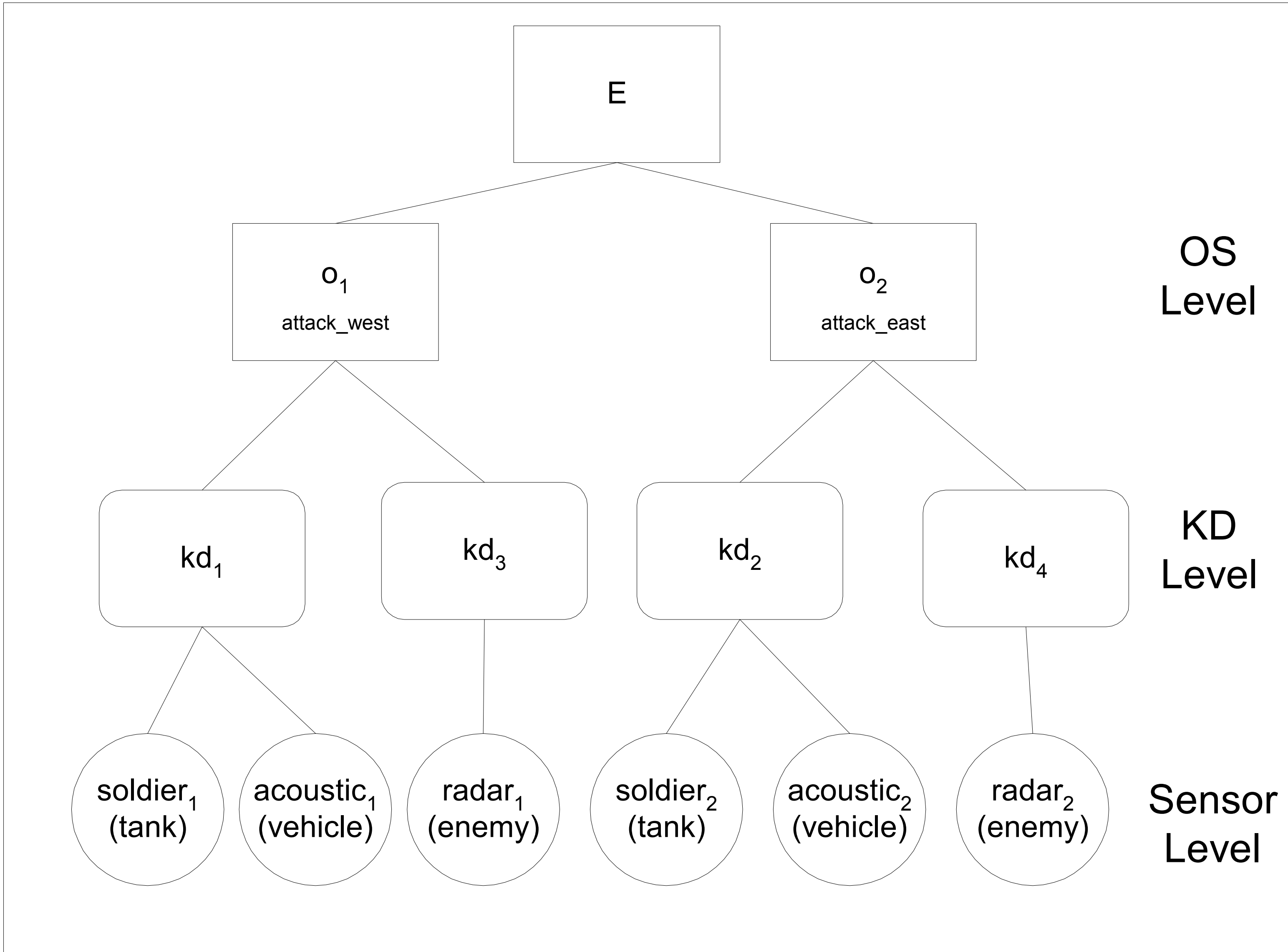
$$E = \{ (z, e) : z \subseteq O, \ e = \{ e_1, e_2, \ldots, e_t\}, e_i \in \mathcal{R}\}$$

Each value e_i in the vector e corresponds to the overall certainty of an operational state o_i when evaluated in the context of *all operational states*.

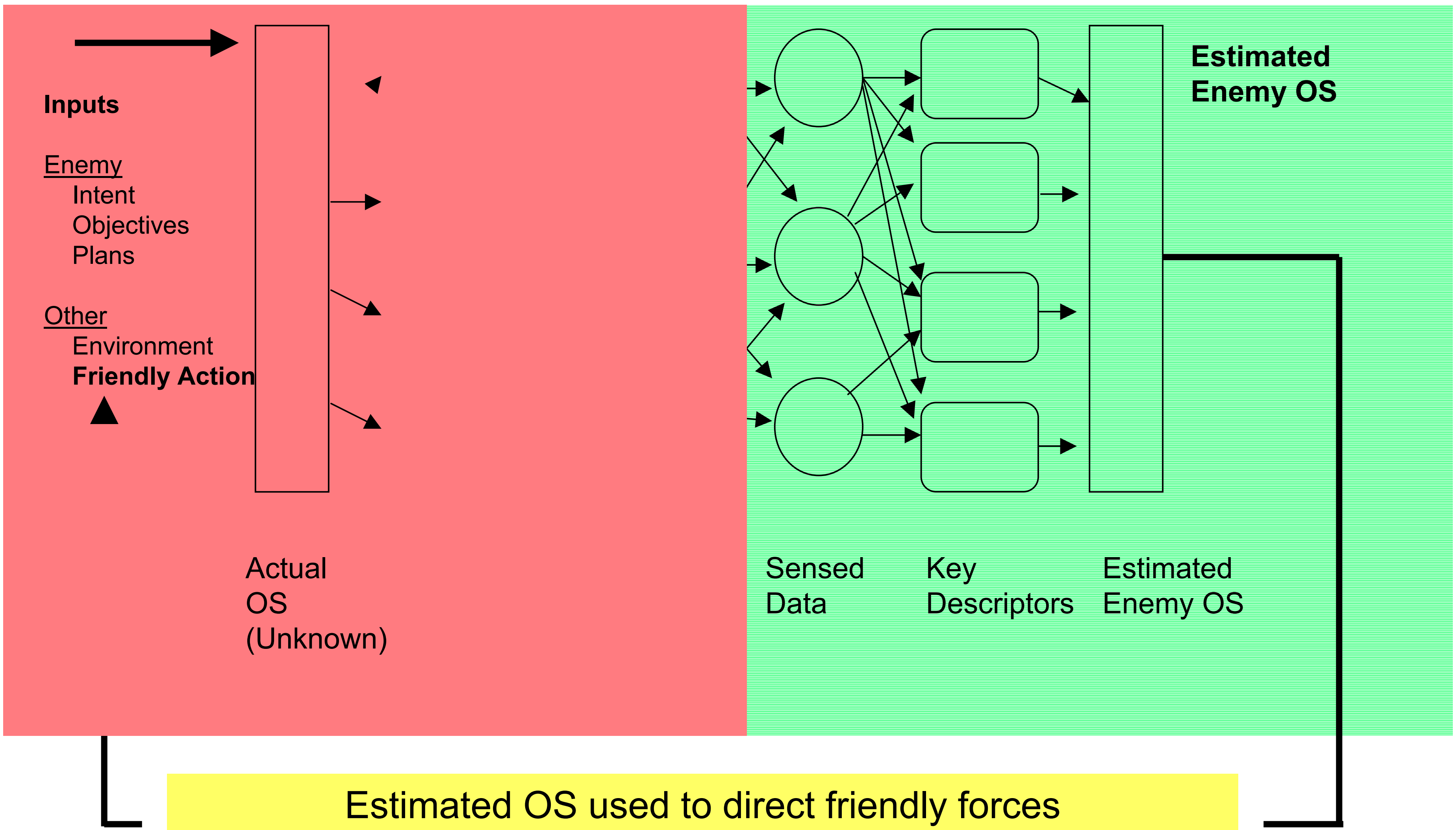
INFORMATION FUSION META-MODEL



EXAMPLE INFORMATION FUSION META-MODEL

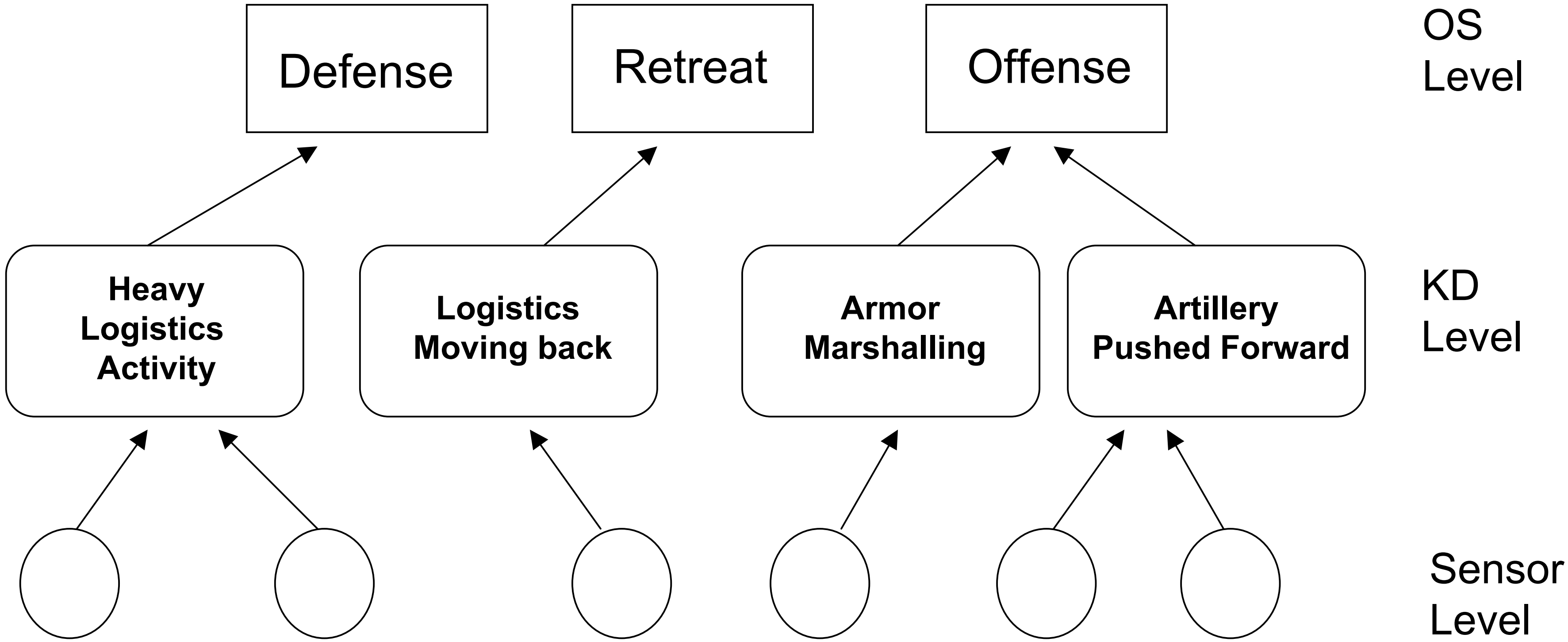


ESTIMATION PROCESS



Estimated OS used to direct friendly forces
against actual enemy OS in order to disrupt/defeat

