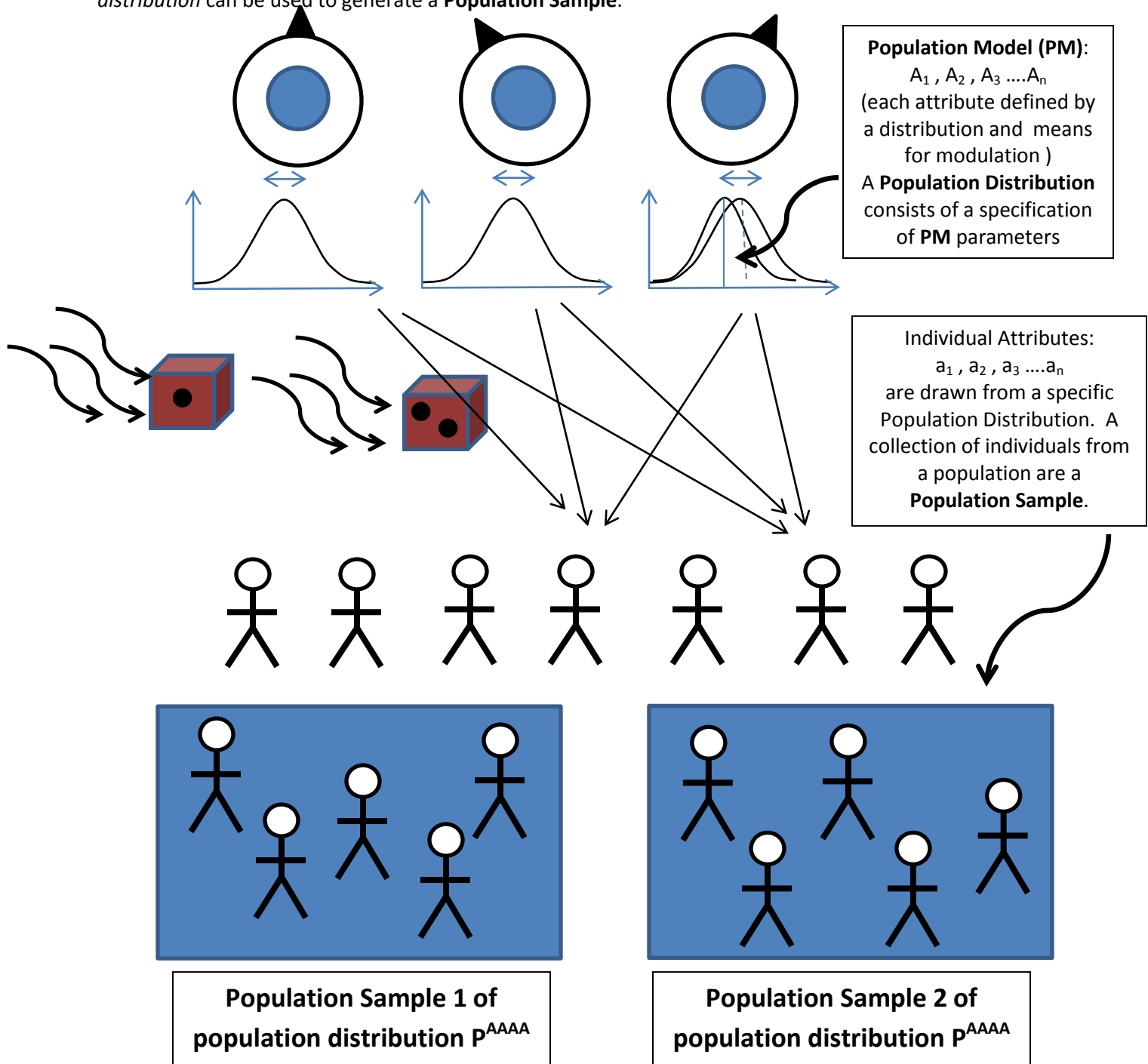**THE MODEL**

A **Model** is composed of three separate elements: a **Population Model**, an **Agent Model**, and a **Trajectory Model.**

The **Population Model** will consist of a list of those agent attributes deemed necessary for the Agent Model. For each item on the list, it will specify a distribution and a means for modulating that distribution…for example, we can imagine an agent attribute called "courage" that would be specified in the model by Courage, and determined by a Gaussian that is modulated via changes in the mean and standard deviation. A **Population Distribution** is a single example of the *population model*, and can be specified by the distributions and means for modulating each Attribute. So, for example, a population distribution could be specified as Aggressiveness (μ = 0, σ=0.5); Courage (μ = 0.25, σ=0.75), and so on. Changes in the modulation settings thus specify distinct **Populations Distribution**. A *population distribution* can be used to generate a **Population Sample**.

Population Model (PM):
A_1 , A_2 , A_3 ....A_n
(each attribute defined by a distribution and means for modulation)
A **Population Distribution** consists of a specification of **PM** parameters

Individual Attributes:
a_1 , a_2 , a_3 ....a_n
are drawn from a specific Population Distribution. A collection of individuals from a population are a **Population Sample**.

Population Sample 1 of population distribution P^{AAAA}

Population Sample 2 of population distribution P^{AAAA}

A **Trajectory Model** is a series of measures used to represent the dynamic properties of a single simulation run or live Order Book feed. Its components will be specified but an example is the Hurst Exponent. A full Trajectory model will specify each measure and its implementation.

