

An aerial photograph of a coral reef system. The image shows a series of white sand beaches and shallow turquoise lagoons separated by darker, deeper water channels. The perspective is from a high angle, looking down at the ocean. The text 'VENTITY™' is overlaid in the center-left area.

VENTITY™

*and Entity based
system dynamics modelling*

Agenda

- Introduction
- Ventity Concepts
- Ventity Demo
- Discussion

VENTITY helps teams collaborate to build complex models with model-data separation, modularity, dynamic creation of structure, sparse matrix relationships and intelligent agents
... and it's easy to use.

The approach lets you work on real problems, with real data, at a natural level of detail

Multilevel problems are ubiquitous

- Competitive dynamics, with new firms or products introduced midway through a simulation
- Payer/provider ecosystem, with patient migration and churn
- Doctor and patient behavior, with influence propagating on social networks or spatial grids
- Project model, with task prerequisite matrix loaded from data
- Supply chain with arbitrary organization
- Climate integrated assessment with physical and economic sectors delegated to different teams

Aggregation is not a binary choice

- 
- Individual agents
 - Cohorts, arrays, chains and coflows
 - Copy-paste structure
 - Classic lumped SD

Flexible aggregation options are needed

- Some problems require detail (can't aggregate in principle)
- But data availability may limit what is practical
- Sometimes it's hard to draw a stock-flow diagram (too many dimensions)
- Sometimes detail is easier (hard to decide a priori what aggregation is appropriate)
- Exploit big data and individual event statistics
- Integrate the work of distributed teams
- Show decision makers the granularity they live with (promo campaigns, doses, territories, ...)

What do we need to make this work?

- Model-data separation, so model can be more generic
- Modularity:
 - Component reuse
 - Standalone calibration of submodels
 - Parallel development distributed among team members
- Sparse representation (not all drugs need to serve all indications)
- Structure change (addition of new brands)
- Easy data import, preprocessing and comparison with model output
- Fast, robust simulation engine
- Calibration optimization
- Attractive diagrams, output and user controls
- Calendar time instead of abstract months
- Good run management & archiving

Ventity concepts

- Entities
- Attributes
- References
- Collections
- Actions

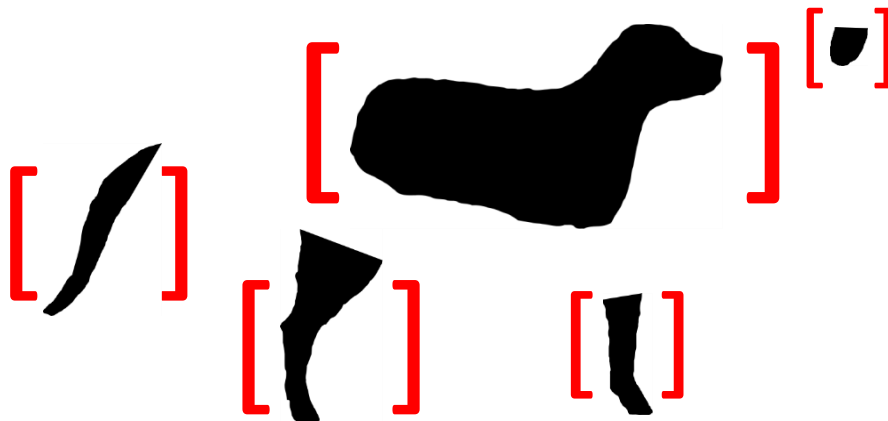
The standard approach: arrays or rows/columns/sheets

- Fast and familiar for simple replication of structure

But ...

- Makes detail a property of a location or an equation, when it is really a property of a larger component ... an **entity**

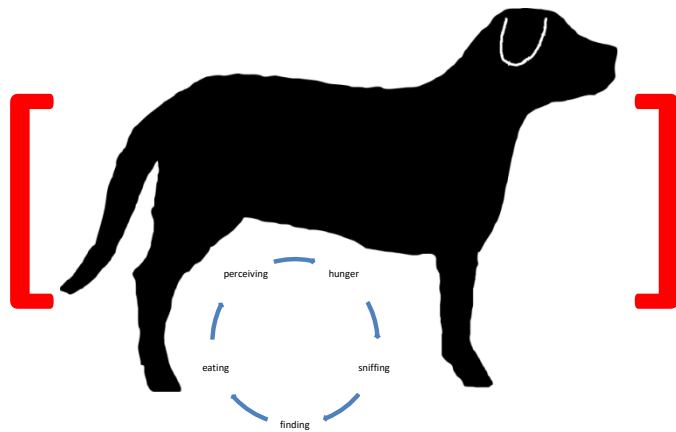
Array Equations



Entity Types

- AKA object, class, subsystem, sector, agent
- Organizes a group of equations with similar detail and function into a single container
- Defines the structure of a generic entity, to be instantiated later
- Singleton entities, like the Model itself, have just one instance

