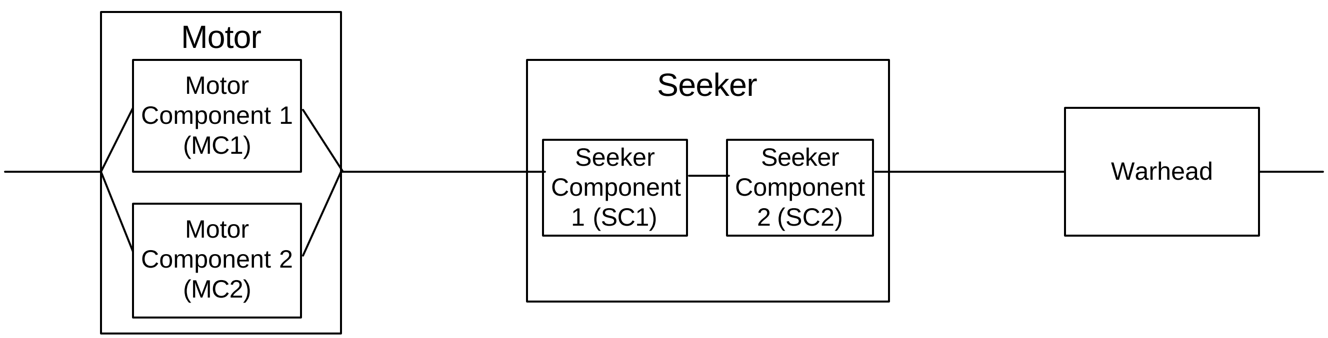
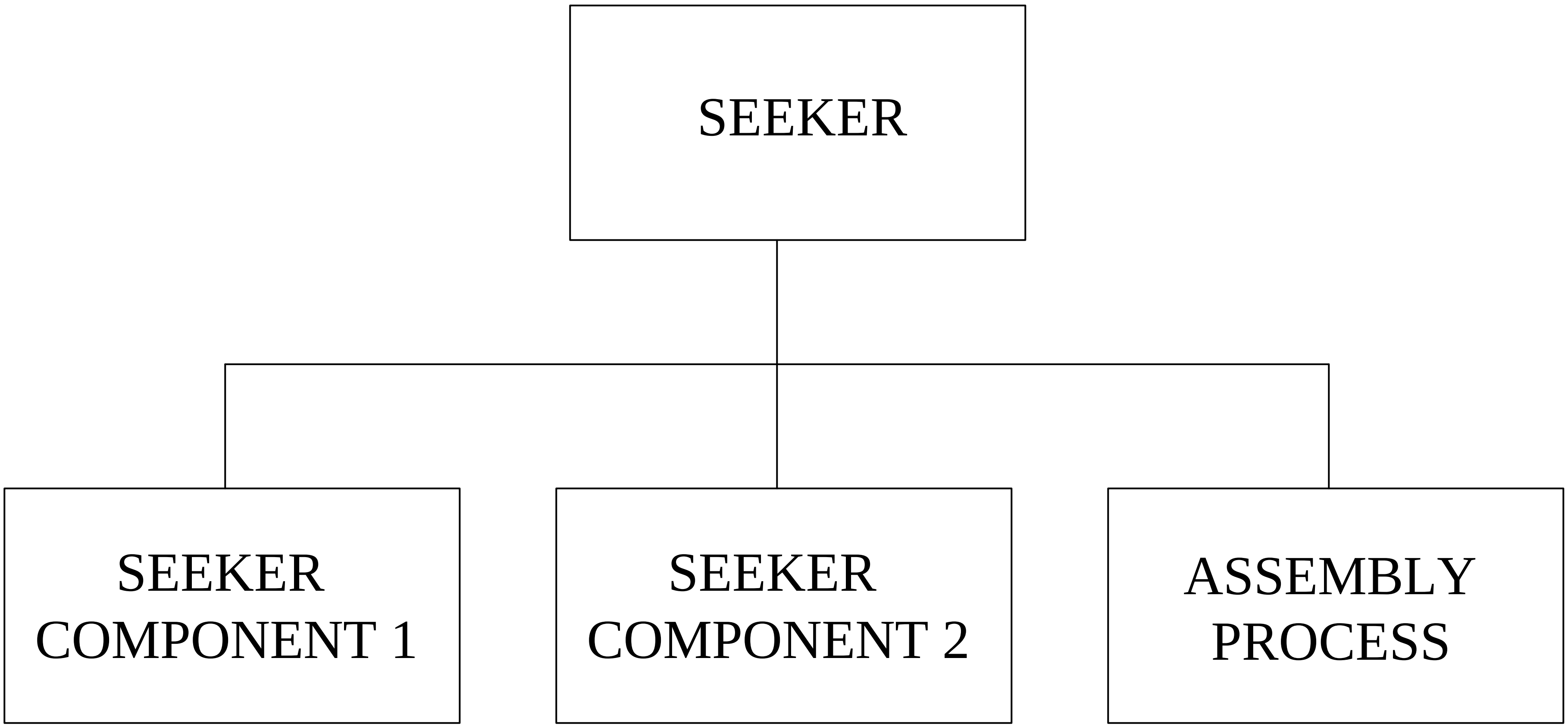
PROPAGATING UNCERTAINTY THROUGH THE SYSTEM