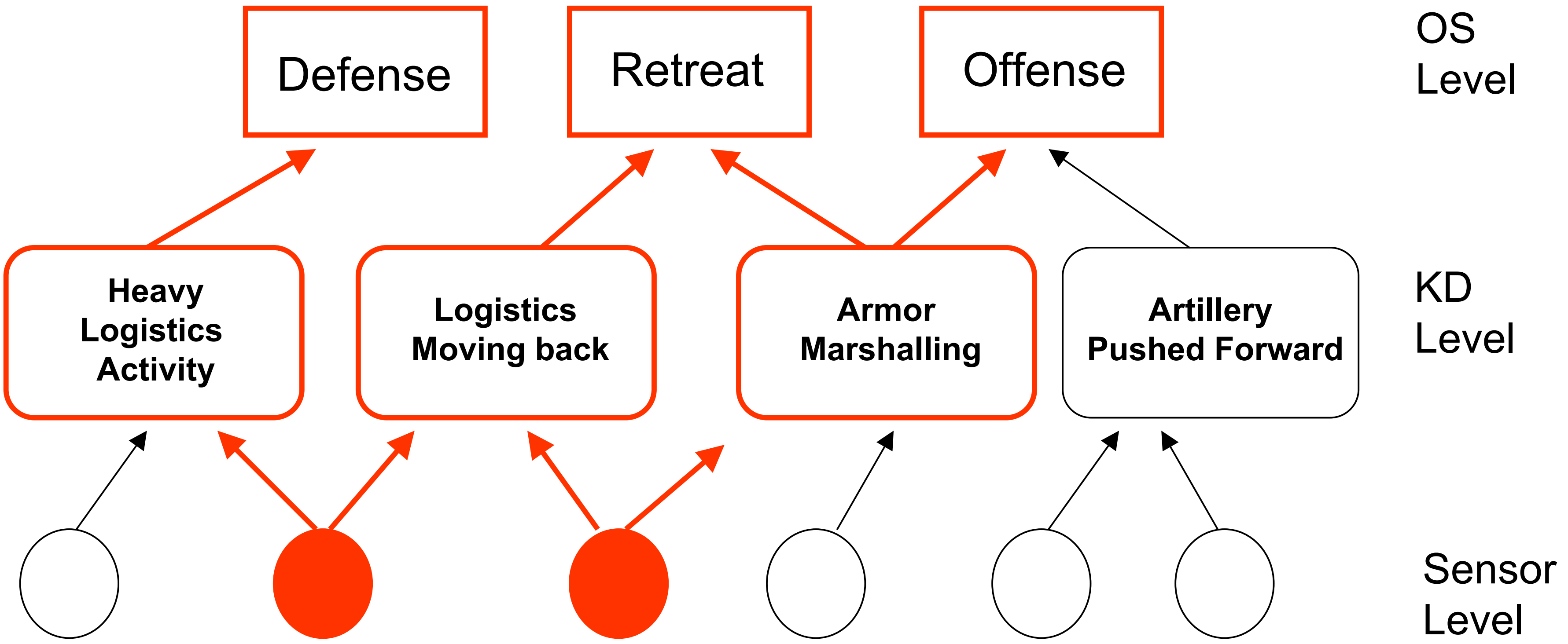
INFORMATION FUSION DIFFICULTIES



Perfect World:

- minimum set of sensors that define each key descriptor are mutually exclusive
- minimum set of key descriptors that define each EOS are mutually exclusive
- detecting enemy intent (EOS) is an exercise in mapping sensor data to KD and then KD to EOS

INFORMATION FUSION DIFFICULTIES



However:

- many types of sensor data and KDs might support more than one parent (shown above in red)

- Sensors are matched with multiple KDs and multiple KDs are matched with multiple EOS's

- This creates a level of propagating uncertainty that clouds interpretation of actual enemy intent

INFORMATION FUSION DIFFICULTIES

Potential methods to mitigate uncertainty:

- Design framework to be mutually exclusive

  - might not be possible

  - eliminates the potential benefits of overlapping information (information reuse, collection/processing time, search pruning, etc.)

  - could require potentially more information to reach EOS clarity

- Design framework to <u>minimize overlap and reason about uncertainties</u>

  - allows for more complete representation of similar EOS's

  - reasoning about uncertainty provides EOS clarity while still allowing for similar KDs and EOS's to share some common information.

INFORMATION FUSION USING PROBABILISTIC MODAL LOGIC

PROBABILISTIC MODAL LOGIC (PML) REVIEW

- Adds degree of certainty to Modal Logic (Halpern, 1987)
- Modal operators include probability assignments

- Additional Logic Operators (beyond first order logic):

  $pr_i(x)$: The total probability of x across all accessible worlds from agent i's perspective

- Probability Kripke Structure: ***(W, PR, V)***

  W = set of worlds or states

  PR = { PR_1, PR_2, … PR_n } n = number of agents
  PR_i = probability based relationship between states according to agent i

  V = truth values assigned to predicates inside state S

- From each agent, the sum of probabilities across all possible worlds is 1

- Probability operator can be used to make inferences about predicates

MODAL LOGIC OVERVIEW

- **Example:**

  Two players flip coins. Before they see each others coin, they must either elect to bet that they have the winning hand or fold. Player's who fold forfeit a small ante fee.

- The winner is then decided according to the following table:

P1 coin	P2 coin	Winner
H	H	P2
H	T	P1
T	T	P1
T	H	P2

MODAL LOGIC OVERVIEW

- **Example (continued):**

Lets say player one rigs player two's coin so it lands tail's side up 70% of the time.

Model M = { W, PR, V }:

W = { (H/H) (H/T) (T/T) (T/H) }

PR_1 = { PR_1[(H/H), (H/T)] = 0.7
PR_1[(H/H), (H/H)] = 0.3
PR_1[(T/T), (T/H)] = 0.3
PR_1[(T/T), (T/T)] = 0.7
PR_1[(T/H), (T/H)] = 0.3
PR_1[(T/H), (T/T)] = 0.7
PR_1[(H/T), (H/T)] = 0.7
PR_1[(H/T), (H/H)] = 0.3 }

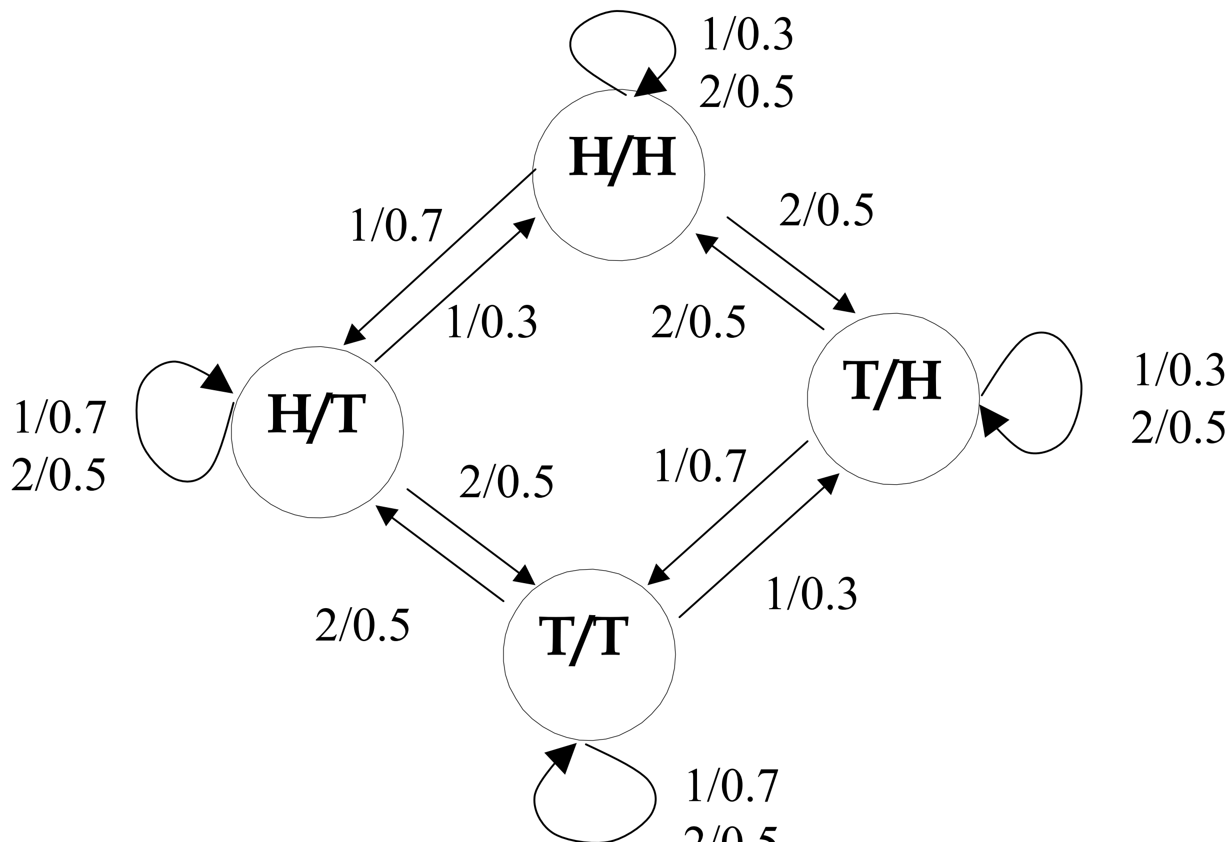
PR_2 = { PR_2[(H/H), (H/T)] = 0.5
PR_2[(H/H), (H/H)] = 0.5
PR_2[(T/T), (T/H)] = 0.5
PR_2[(T/T), (T/T)] = 0.5
PR_2[(H/T), (H/T)] = 0.5
PR_2[(H/T), (H/H)] = 0.5
PR_2[(T/H), (T/H)] = 0.5
PR_2[(T/H), (T/T)] = 0.5 }

Notational Note: PR[(H/H) , (H/T)] = 0.7 indicates that if player 1 is in state (H/H), he will tend to think with probability (0.7) that he is in state (H/T). He knows his coin is H, and thinks that the other player's head is likely tails, because he rigged it.