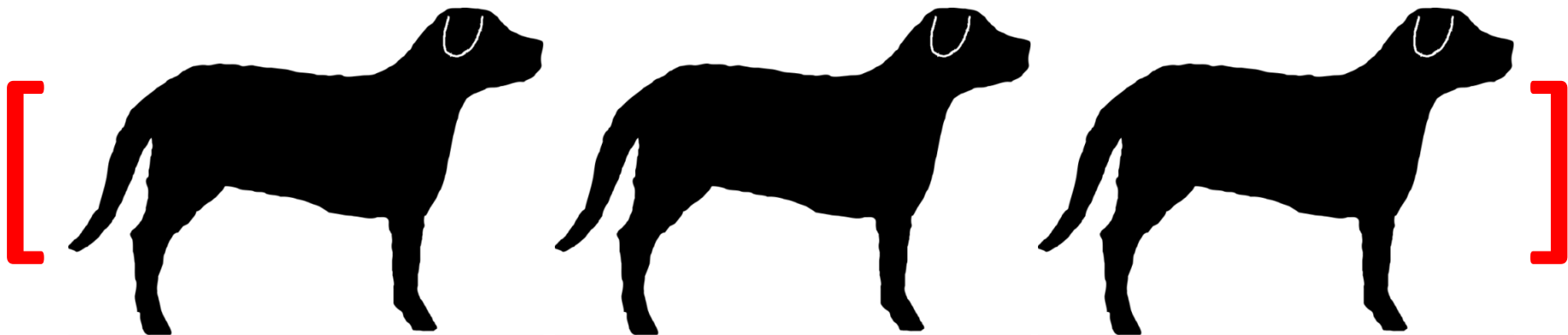
An Entity



Creating Entities

- Instantiate from data, e.g., a relational table
- Create on the fly, during a simulation, via Actions

A Collection of Entities



How do you know which entity is which?

- **Attributes** identify entities and contain ancillary properties
- Attributes are like array dimensions, but...
 - No need to define all combinations of elements, so sparse matrices are easy to handle
 - Redundancy is cheap
 - Mapping from one to many and back is easy

Attributes

Color

blue

red

green

Food

steak

steak

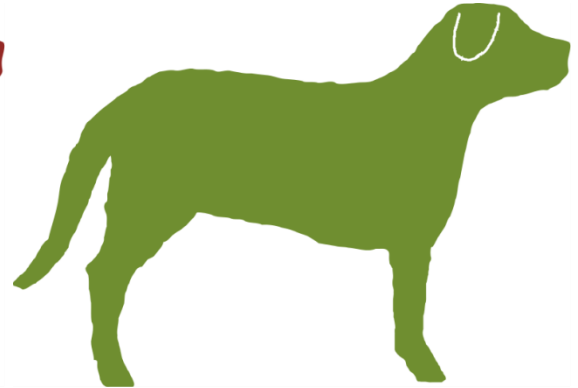
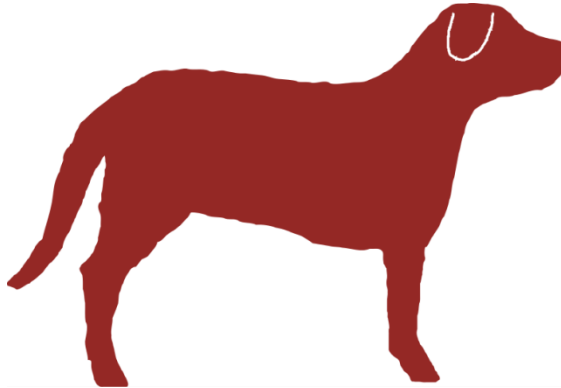
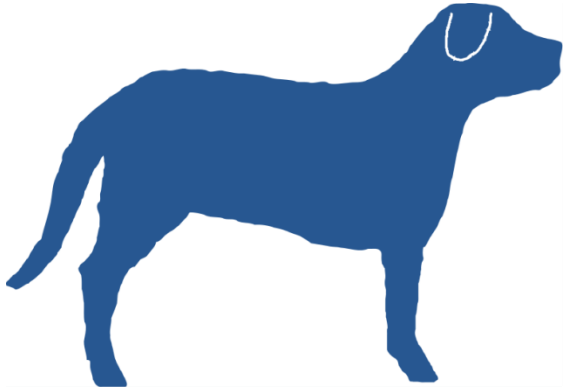
chicken

Name
(key)