REALLY DEADLY MISSILE SYSTEM (RDMS)



RDMS EXAMPLE



PROPAGATING UP

Many of the models presented during the course for integrating information “within a box” are applicable “across the boxes”: for example, the reliability models and the hierarchical and non-hierarchical information combination techniques.

However, for “rolling up” information, one of the most common techniques we use is Monte Carlo simulation.

PROPAGATING UP

SEEKER

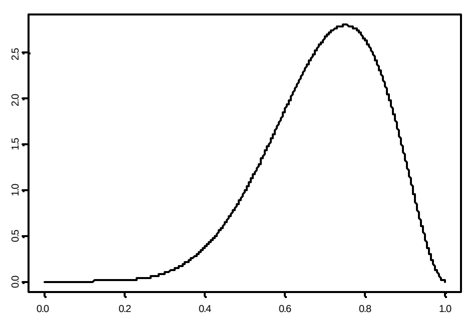
$p_S = p_1 \, p_2 \, p_A$

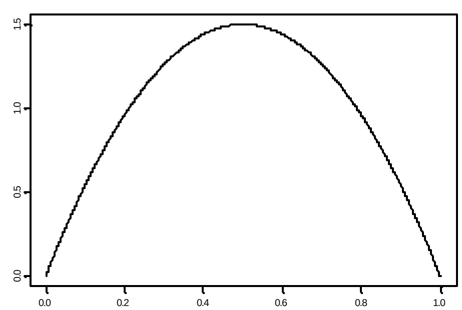
SEEKER COMPONENT 1

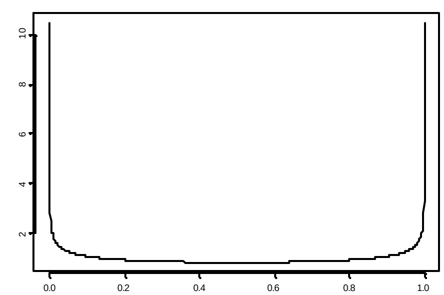
SEEKER COMPONENT 2

ASSEMBLY PROCESS

$p_1 \sim \text{Beta}(7,3)$ $\qquad p_2 \sim \text{Beta}(2,2)$ $\qquad p_A \sim \text{Beta}(0.75,0.75)$