V = ((H,H) $\models$ winner(P2) , (H,T) $\models$ winner(P1) ,
(T,T) $\models$ winner(P1) , (T,H) $\models$ winner(P2))

MODAL LOGIC OVERVIEW

Kripke Model:

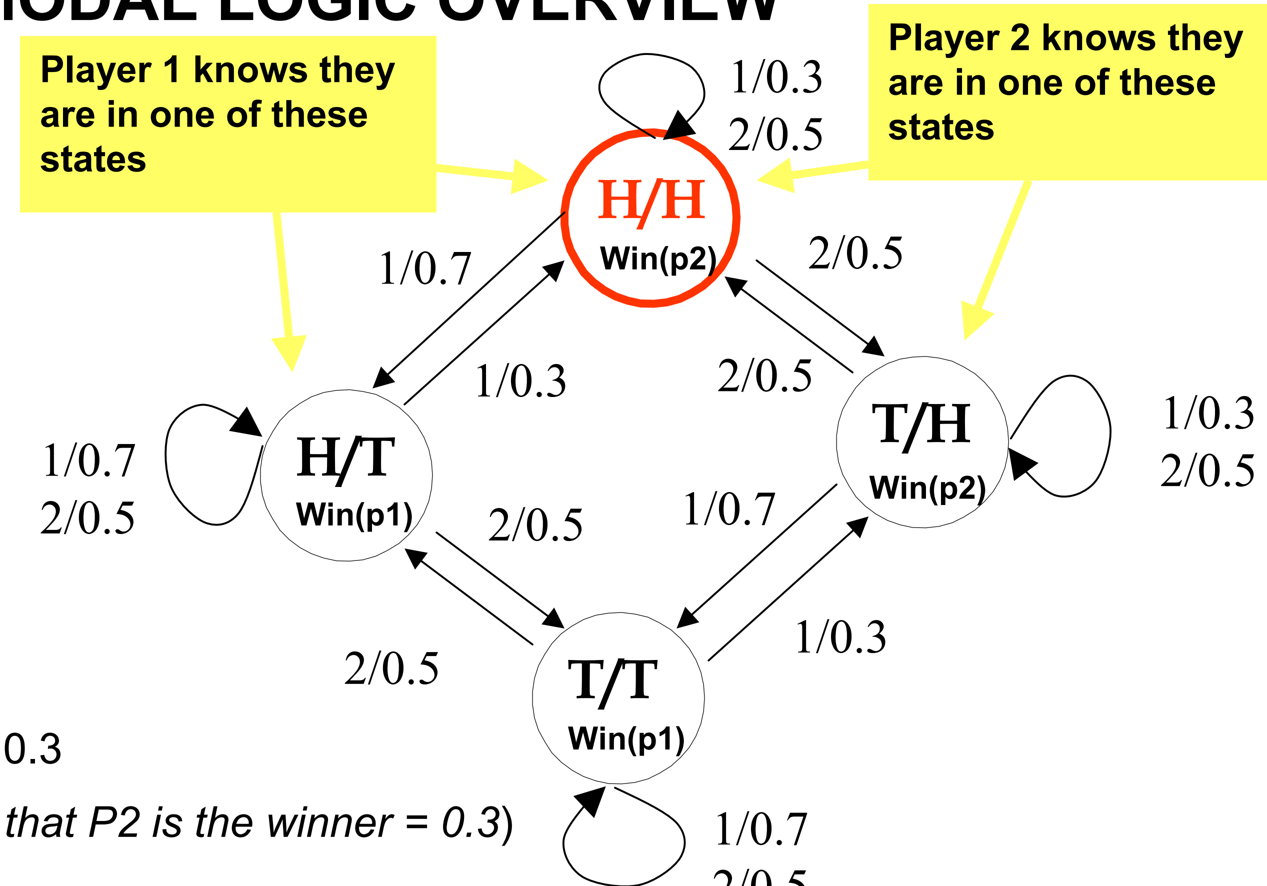


MODAL LOGIC OVERVIEW

Kripke Model:

Let say each player flips heads….

P1 coin	P2 coin	Winner
H	H	P2
H	T	P1
T	T	P1
T	H	P2



Possible logical inferences:

(H,H) $\models$ pr $_1$ [winner (P2)] = 0.3

(*in P1's view, the probability that P2 is the winner = 0.3*)

$\exists$(x) | (H,H) $\models$ pr $_1$ [winner (x)] > 0.6

(*in P1's view there is a player (i.e. P1) who is more likely to be the winner*)

(H,H) $\models$ pr $_2$ [winner (P2)] = 1.0

(*in P2's view, the probability that P2 is the winner is 1*)

PROBABILISTIC MODAL LOGIC (PML) w/ PROBABILITY VALUES

- This combination is still very theoretical
- Probability Kripke Structure: ***(W, PR, V)***

W = set of worlds or states
PR = probability based relationship between states

V = truth values assigned to predicates inside state S

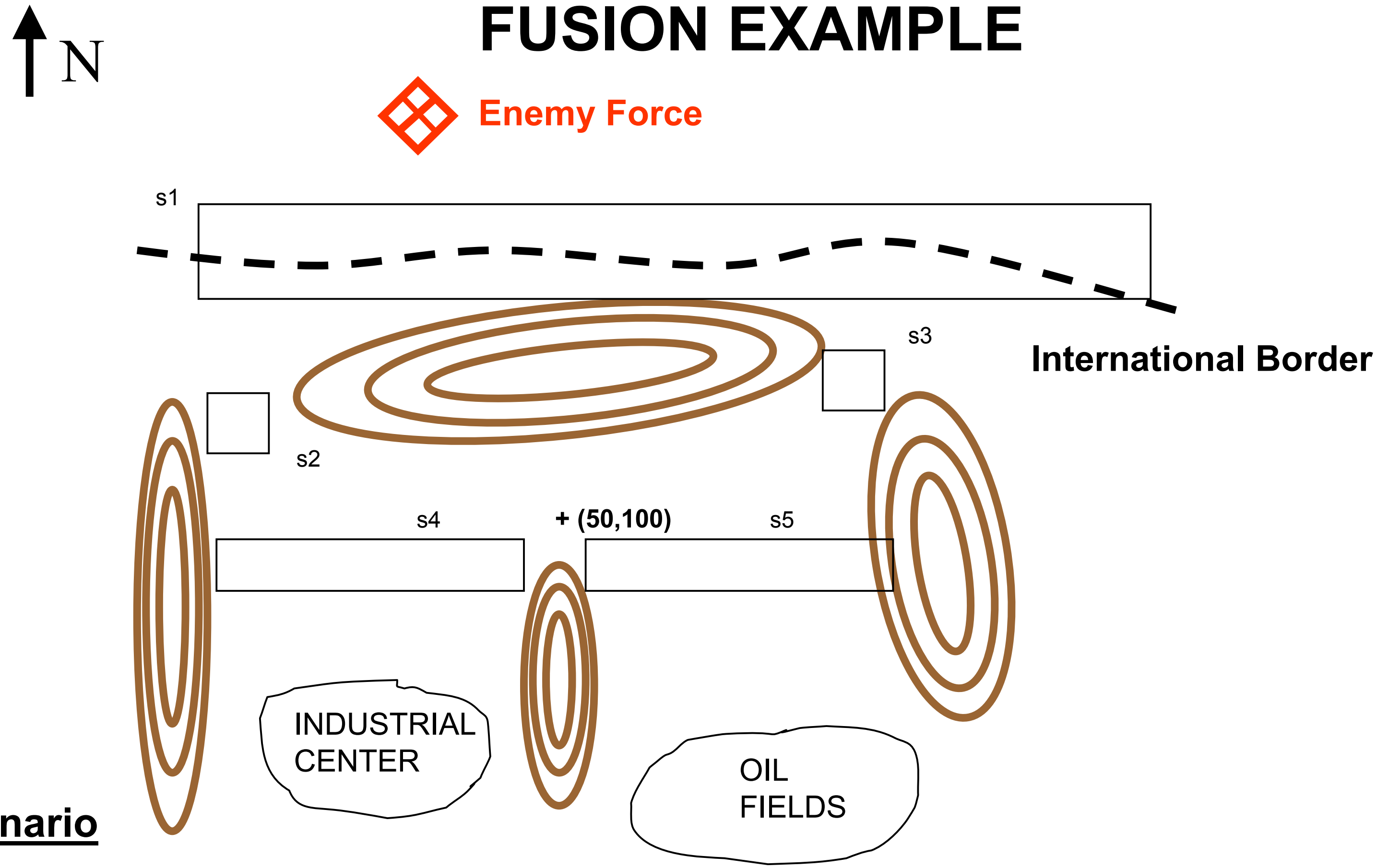
V specifies a probability value (p) to every truth value.

<u>Example</u>:

V(S1) = $\quad$ p(DOG $\Rightarrow$ p(FLEAS))= 0.8

In state 1, DOG implies FLEAS with probability value 0.8

FUSION EXAMPLE

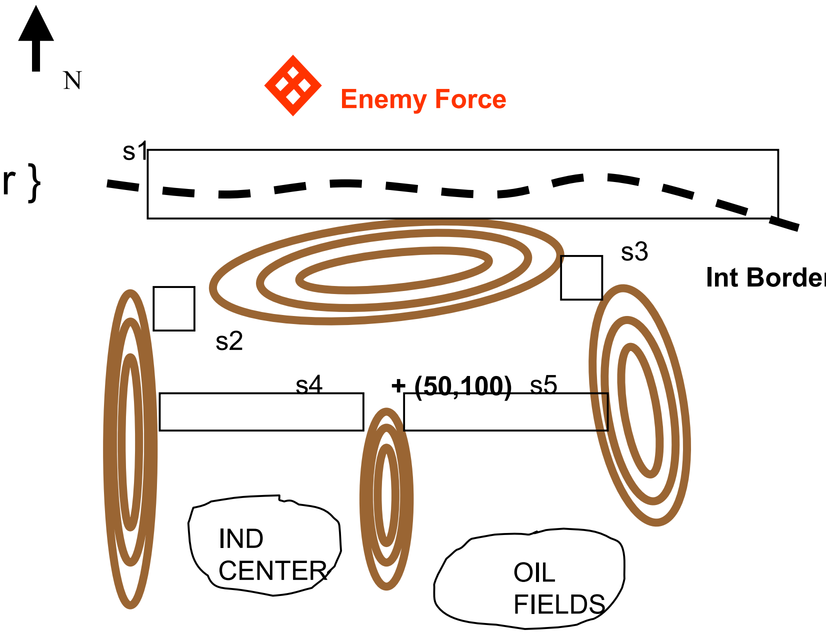


<u>Scenario</u>

- Enemy forces massing on international border
- Enemy has desire and ability to seize industrial center **OR** oil fields
- What will the enemy do?

  (a) Minor incursions to try and draw us into firing first shots (provoke)
  (b) Attack to seize industrial center (attack west)
  (c) Attack to seize oil fields (attack east)

EXAMPLE INFORMATION FUSION META-MODEL



M = { S, V, K, O , E }
S = { S1, S2 , S3 , S4 , S5 }
V = { x where x = number of tanks read by sensor }
K = { $kd_1 = (S1 > 0)$
$kd_2 = (S2 > 0) \cup (S3 > 0)$
$kd_3 = (S4 > 0) \cup (S5 > 0)$
$kd_4 = (S4 > 0) \cap (S5 = 0)$
$kd_5 = (S4 = 0) \cap (S5 > 0)$
$kd_6 = centerX(s1, s2, \ldots, s5) > 50 \cap centerY(s1, s2, \ldots, s5) > 100$
$kd_7 = centerX(s1, s2, \ldots, s5) <= 50 \cap centerY(s1, s2, \ldots, s5) > 100$ }

Note:

$$centerX(s1, s2, \ldots, s5) = \frac{1}{\sum s_i} * \sum location_x(s_i) * s_i$$

$$centerY(s1, s2, \ldots, s5) = \frac{1}{\sum s_i} * \sum location_y(s_i) * s_i$$

