ASHA Places Model for Populations of Unit-Individuals,

whether conventional or not, in Places or elsewhere, but always well-mixed

The ASHA model has been developed to explore potential standard ways for managing dynamic populations in Evolvix that combine maximal simplicity, flexibility and scalability. ASHAs randomly assign Places to the unit-sized individuals of a population, but without tracking where those places are, as long as counted correctly. The core idea is that of an

Aggregated State Homogeneity Approximator, ASHA vs
Simulated Explicit Heterogeneity Approximator, SEHA

where SEHAs detail what ASHAs abstract away. Populations at Places `With` have

- a **hard limit** (`Aces`, as tracked with the help of Placer `Dice` and Places `Lack`);
- a **soft limit** (from balancing of Actions `Grow`, `Fade`, `Gain`, `Loss`, all in the hard limit);
- **neutralizers** to hide the ASHA at certain values if used as rate modifier (`InIt`, `OuOf`).

The spatially explicit thinking behind ASHAs was first developed for the Evolvix 'Places' model; it can make code more readable (eg. `PlWith` vs `ASHA_With`). To simplify use of the ASHA code motif, the overviews below are ready for copy-paste-adapting (see code).

Brief	_Frag_	in an ASHA	Usual Example ASHA Name	Explicit ASHA_Places Feature Name
Aces	_Aces_	ASHA_Aces_	ASHA_Aces_MyExampleASHA	ASHA Placed Aces Maximal Count
Dice	_Dice_	ASHA_Dice_	ASHA_Dice_MyExampleASHA	ASHA Placer Dice Probability
With	_With_	ASHA_With_	ASHA_With_MyExampleASHA	ASHA Places With Item Counted
Lack	_Lack_	ASHA_Lack_	ASHA_Lack_MyExampleASHA	ASHA Places Lacking Item Counted
InIt	_InIt_	ASHA_InIt_	ASHA_InIt_MyExampleASHA	ASHA In It Invisible With Scaling
OuOf	_OuOf_	ASHA_OuOf_	ASHA_OuOf_MyExampleASHA	ASHA OutOf Invisible Lack Scaling
Gain	_Gain_	ASHA_Gain_	ASHA_Gain_MyExampleASHA	ASHA Placer Gain for any Lacking
Loss	_Loss_	ASHA_Loss_	ASHA_Loss_MyExampleASHA	ASHA Placer Loss for losing With
Grow	_Grow_	ASHA_Grow_	ASHA_Grow_MyExampleASHA	ASHA Placer Grow by Reproducing
Fade	_Fade_	ASHA_Fade_	ASHA_Fade_MyExampleASHA	ASHA Placer Fade to stop Crowding

Brief Summarizing Explanation of Feature Definition

During simulations: `FIX`**ed or** `VAR`**iable**

`Aces` X_A Count of **All Computationally Equivalent Spaces**; sum of all notional Places held in an ASHA, defining a hard limit of all its space; limit enforced by `_Aces_=_Shut_+_Open_=_Shut_+_With_+_Lack_`, always tracked.
`FIX` at Quest start (by User)

`Dice` X_D Expected frequency of randomly selecting 1 of all existing `Aces` for some an unspecified `Action` (without orienting the probability as available for `_With_` and `_Lack_`); `_Dice_` = 1/(`_Open_`), categorically excluding `_Shut_`.
`FIX` at start (Must be 1/`_Aces_`)

`With` X_W Current Count of all `_Aces_` `_With_` a unit Item of the nominal Type defined by this ASHA (Name, Context, and how `_With_` is used); works well to slow unwanted `Actions`, less so for increasing wanted rates.
`VAR` = 0 or set by User at start

`Lack` X_L Current Count of all `_Aces_` `_Lack_`ing a unit Item of the nominal Type defined by this ASHA (Name, Context, and how `_Lack_` is used); works well to slow wanted `Actions`, less so for increasing unwanted rates.
`VAR` = 0 or set (Must add up to `_Aces_`)

`InIt` X_i Neutralizing factor for `_With_` to hide the ASHA in (`_Dice_*_With_*_InIt_`) products in an `Action` Rate/Probability that is controllable by this ASHA; use `_InIt_=(_Aces_/_With_)` for ASHA-free null-models.
`FIX` = 2 if `_With_`:`_Lack_` is 50:50 ...

`OuOf` X_o Neutralizing factor for `_Lack_` to hide the ASHA in (`_Dice_*_Lack_*_OuOf_`) products in an `Action` Rate/Probability that is controllable by this ASHA; use `_OuOf_=(_Aces_/_Lack_)` for ASHA-free null-models.
`FIX` ... usually a good starting point

`Gain` X_n Import `Actions` must change 1 `_Lack_` to 1 `_With_` and scale Rates by (`_Dice_*_Lack_*_OuOf_*_Gain_`) to properly import 1 external Item into the ASHA - as 1 random `Place` `Lacking` must be found to `Place` the `Gain`.
`FIX` at start; add Gain Action

`Loss` X_s Spontaneous `Loss` or Decay of 1 Item from all Places `With` takes 1 `Action` changing 1 `_With_` to 1 `_Lack_` to properly release 1 Item from the ASHA; scale by (`_With_*_InIt_*_Loss_`); no Placer `Dice` search occurs.
`FIX` at start; add `Loss Action`

`Grow` X_r To properly `Grow` 1 new Item by Items at Places With, 1 density-dependent `Grow Action` must change 1 `_Lack_` to 1 `_With_` at a Rate scaled by (`_Dice_*_Lack_*_OuOf_*_Grow_`) as 1 random Place `Lacking` is required.
`FIX` Slo-Mo Explosion speed, `Grow` Action

`Fade` X_e As density-dependent failure, stress, ... increase in Slow-Motion Explosions, `Fade` Actions changing 1 `_With_` to 1 `_Lack_` at Rates scaled by (`_Dice_*_With_*_InIt_*_Fade_`); this ends all SloMo Explosions.
`FIX` Slo-Mo Explosion limiting `Fade` Action

QQr8p2

Careful: if controlling >1 Action by 1 ASHA, the underpinning mechanics must be crystal clear, or else confusing model behavior will be introduced by the extra constraints the ASHA places on the rates of those Actions that are then forced to always share a factor. In turn, many ASHAs for 1 Action are OK, since each ASHA can be switched off independently at will; no extra constraints exist.