PROPAGATING UP

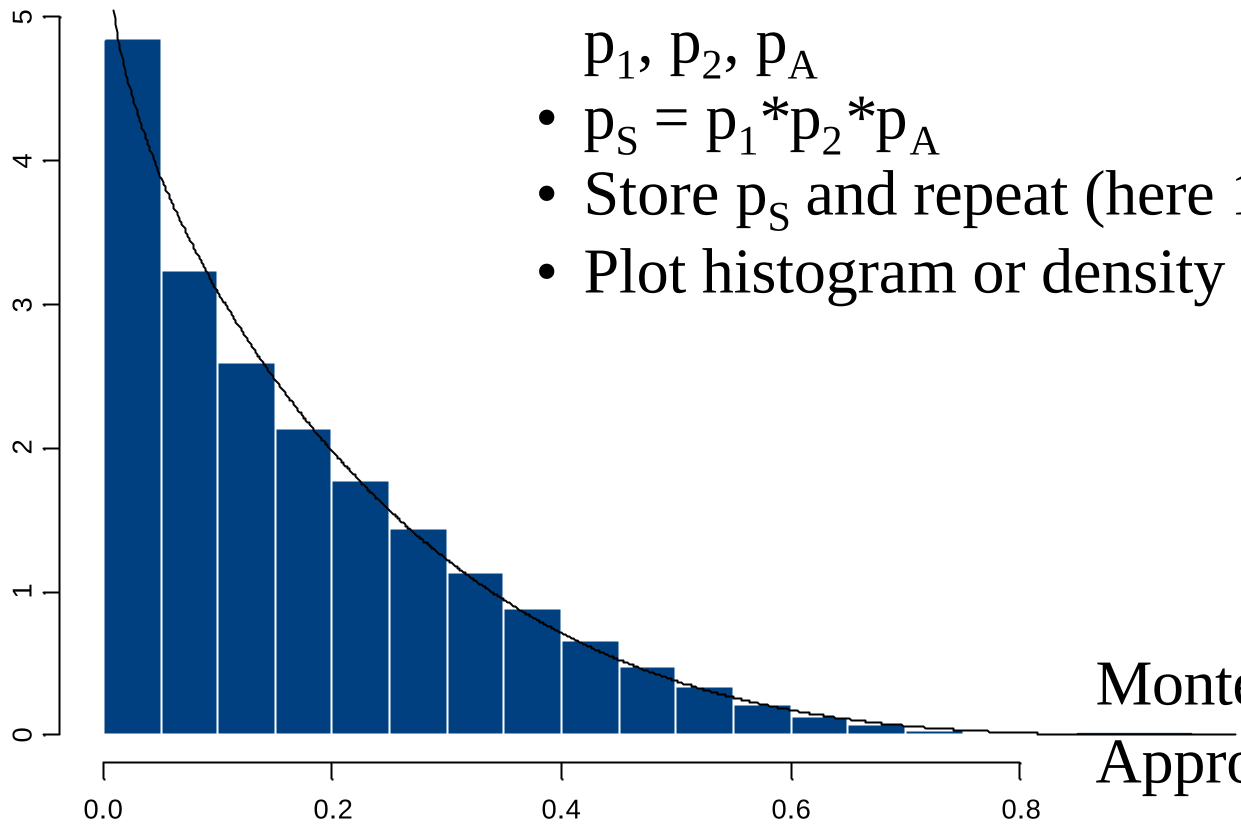
How do we calculate the distribution of p_S?

- Analytically (hard for this problem)
- Martz and Waller (1990) approximation, which gives a Beta(0.932,4.39)
- Monte Carlo