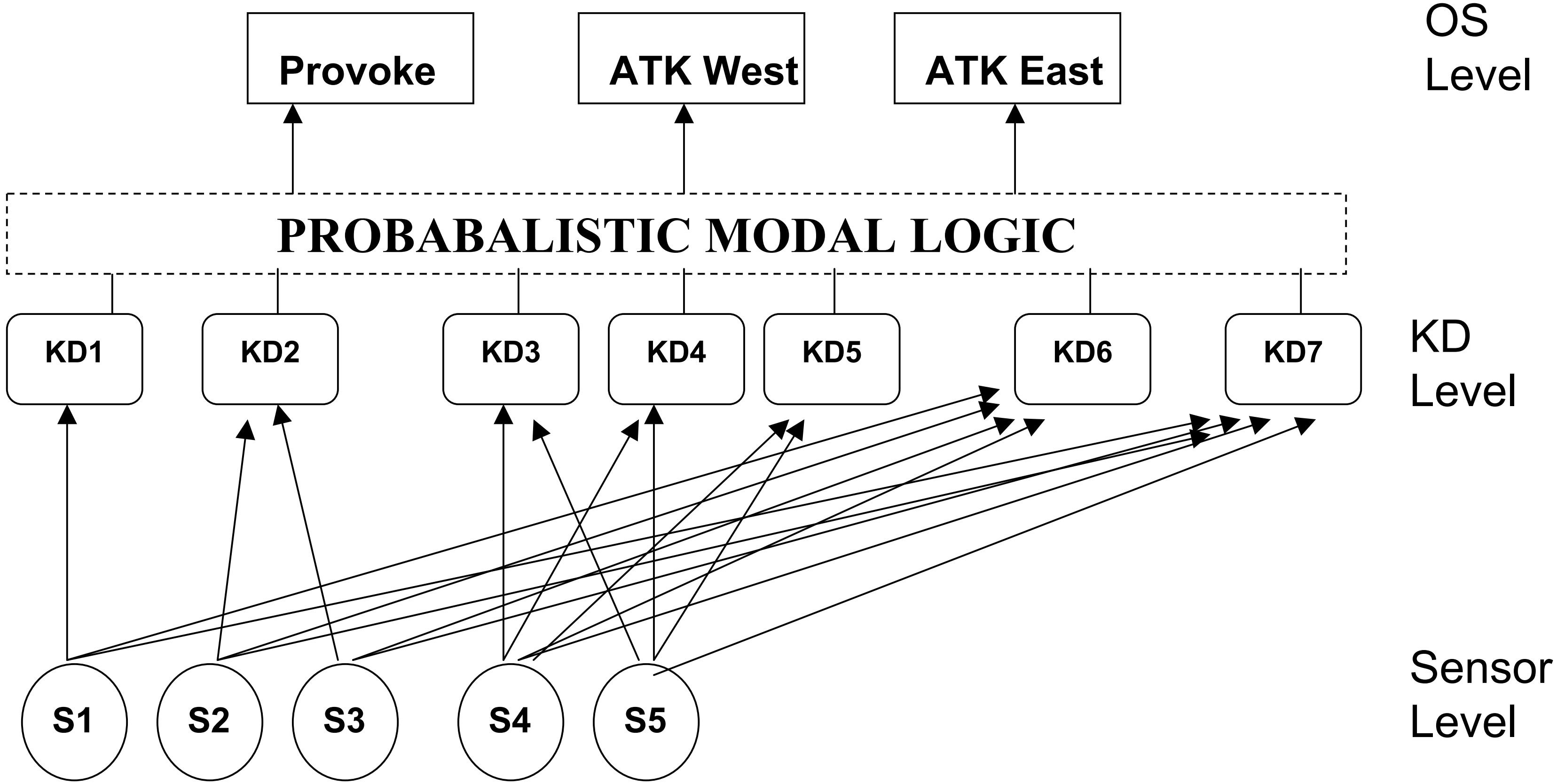
O = { provoke, attack_west, attack_east }
*Each of these states is a function that will be calculated by fusion process

E = O (Our final estimate depends only on probabilities associated w/ each state)

EXAMPLE INFORMATION FUSION META-MODEL

S = { S1, S2 , S3 , S4 , S5 }
V = { x where x = number of tanks read by sensor }
K = { kd_1 = (S1 > 0)
kd_2 = (S2 > 0) ∪ (S3 > 0)
kd_3 = (S4 > 0) ∪ (S5 > 0)
kd_4 = (S4 > 0) ∩ (S5 = 0)
kd_5 = (S4 = 0) ∩ (S5 > 0)
kd_6 = centerX(s1, s2, …, s5) > 50 ∩ centerY(s1, s2, …, s5) > 100
kd_7 = centerX(s1, s2, …, s5) <= 50 ∩ centerY(s1, s2, …, s5) > 100 }

O = { provoke, attack_west, attack_east } E = O



EXAMPLE PML MODEL

PML model R = { S, PR, V}

S= the **current** state of the enemy = { OS1 = provoke
OS2 = attack_west
OS3 = attack_east }

PR = { PR(OSi, OSj) = w, for i=1..3, j=1..3

w = probability that, if you think you are state i, you are really in state j
i.e. – indistinguishable states

(See next slide for w values)

}

V = { V_{s1}, V_{s2}, V_{s3} }

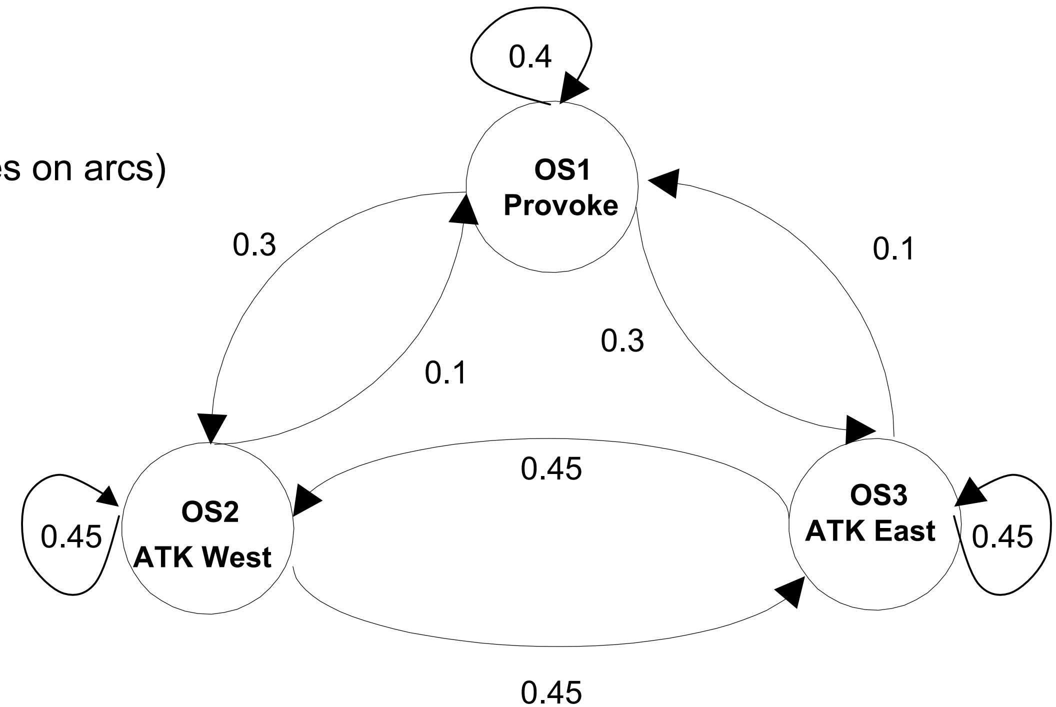
V is the set of truth values for ***each world***. These truth values will be in the following form:

$$kd_i\ (x,y) \Rightarrow p(NS(z))$$

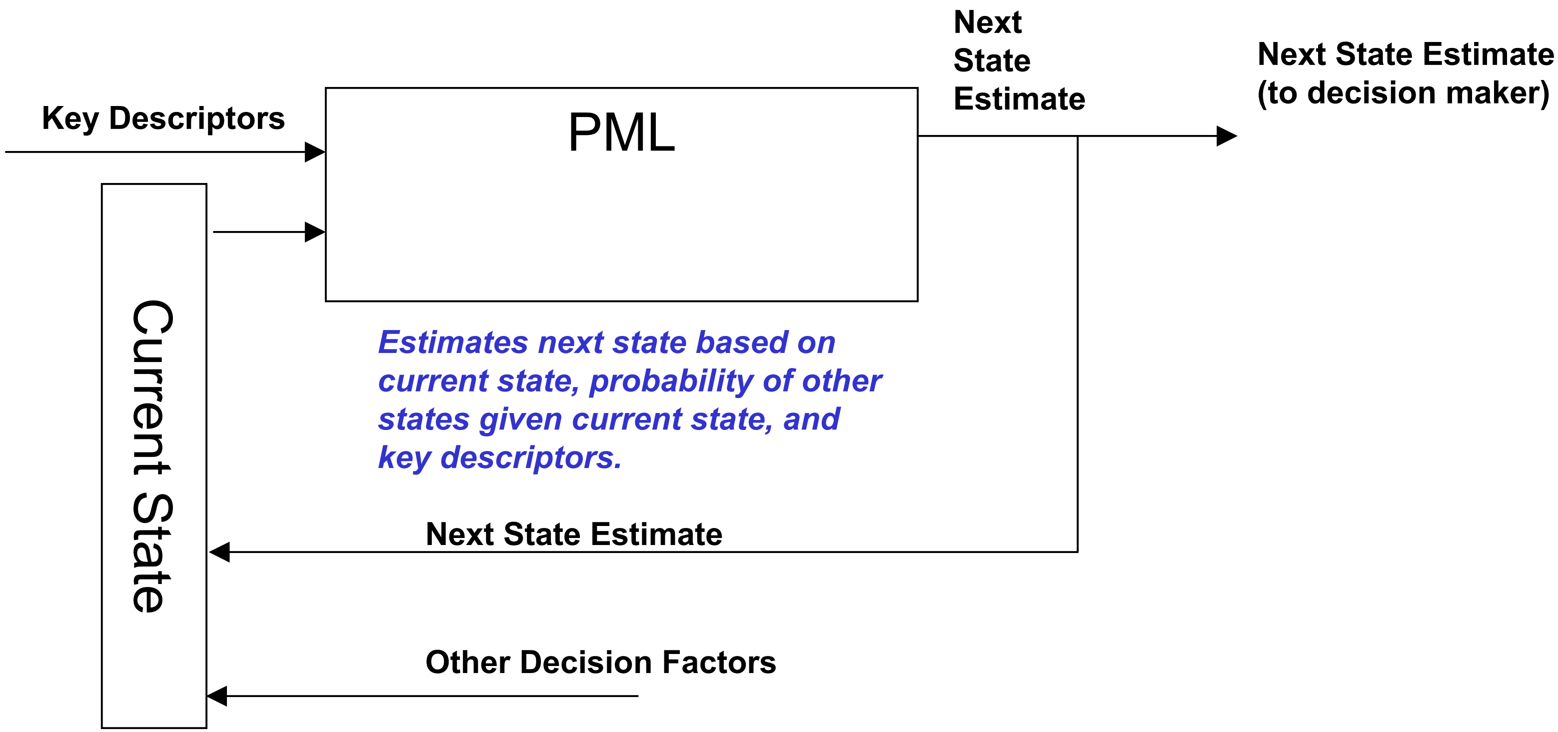
- kd_i will be taken from our set of KDs
- NS(z) indicates that the enemy’s next state is state Z (what we want to know)
- p(NS(z)) indicates probability that next state is in fact state Z