Fido

Rover

Elvis

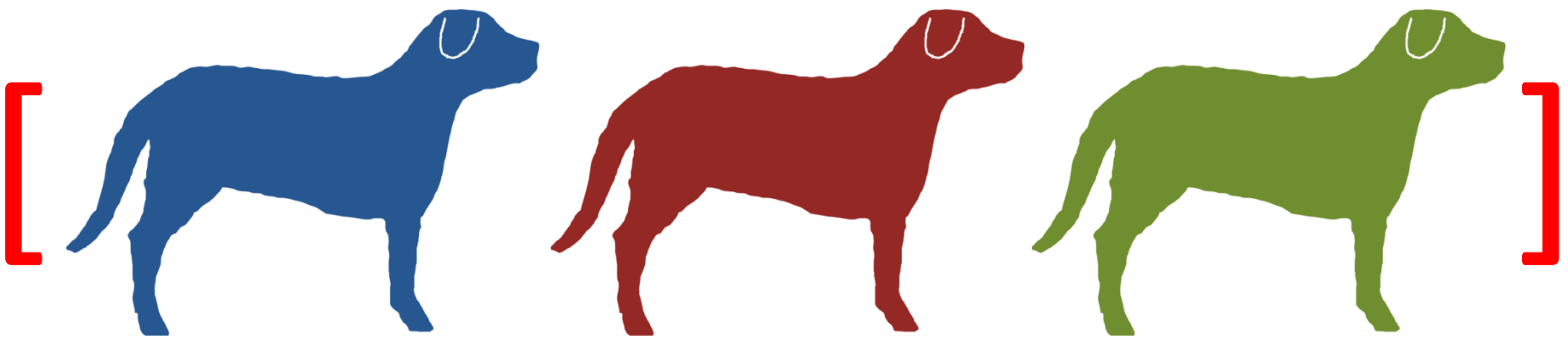


Collections

- **Collections** are lists of entities
- **Subcollections** slice and dice collections **by an attribute**, like all the restaurants in a city
- **Aggregate functions** take sums, products, etc. of variables in the members of a collection

A Subcollection by Food

Color	blue	red	green
Food	steak	steak	chicken
Name (key)	Fido	Rover	Elvis



The image shows three dog silhouettes in profile, facing right. The first is blue, the second is red, and the third is green. Above each silhouette is a table row. The blue dog is labeled 'Fido' and 'steak'. The red dog is labeled 'Rover' and 'steak'. The green dog is labeled 'Elvis' and 'chicken'. The table has three columns: 'Color', 'Food', and 'Name (key)'. The 'Food' column has two 'steak' entries and one 'chicken' entry. The 'Color' column has 'blue', 'red', and 'green'. The 'Name (key)' column has 'Fido', 'Rover', and 'Elvis'. A red bracket on the left groups the blue and red dogs, and a red bracket on the right groups the green dog.

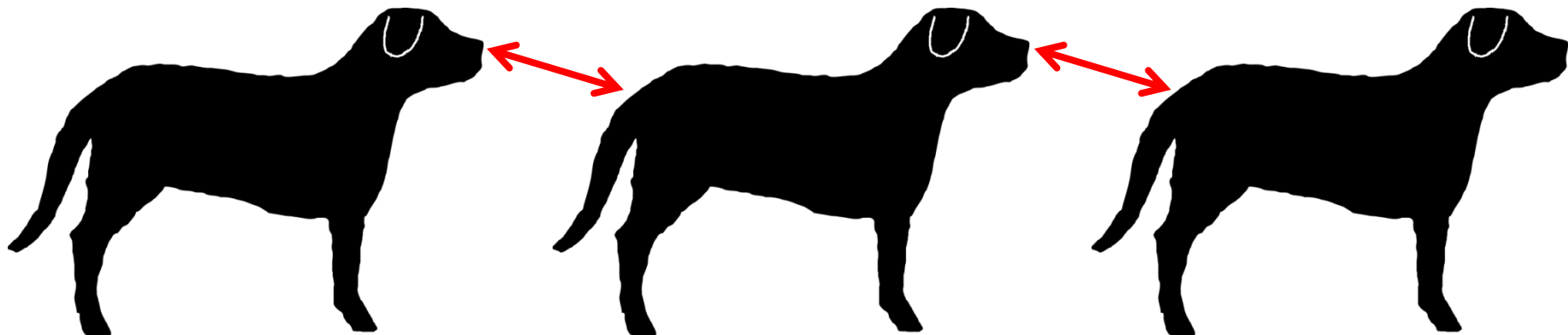
References

- References are special attributes that connect entities of the same or different types
 - Restaurants $\leftrightarrow$ Cities
 - People $\leftrightarrow$ People (a social network)
 - Project tasks $\leftrightarrow$ Prerequisites, Resources

Dog ID

Friend
Dog ID

Relationships



Triggers and Actions

- **Triggers** cause actions
- **Actions** are discrete events that:
 - Create new entities
 - Delete entities
 - Change states
 - Iterate over lists (collections)