PROPAGATING UP

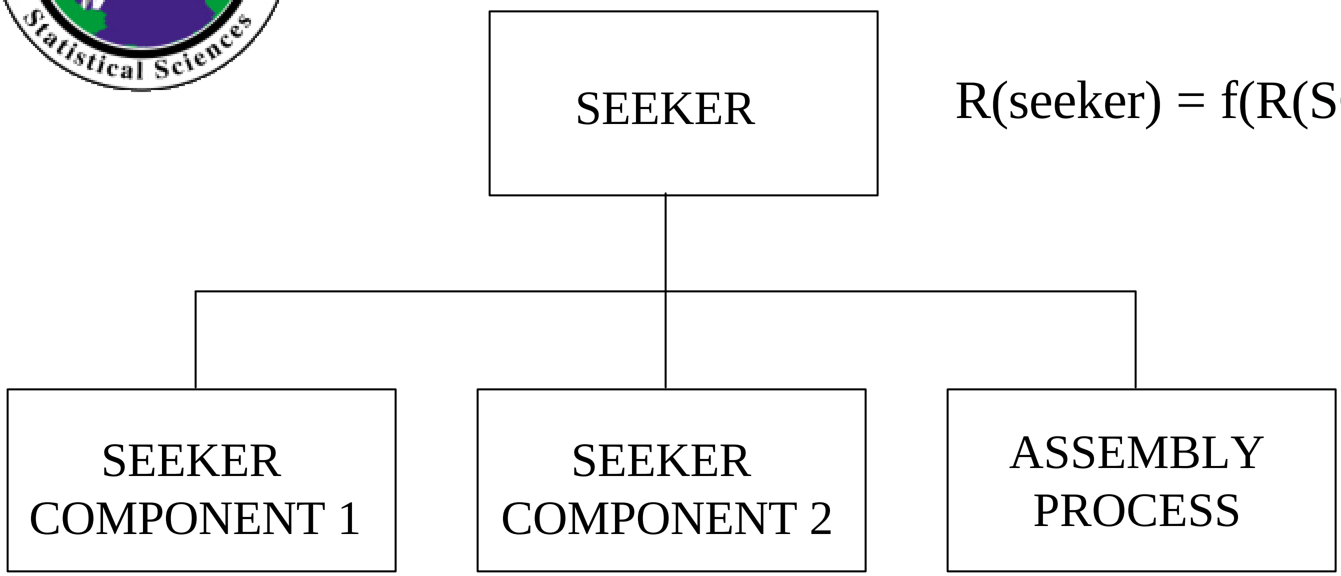
How does the Monte Carlo work?

- Take one sample from the distributions for p_1, p_2, p_A
- $p_S = p_1 * p_2 * p_A$
- Store p_S and repeat (here 10,000 times)
- Plot histogram or density of p_S, or use sample



Monte Carlo histogram; Approximate Beta solid

PROPAGATING DOWN



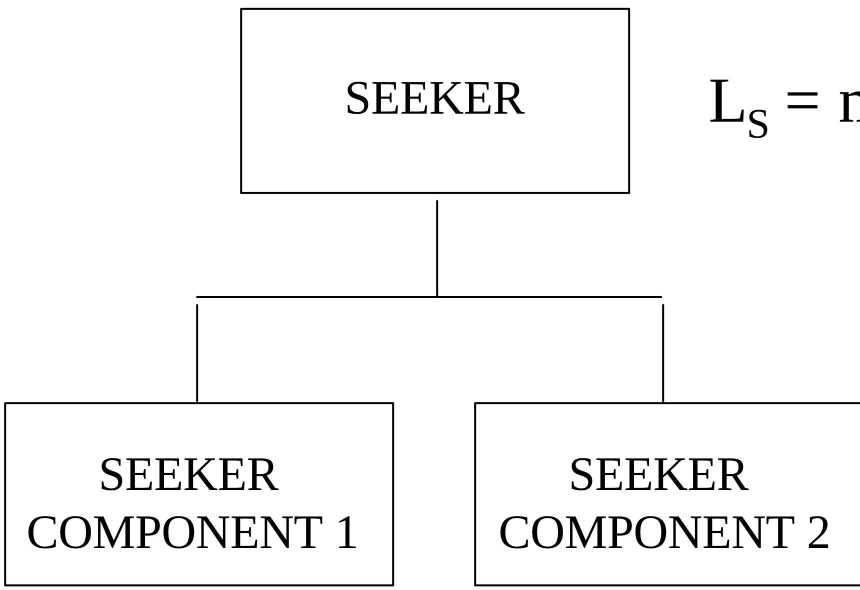
R(seeker) = f(R(SC1),R(SC2),R(AP))

Suppose that we have information about the seeker, and we want to understand what kind of information that gives us about its components. At many stages of the analysis, we may want to answer questions about subsystems (e.g., should I test, should I redesign, what if I . . . .)