EXAMPLE PML MODEL

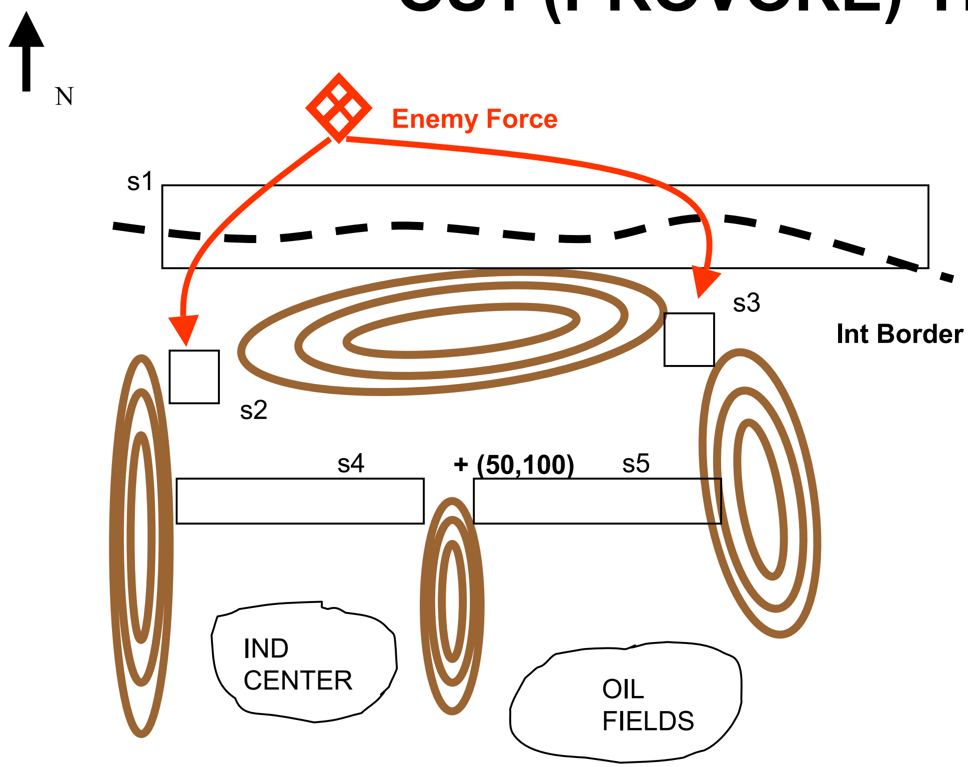
(w values on arcs)



MODAL LOGIC INFORMATION FUSION PROCESS



OS1 (PROVOKE) TRUTH VALUES



$kd_1 = (S1 > 0)$

$kd_2 = (S2 > 0) \cup (S3 > 0)$

$kd_3 = (S4 > 0) \cup (S5 > 0)$

$kd_4 = (S4 > 0) \cap (S5 = 0)$

$kd_5 = (S4 = 0) \cap (S5 > 0)$

$kd_6 = \text{centerX}(s1, s2, \ldots, s5) > 50 \cap$
$\text{centerY}(s1, s2, \ldots, s5) > 100$

$kd_7 = \text{centerX}(s1, s2, \ldots, s5) <= 50 \cap$
$\text{centerY}(s1, s2, \ldots, s5) > 100 \; \}$

<u>V(OS1)</u>

KD1 $\Rightarrow$ $(p(NS(OS1)) = 0.8) \cap (p(NS(OS2)) = 0.1) \cap (p(NS(OS3)) = 0.1)$

KD2 $\Rightarrow$ $(p(NS(OS1)) = 0.4) \cap (p(NS(OS2)) = 0.6) \cap (p(NS(OS3)) = 0.6)$

KD3 $\Rightarrow$ $(p(NS(OS1)) = 0.4) \cap (p(NS(OS2)) = 0.8) \cap (p(NS(OS3)) = 0.8)$

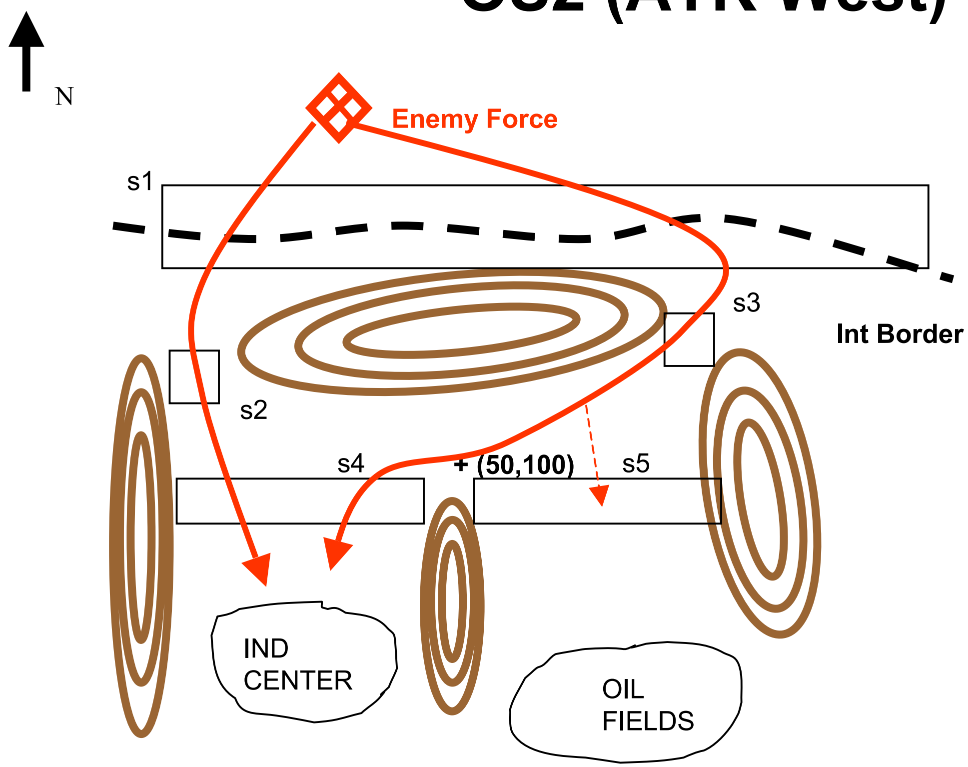
KD4 $\Rightarrow$ $(p(NS(OS1)) = 0.3) \cap (p(NS(OS2)) = 0.7) \cap (p(NS(OS3)) = 0.3)$

KD5 $\Rightarrow$ $(p(NS(OS1)) = 0.3) \cap (p(NS(OS2)) = 0.3) \cap (p(NS(OS3)) = 0.7)$

KD6 $\Rightarrow$ $(p(NS(OS1)) = 0.1) \cap (p(NS(OS2)) = 0.3) \cap (p(NS(OS3)) = 0.9)$

KD7 $\Rightarrow$ $(p(NS(OS1)) = 0.1) \cap (p(NS(OS2)) = 0.9) \cap (p(NS(OS3)) = 0.3)$

OS2 (ATK West) Truth Values



kd_1 = (S1 > 0)
kd_2 = (S2 > 0) ∪ (S3 > 0)
kd_3 = (S4 > 0) ∪ (S5 > 0)
kd_4 = (S4 > 0) ∩ (S5 = 0)
kd_5 = (S4 = 0) ∩ (S5 > 0)
kd_6 = centerX(s1, s2, …, s5) > 50 ∩
centerY(s1, s2, …, s5) > 100

kd_7 = centerX(s1, s2, …, s5) <= 50 ∩
centerY(s1, s2, …, s5) > 100 }

<u>V(OS2)</u>

KD1 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 1.0) ∩
(p(NS(OS3)) = 0.4)

KD2 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 1.0) ∩
(p(NS(OS3)) = 0.4)

KD3 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 1.0) ∩
(p(NS(OS3)) = 0.4)

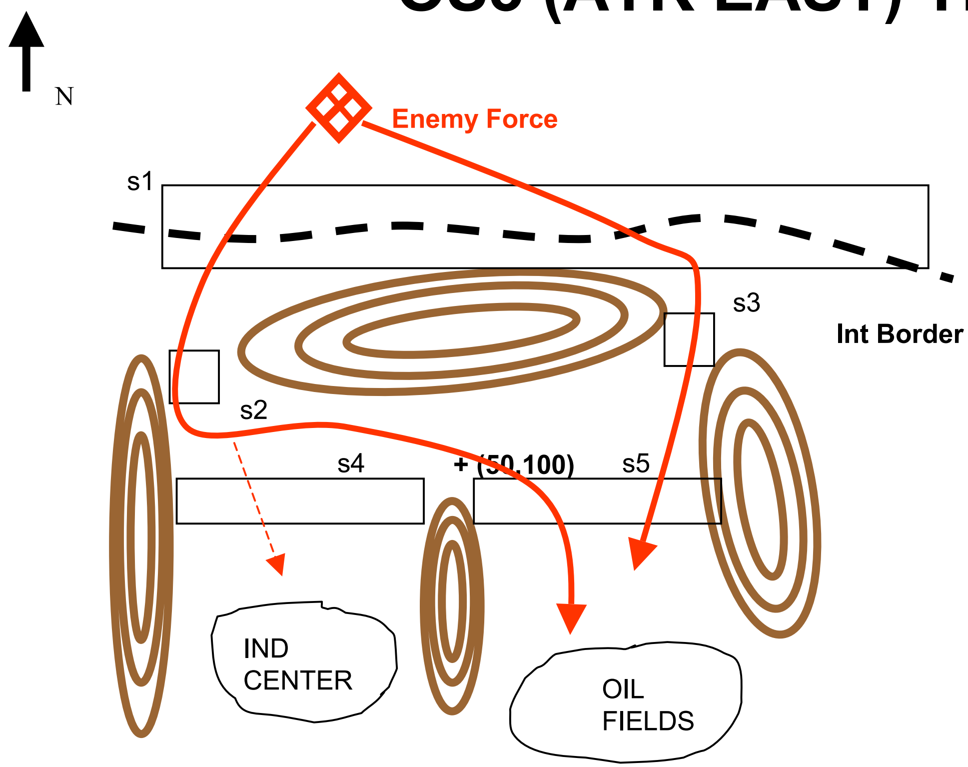
KD4 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 1.0) ∩
(p(NS(OS3)) = 0.4)

KD5 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 0.4) ∩
(p(NS(OS3)) = 1.0)

KD6 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 0.3) ∩
(p(NS(OS3)) = 1.0)

KD7 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 0.3) ∩
(p(NS(OS3)) = 1.0)

OS3 (ATK EAST) TRUTH VALUES



$kd_1 = (S1 > 0)$
$kd_2 = (S2 > 0) \cup (S3 > 0)$
$kd_3 = (S4 > 0) \cup (S5 > 0)$
$kd_4 = (S4 > 0) \cap (S5 = 0)$
$kd_5 = (S4 = 0) \cap (S5 > 0)$
$kd_6 = \text{centerX}(s1, s2, \ldots, s5) > 50 \cap \text{centerY}(s1, s2, \ldots, s5) > 100$

$kd_7 = \text{centerX}(s1, s2, \ldots, s5) <= 50 \cap \text{centerY}(s1, s2, \ldots, s5) > 100 \}$

V(OS3)

KD1 $\Rightarrow$ (p(NS(OS1)) = 0.2) $\cap$ (p(NS(OS2)) = 0.4) $\cap$ (p(NS(OS3)) = 1.0)

KD2 $\Rightarrow$ (p(NS(OS1)) = 0.2) $\cap$ (p(NS(OS2)) = 0.4) $\cap$ (p(NS(OS3)) = 1.0)

KD3 $\Rightarrow$ (p(NS(OS1)) = 0.2) $\cap$ (p(NS(OS2)) = 0.4) $\cap$ (p(NS(OS3)) = 1.0)