Agent Model: this is a multi-stage decision making algorithm followed by all agents. At each stage it makes stochastic decisions that are influenced by the current state of the order book and by the **Agent Attributes** of the individual agent. Please note that the order book statistics and agent attributes given as examples below are for demonstration/intuition and are *not* currently part of the model.

The **Agent Model** will have a series of decision making steps followed by each agent.

1) **Action Prompt:** this will be an agent function that determines the agent's activity periodicity. Upon receiving the prompt, the agent sends for inputs from the order book (for example: *volume*) and from the agent's individual attributes (for example: *patience*). The function produces a probability (0→1) that the agent will decide to *act*. If the stochastic decision produces a decision to *not act*, then the agent schedules the next *action prompt* and becomes dormant. The decision to act considers whether the agent has one or more open orders in the order book. If there is one or more open orders in the order book, the *cancel prompt* is triggered. If the agent does not have an open order, then the *make order prompt* is triggered.

2) **Cancel Prompt:** this will be an agent function that determines whether to cancel an existing order. Upon receiving this prompt, the agent will send for inputs from the order book (for example: *volatility*) and from the agent's individual attributes (for example: *risk tolerance*). The function will produce a probability (0→1) that oldest unconsidered order placed by the agent will be canceled. If the decision is to cancel the order, then the order is canceled and *cancel prompt* is invoked again immediately. If the decision is to *not cancel*, then order is marked as "considered" and the agent invokes the *cancel prompt*.

3) **Order Prompt:** this will be an agent function that determines whether to place an order in the order book. Upon receiving this prompt, the agent will send for inputs from the order book (for example: *volume*) and from the agent's individual attributes (for example: *capacity*). The function produces a probability (0→1) that the agent will decide to *make an order* against which a stochastic decision will be made. The decision to make an order will trigger the **Bid Ask Prompt**. This routine is scheduled immediately. If the result is *not order*, then the agent schedules the next *action prompt* and becomes dormant.

4) **Bid Ask Prompt:** this will be an agent function that determines whether the act is to be a *bid* or an *ask*. Upon receiving this prompt, the agent will send for inputs from the order book (for example: *slope of price*) and from the agent's individual attributes (for example: *optimism*). The function produces a probability (0→1) that the agent will decide to *bid* against which a stochastic decision will be made. The decision to *bid* will trigger the **Bid Price Prompt**. This routine is scheduled immediately. If the result is *not bid*, then the order will be an *ask* and will trigger the **Ask Price Prompt**. This routine is scheduled immediately.

5) **Bid Price Prompt:** this will be an agent function that determines the price of a *bid order*. Upon receiving this prompt, the agent will send for inputs from the order book (for example: *current best ask* or *current price*) and from the agent's individual attributes (for example: *ambition*).

Individual attributes and order book data will be used to modify the shape of an ***ideal bid price distribution***. A stochastic throw against the skewed distribution will determine the **Bid Price**. Following the setting of the *bid price*, the **Bid Size Prompt** is scheduled immediately.

6) **Bid Size Prompt:** this will be an agent function that determines the size of a *bid order*. Upon receiving this prompt, the agent will send for inputs from the order book (for example: *average bid size*) and from the agent's individual attributes (for example: *courage*). Individual attributes and order book data will be used to skew an ***ideal bid size distribution***. A stochastic throw against the skewed distribution will determine the **Bid Size**. Following the setting of the *bid size*, the **order (bid price; size)** is entered into the order book. The agent schedules the next *action prompt* and goes dormant.

7) **Ask Price Prompt:** this will be an agent function that determines the price of a *ask order*. Upon receiving this prompt, the agent will send for inputs from the order book (for example: *current best bid* or *current price*) and from the agent's individual attributes (for example: *ambition*). Individual attributes and order book data will be used to skew an ***ideal ask price distribution***. A stochastic throw against the skewed distribution will determine the **Ask Price**. Following the setting of the *ask price*, the **Ask Size Prompt** is scheduled immediately.

8) **Ask Size Prompt:** this will be an agent function that determines the size of an *ask order*. Upon receiving this prompt, the agent will send for inputs from the order book (for example: *average ask size*) and from the agent's individual attributes (for example: *courage*). Individual attributes and order book data will be used to skew an ***ideal ask size distribution***. A stochastic throw against the skewed distribution will determine the **Ask Size**. Following the setting of the *ask size*, the **order (ask price; size)** is entered into the order book. The agent schedules the next *action prompt* and goes dormant.

The elements of the agent model are thus:

- Individual agent attributes (a_1, a_2,…a_i)
- Order book statistics (s_1, s_2,…s_n)
- An Ideal Ask Price Distribution; Ideal Ask Size Distribution; Ideal Bid Price Distribution; Ideal Bid Size Distribution.
- The decision making function for each of the 8 prompts.

THE MODEL

Action ?

yes

no

Schedule next...

This loop is repeated until all open orders have been considered...

Is there an unconsidered open order?

no

yes

no

Make Order ?

no

yes

no

Cancel ?

yes

Bid / Ask ?

bid

ask

yes

Cancel Order !

Set Bid Price

Set Ask Price

Bid Price

Set Bid Size

Ask Price

Set Ask Size

Bid Size

Note: black arrows are competed immediately. The red arrows indicate a return to dormant. Dashed lines indicate outputs rather than actions.

Make Order (direction, price, size) !

Ask Size