We want the distributions throughout the representation to stay internally consistent.

PROPAGATING DOWN

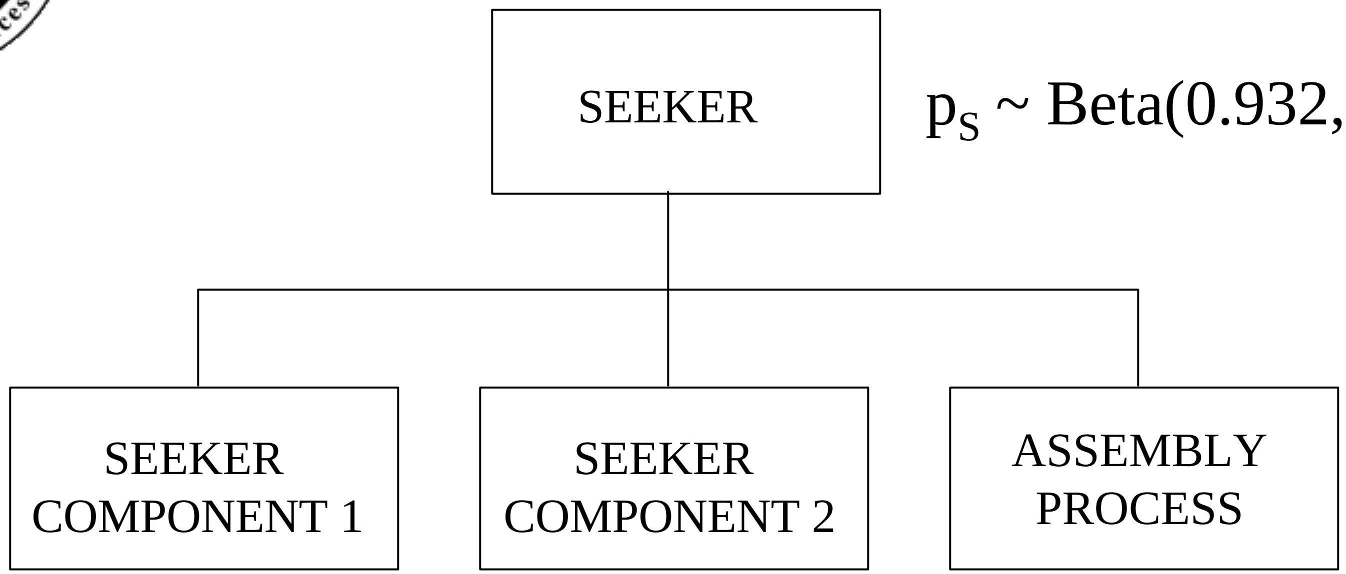
It is not immediately obvious how to do this,
nor is the solution necessarily unique.



$$L_S = \min(L_1, L_2) \sim \mathrm{Exp}(t_1 + t_2)$$

$L_1 \sim \mathrm{Exp}(t_1)$ $\quad$ $L_2 \sim \mathrm{Exp}(t_2)$

PROPAGATING DOWN



$p_S \sim \text{Beta}(0.932,4.39)$

$p_1 \sim \text{Beta}(7,3)$ $p_2 \sim \text{Beta}(2,2)$ $p_A \sim \text{Beta}(0.75,0.75)$

Suppose we get new test data on the seeker: we run 12 tests and 10 are successful. We update our distribution on p_S to Beta(10.932,6.39). How do we update the distributions of p_1, p_2, and p_A to be consistent with this information?