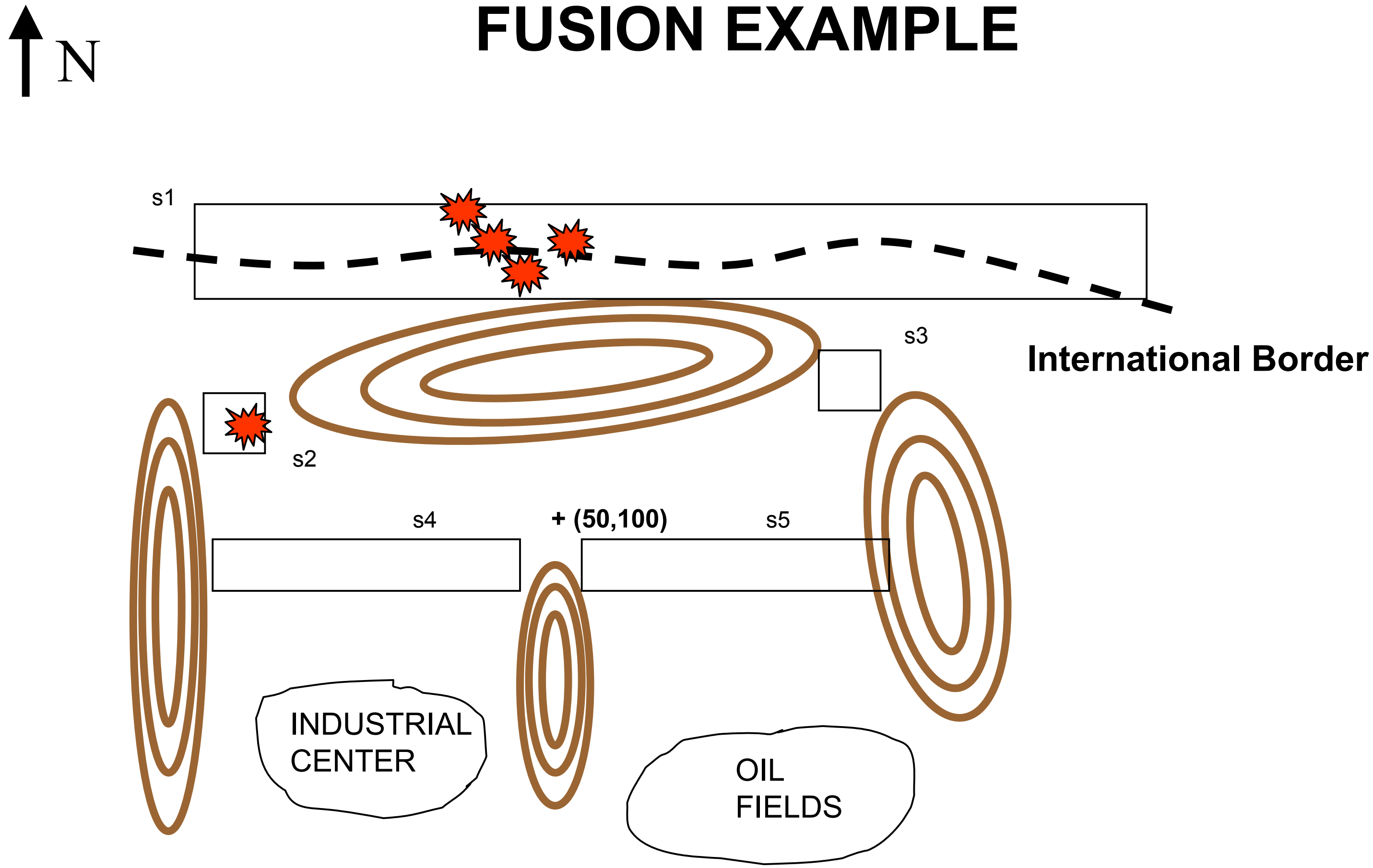
KD4 $\Rightarrow$ (p(NS(OS1)) = 0.2) $\cap$ (p(NS(OS2)) = 1.0) $\cap$ (p(NS(OS3)) = 0.4)

KD5 $\Rightarrow$ (p(NS(OS1)) = 0.2) $\cap$ (p(NS(OS2)) = 0.4) $\cap$ (p(NS(OS3)) = 1.0)

KD6 $\Rightarrow$ (p(NS(OS1)) = 0.2) $\cap$ (p(NS(OS2)) = 0.3) $\cap$ (p(NS(OS3)) = 1.0)

KD7 $\Rightarrow$ (p(NS(OS1)) = 0.2) $\cap$ (p(NS(OS2)) = 0.3) $\cap$ (p(NS(OS3)) = 1.0)

FUSION EXAMPLE



Scenario

- Current state established as OS1 (provoke)
- Enemy tanks observed at S1 and S2 (i.e. S1 > 0, S2 > 0)
- KD_1, KD_2 = T, KD_i = F for all other KD

FUSION EXAMPLE

PR = PR(OS1, OS1) = 0.4
PR(OS1, OS2) = 0.3
PR(OS1, OS3) = 0.3

V(OS1)

KD1 ⇒ (p(NS(OS1)) = 0.8) ∩
(p(NS(OS2)) = 0.1) ∩
(p(NS(OS3)) = 0.1)

KD2 ⇒ (p(NS(OS1)) = 0.4) ∩
(p(NS(OS2)) = 0.6) ∩
(p(NS(OS3)) = 0.6)

V(OS2)

KD1 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 1.0) ∩
(p(NS(OS3)) = 0.4)

KD2 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 1.0) ∩
(p(NS(OS3)) = 0.4)

V(OS3)

KD1 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 0.4) ∩
(p(NS(OS3)) = 1.0)

KD2 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 0.4) ∩
(p(NS(OS3)) = 1.0)

Now we have a problem.

The presence of multiple key descriptors, results in two values for p(NS(OS_i)) in each world.

Solutions:

(1) Ensure every value in V(OS_i) is mutually exclusive.

(2) Use some sort of combining scheme for conjunction and disjunction (CF, Fuzzy Sets)

FUSION EXAMPLE

PR = PR(OS1, OS1) = 0.4
PR(OS1, OS2) = 0.3
PR(OS1, OS3) = 0.3

V(OS1)

KD1 ⇒ (p(NS(OS1)) = 0.8) ∩
(p(NS(OS2)) = 0.1) ∩
(p(NS(OS3)) = 0.1)

KD2 ⇒ (p(NS(OS1)) = 0.4) ∩
(p(NS(OS2)) = 0.6) ∩
(p(NS(OS3)) = 0.6)

V(OS2)

KD1 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 1.0) ∩
(p(NS(OS3)) = 0.4)

KD2 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 1.0) ∩
(p(NS(OS3)) = 0.4)

V(OS3)

KD1 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 0.4) ∩
(p(NS(OS3)) = 1.0)

KD2 ⇒ (p(NS(OS1)) = 0.2) ∩
(p(NS(OS2)) = 0.4) ∩
(p(NS(OS3)) = 1.0)

We use McAllister's rule - $CF_a + CF_b * (1- CF_a)$ - to handle multiple evidence values:

Combined:

(p(NS(OS1)) = 0.88) ∩
(p(NS(OS2)) = 0.64) ∩
(p(NS(OS3)) = 0.64)

Combined:

(p(NS(OS1)) = 0.36) ∩
(p(NS(OS2)) = 1.00) ∩
(p(NS(OS3)) = 0.64)

Combined:

(p(NS(OS1)) = 0.36) ∩
(p(NS(OS2)) = 0.64) ∩
(p(NS(OS3)) = 1.00)

Fusion Step:

PR(NS(OS1)) =0.4 * 0.88 + 0.3 * 0.36 + 0.3 * 0.36 = **0.568**
PR(NS(OS2)) =0.4 * 0.64 + 0.3 * 1 + 0.3 * 0.64 = **0.748**
PR(NS(OS3)) =0.4 * 0.64 + 0.3 * 0.64 + 0.3 * 1 = **0.748**

↑ PR(OS1,OS1) * $p_{OS1}(NS(OS_i))$

↑ PR(OS1,OS2) * $p_{OS1}(NS(OS_i))$

↑ PR(OS1,OS3) * p_{OS1}(NS(O

INFORMATION FUSION
USING
BAYESIAN BELIEF NETWORKS

BAYESIAN BELIEF NETWORK REVIEW

Let : H = a hypothesis (i.e. the enemy's next state is OS1)
e = an event, or evidence (i.e. KD2 is True)

$$P(H \mid e) = \frac{P(e \mid H)\, P(H)}{P(e)}$$

(Inversion formula from Bayesian Statistics)

$$P(H \mid e) = \alpha\, P(e \mid H) \;*\; P(H)$$

where $\alpha = 1/P(e)$

posterior prob — *conditional prob* — *prior prob*

Extended to multiple hypotheses, and multiple sources of evidence (assuming independence among evidence):

$$P(H_i \mid e_1, e_2, \ldots, e_n) = \alpha\, P(H_i) \prod_{k=1}^{n} P(e_k \mid H_i)$$

<u>*Example Values*</u>

(0.4, 0.3, 0.3)
Prior prob. of OS1, OS2, OS3

KD1 \ H_i	OS1	OS2	OS3
T	0.9	0.7	0.7
F	0.1	0.3	0.3

Conditional prob of KD given OS

BAYESIAN BELIEF NETWORK REVIEW

Notation Review (Judea Pearl)

$\lambda_e(H)$

$$P(H_i \mid e_1, e_2, \ldots, e_n) = \alpha \, P(H_i) \prod_{k=1}^{n} P(e_k \mid H_i)$$

BEL(H)

$\pi(H)$

$\lambda_e(H) = \lambda(e) * M_{e|H}$

$\Lambda_e(H)$
(entire product)

conditional prob matrix

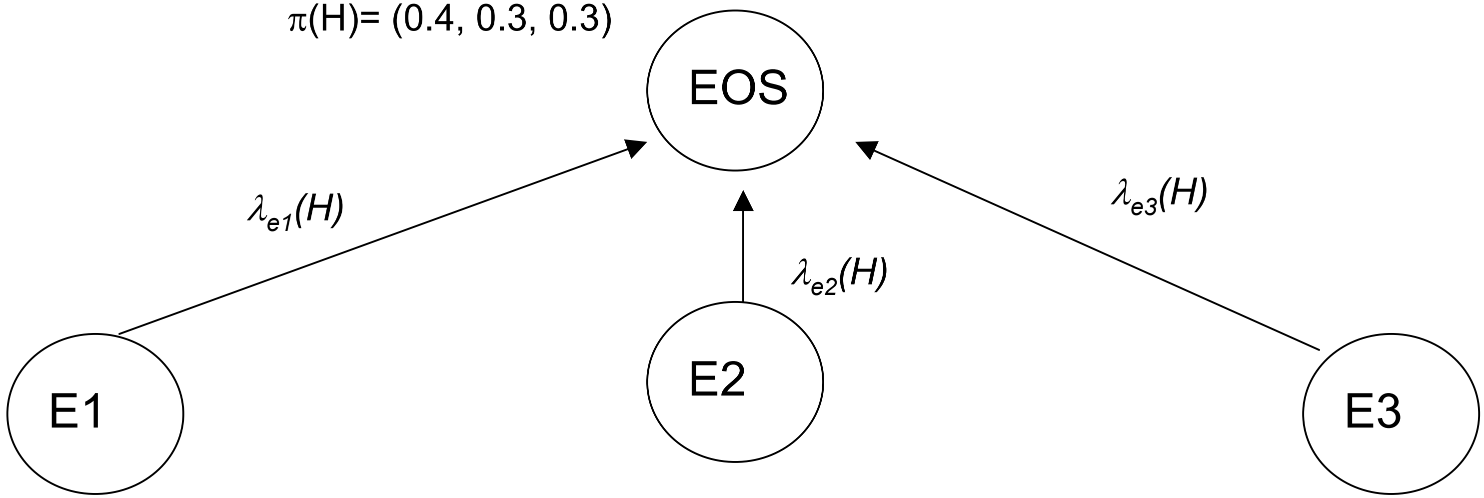
		H_i	
KD1	**OS1**	**OS2**	**OS3**
T	0.9	0.7	0.7
F	0.1	0.3	0.3

* *$\lambda(e)$*

$\lambda(e)$ = vector of probabilities associated with these values of e

BBN REVIEW

H = {H1, H2, H3 }
π(H)= (0.4, 0.3, 0.3)



E1	H1	H_i H2	H3
T	0.9	0.7	0.7
F	0.1	0.3	0.3

$M_{e1|H}$

E1	H1	H_i H2	H3
T	0.9	0.7	0.7
F	0.1	0.3	0.3

$M_{e2|H}$

E1	H1	H_i H2	H3
T	0.9	0.7	0.7
F	0.1	0.3	0.3

$M_{e3|H}$

$\lambda(e1) = [x , y]^T$
X : (prob e1=T)
Y : (prob e1=F)

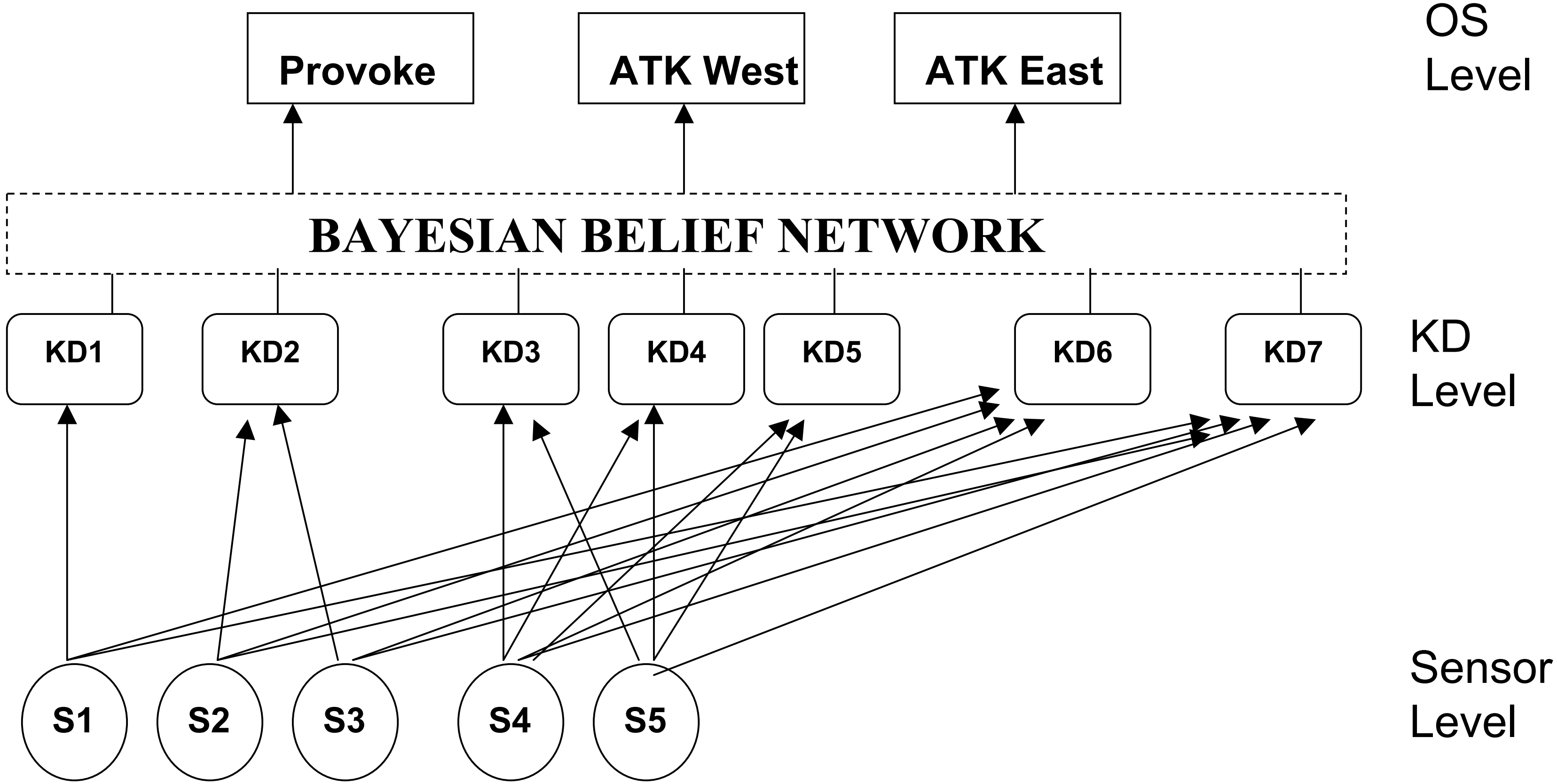
$\lambda(e1) = [x , y]^T$
X : (prob e1=T)
Y : (prob e1=F)

$\lambda(e1) = [x , y]^T$
X : (prob e1=T)
Y : (prob e1=F)

EXAMPLE INFORMATION FUSION FRAMEWORK

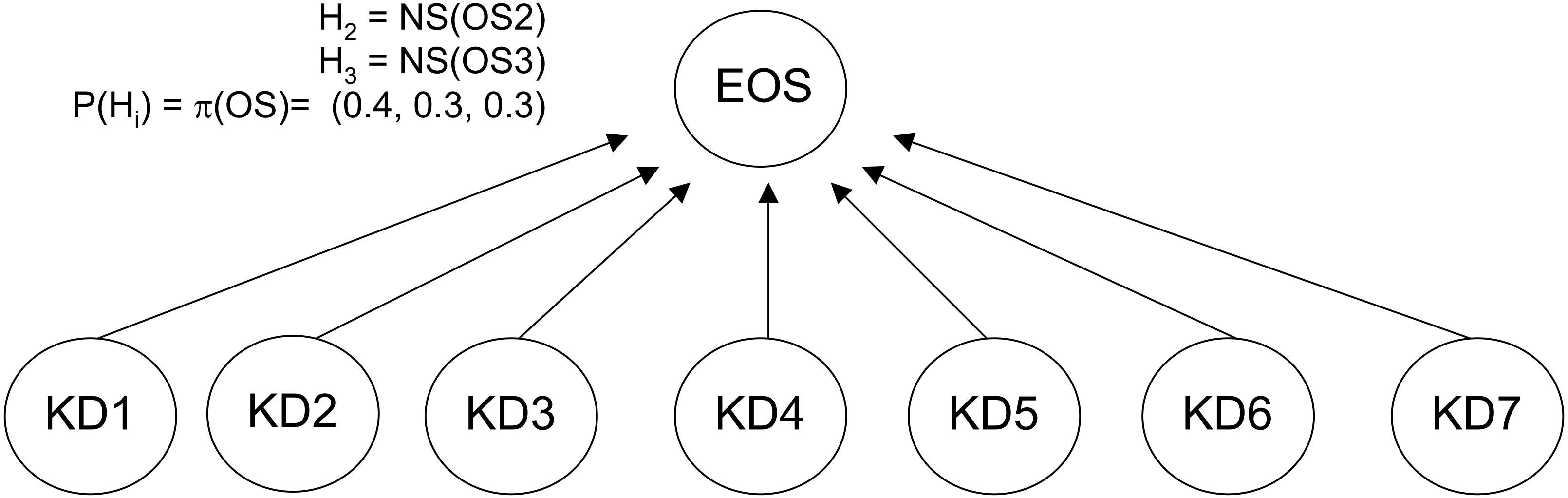
S = { S1, S2 , S3 , S4 , S5 }
V = { x where x = number of tanks read by sensor }
K = { $kd_1 = (S1 > 0)$
$kd_2 = (S2 > 0) \cup (S3 > 0)$
$kd_3 = (S4 > 0) \cup (S5 > 0)$
$kd_4 = (S4 > 0) \cap (S5 = 0)$
$kd_5 = (S4 = 0) \cap (S5 > 0)$
$kd_6 = centerX(s1, s2, \ldots, s5) > 50 \cap centerY(s1, s2, \ldots, s5) > 100$
$kd_7 = centerX(s1, s2, \ldots, s5) <= 50 \cap centerY(s1, s2, \ldots, s5) > 100$ }

O = { provoke, attack_west, attack_east }   E = O



EXAMPLE BBN MODEL

$H_1 = NS(OS1)$
$H_2 = NS(OS2)$
$H_3 = NS(OS3)$
$P(H_i) = \pi(OS) = (0.4, 0.3, 0.3)$



KD1 \ H_i	OS1	OS2	OS3
T	0.9	0.7	0.7
F	0.1	0.3	0.3

KD3 \ H_i	OS1	OS2	OS3
T	0.1	0.8	0.8
F	0.9	0.2	0.2

KD5 \ H_i	OS1	OS2	OS3
T	0.1	0.2	0.8
F	0.9	0.8	0.2

KD7 \ H_i	OS1	OS2	OS3
T	0.1	0.8	0.2
F	0.9	0.2	0.8

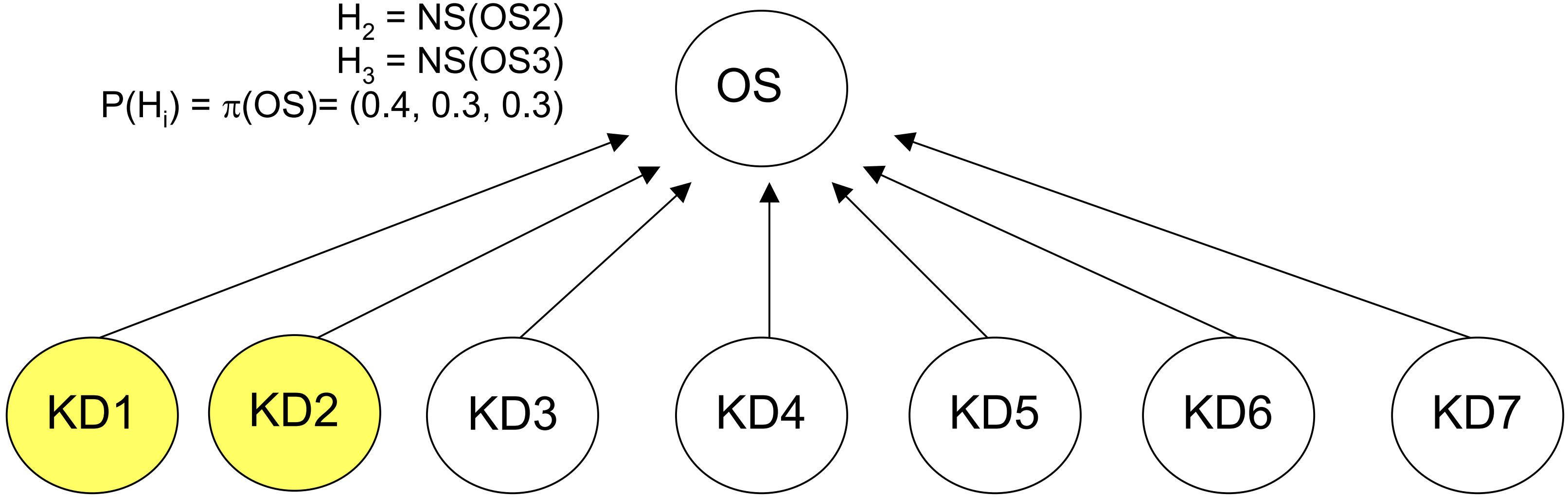
KD2 \ H_i	OS1	OS2	OS3
T	0.3	0.8	0.8
F	0.7	0.2	0.2