PROPAGATING DOWN

There may not be a unique choice, but there is a good choice that works in many situations.

- We start with a prior distribution for (p_1, p_2, p_A) that induces a prior distribution on p_S.
- We get new information that leads to a new posterior distribution on p_S.
- We want the new posterior distribution for (p_1, p_2, p_A).

- Using Bayes Theorem:

$$\pi(p_1, p_2, p_A|D) \propto f(\text{data}|p_1, p_2, p_A)\ \pi(p_1, p_2, p_A)$$

- We don't know $f(\text{data}|p_1, p_2, p_A)$
- But we do know: $\pi(p_S|\text{data}) \propto f(\text{data}|p_S)\ \pi(p_S)$

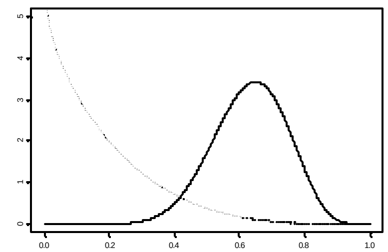
So, we substitute $f(\text{data}|p_1, p_2, p_A) = f(\text{data}|p_S)$.

This works when $\pi(p_1, p_2, p_A| p_S,\text{data}) = \pi(p_1, p_2, p_A| p_S)$.

PROPAGATING DOWN

Priors: dotted
Posteriors: solid

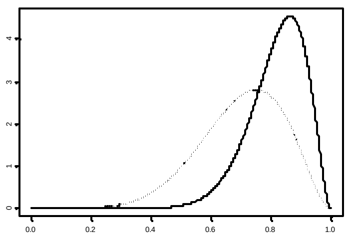
SEEKER

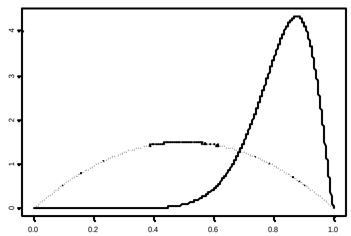


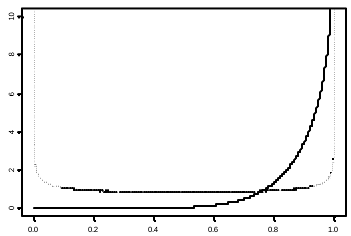
SEEKER COMPONENT 1

SEEKER COMPONENT 2

ASSEMBLY PROCESS







And there is correlation between the subcomponents and subprocesses.

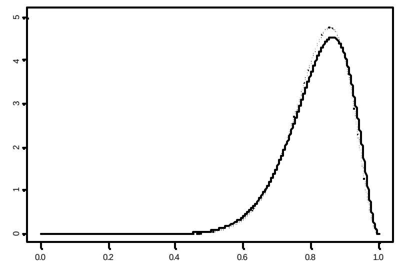
PROPAGATE LATERALLY

What if we ran four more tests on Seeker Component 2?

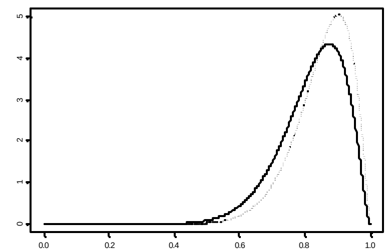
- Information comes in at the component level, but since information has been propagated down from the top level, the distributions of the components and assembly process are *dependent*.
- Consequently, the first step is to propagate the information laterally using Bayes rule.

Suppose we had four successes in four trials.

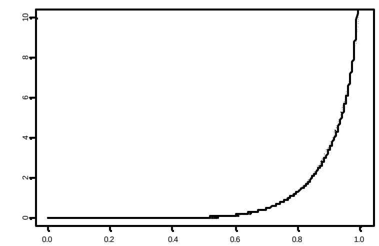
SEEKER COMPONENT 1



SEEKER COMPONENT 2



ASSEMBLY PROCESS



PROPAGATE UP

Do we gain enough information about the system that it is worthwhile to do the new tests?