KD4 \ H_i	OS1	OS2	OS3
T	0.1	0.8	0.2
F	0.9	0.2	0.8

KD6 \ H_i	OS1	OS2	OS3
T	0.1	0.2	0.8
F	0.9	0.8	0.2

EXAMPLE BBN MODEL

H_1 = NS(OS1)
H_2 = NS(OS2)
H_3 = NS(OS3)
$P(H_i) = \pi(OS) = (0.4, 0.3, 0.3)$



KD1 \ H_i	OS1	OS2	OS3
T	0.9	0.7	0.7
F	0.1	0.3	0.3

KD2 \ H_i	OS1	OS2	OS3
T	0.3	0.8	0.8
F	0.7	0.2	0.2

KD3 \ H_i	OS1	OS2	OS3
T	0.1	0.8	0.8
F	0.9	0.2	0.2

KD4 \ H_i	OS1	OS2	OS3
T	0.1	0.8	0.2
F	0.9	0.2	0.8

KD5 \ H_i	OS1	OS2	OS3
T	0.1	0.2	0.8
F	0.9	0.8	0.2

KD6 \ H_i	OS1	OS2	OS3
T	0.1	0.2	0.8
F	0.9	0.8	0.2

KD7 \ H_i	OS1	OS2	OS3
T	0.1	0.8	0.2
F	0.9	0.2	0.8

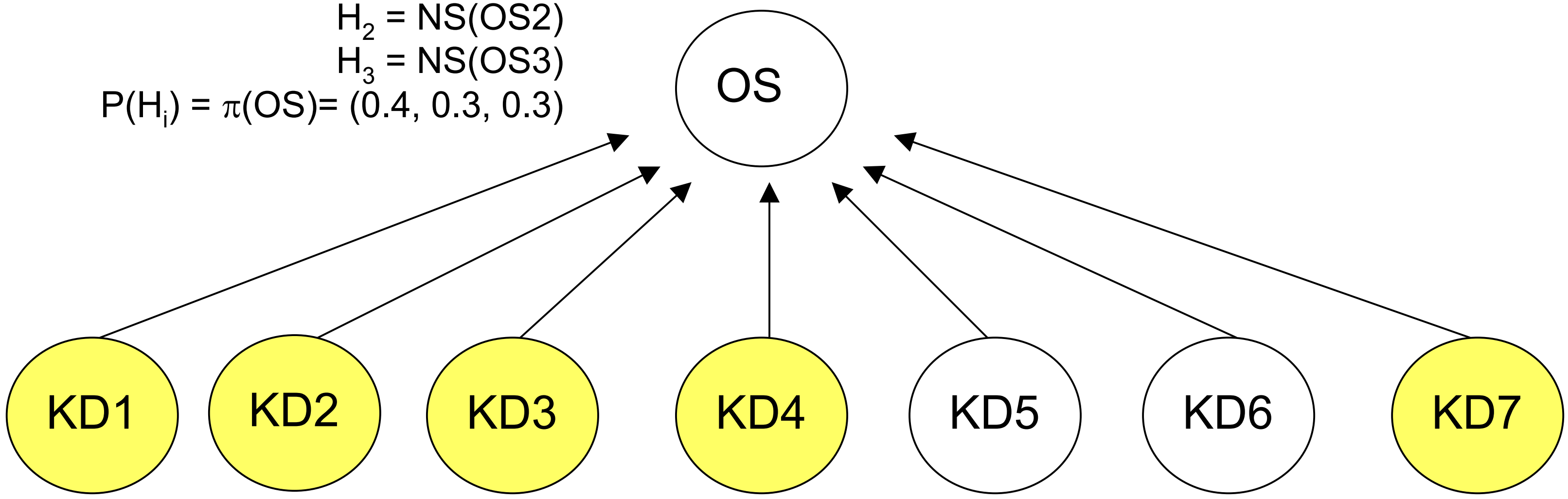
$\Lambda_{kd}(OS) = (0.9, 0.7, 0.7)(0.3, 0.8, 0.8)(0.9, 0.2, 0.2)(0.9, 0.2, 0.8)\ (0.9, 0.8, 0.2)\ (0.9, 0.8, 0.2)\ (0.9, 0.2, 0.8)$

$\Lambda_{kd}(OS) = (0.1594, 0.0029, 0.0029)$

$\pi(OS) = \alpha\ (0.1594, 0.0029, 0.0029)\ (0.4, 0.3, 0.3) = \alpha\ (0.0637, 0.0009, 0.0009) \approx (0.9737, 0.0131, 0.0131)$

EXAMPLE BBN MODEL

H_1 = NS(OS1)
H_2 = NS(OS2)
H_3 = NS(OS3)
$P(H_i) = \pi(OS) = (0.4, 0.3, 0.3)$



KD1 \ H_i	OS1	OS2	OS3
T	0.9	0.7	0.7
F	0.1	0.3	0.3

KD3 \ H_i	OS1	OS2	OS3
T	0.1	0.8	0.8
F	0.9	0.2	0.2

KD5 \ H_i	OS1	OS2	OS3
T	0.1	0.2	0.8
F	0.9	0.8	0.2

KD7 \ H_i	OS1	OS2	OS3
T	0.1	0.8	0.2
F	0.9	0.2	0.8

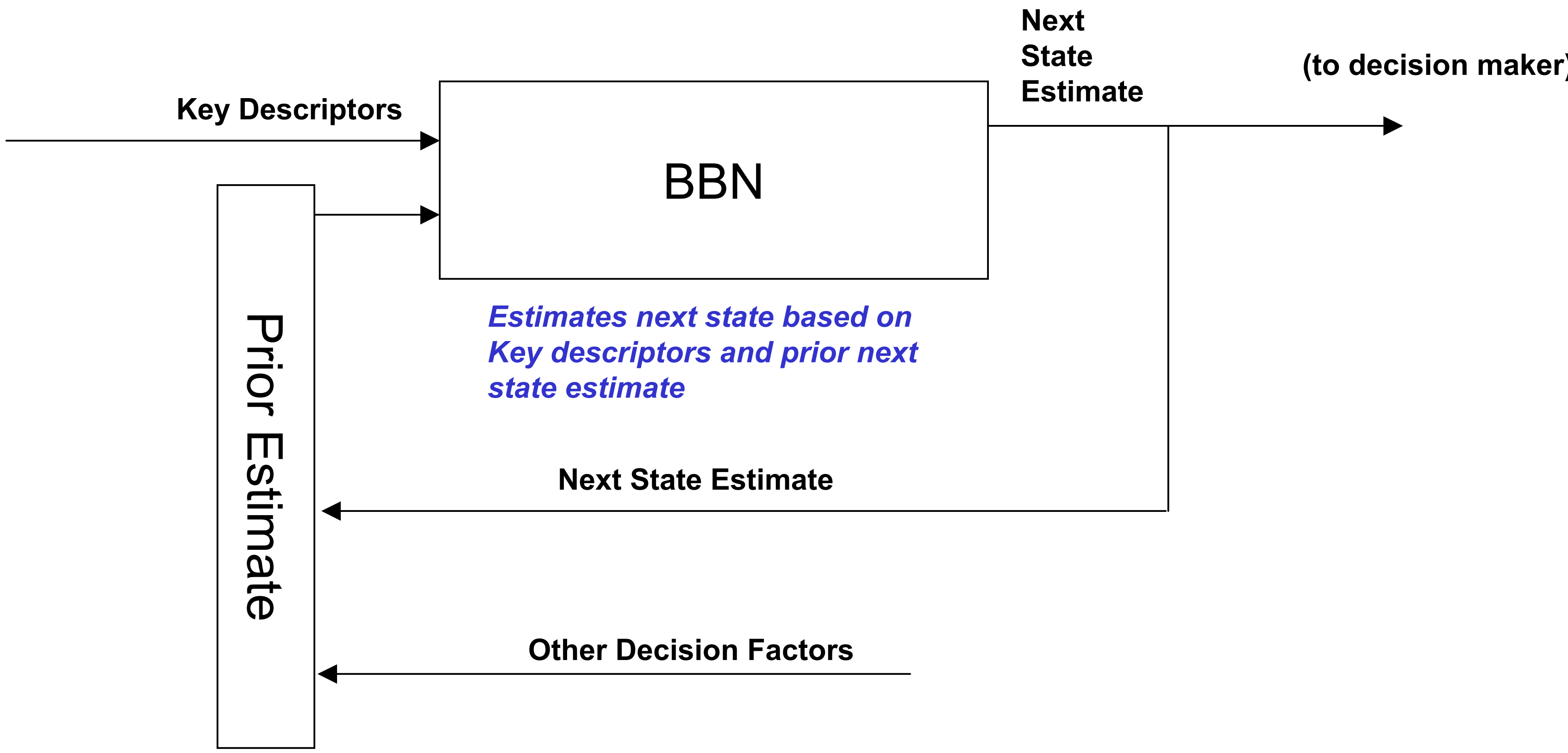
KD2 \ H_i	OS1	OS2	OS3
T	0.3	0.8	0.8
F	0.7	0.2	0.2

KD4 \ H_i	OS1	OS2	OS3
T	0.1	0.8	0.2
F	0.9	0.2	0.8

KD6 \ H_i	OS1	OS2	OS3
T	0.1	0.2	0.8
F	0.9	0.8	0.2

$\Lambda_{kd}(OS) = (0.9, 0.7, 0.7)(0.3, 0.8, 0.8)(0.1, 0.8, 0.8)(0.1, 0.8, 0.2)\ (0.9, 0.8, 0.2)\ (0.9, 0.8, 0.2)\ (0.1, 0.8, 0.2)$
$\Lambda_{kd}(OS) = (0.0002, 0.1835, 0.0007)$
$\pi(OS) = \alpha\ (0.0002, 0.1835, 0.0007)\ (0.4, 0.3, 0.3) = \alpha\ (0.00008, 0.05505, 0.00021) \approx (0.001, 0.994, 0.003)$

BBN INFORMATION FUSION PROCESS



BAYESEAN BELIEF NETWORKS

AND

FCS INFORMATION FUSION

SIMULATION DEMONSTRATION

JavaBayes

fcs_sim

FUTURE RESEARCH OBJECTIVES

Short Term Objectives

Implement PML fusion coordinator in simulation

Implement Fuzzy Logic coordinator in simulation

Design and run simulation experiments

- Compare performance of 3 fusion techniques (PML, BBN, Fuzzy)
- Validate fusion framework
- Explore sensitivities of meta model

Long Term Objectives

Use Meta-Model to calculate optimal set of KDs

Use Meta-Model and Fusion techniques to reason about friendly behavior

SUMMARY

- Automating sensor-shooter link and need to discern enemy activity
- Information Fusion Meta-Model
- Modal Logic Applied to Meta-Model
- Bayesian Belief Nets Applied to Meta-Model
- Future Research Objectives

QUESTIONS ?

CONCLUSION

"Statistical thinking will one day be as necessary a qualification for efficient citizenship as the ability to read and write."

--H.G. Wells