Module 1

Modelling in SELES

Overview

Andrew Fall

Landscape Systems Ecologist

Gowlland Technologies Ltd.

April 2024

Module 1 Objectives

What can you expect to learn from this module?

- Some basic guidelines for building complex models
- A high-level understanding of SELES as a tool to build and run landscape models
- An overview of the training material modules

➢ See SELES User Documentation: Part 1 – sections 1 and 2

Landscape Management Challenges

- ⇒ large areas and long time frames
- ⇒ complexity
  - spatio-temporal feedback between pattern and process
- ⇒ many sources of uncertainty
  - spatial information, key processes, natural variability
- ⇒ competing values: conservation, resources, tourism
  - lack of optimal solutions

Yet, landscape managers must make decisions

- land-use, sustainable resource management, endangered species recovery, natural disturbance …

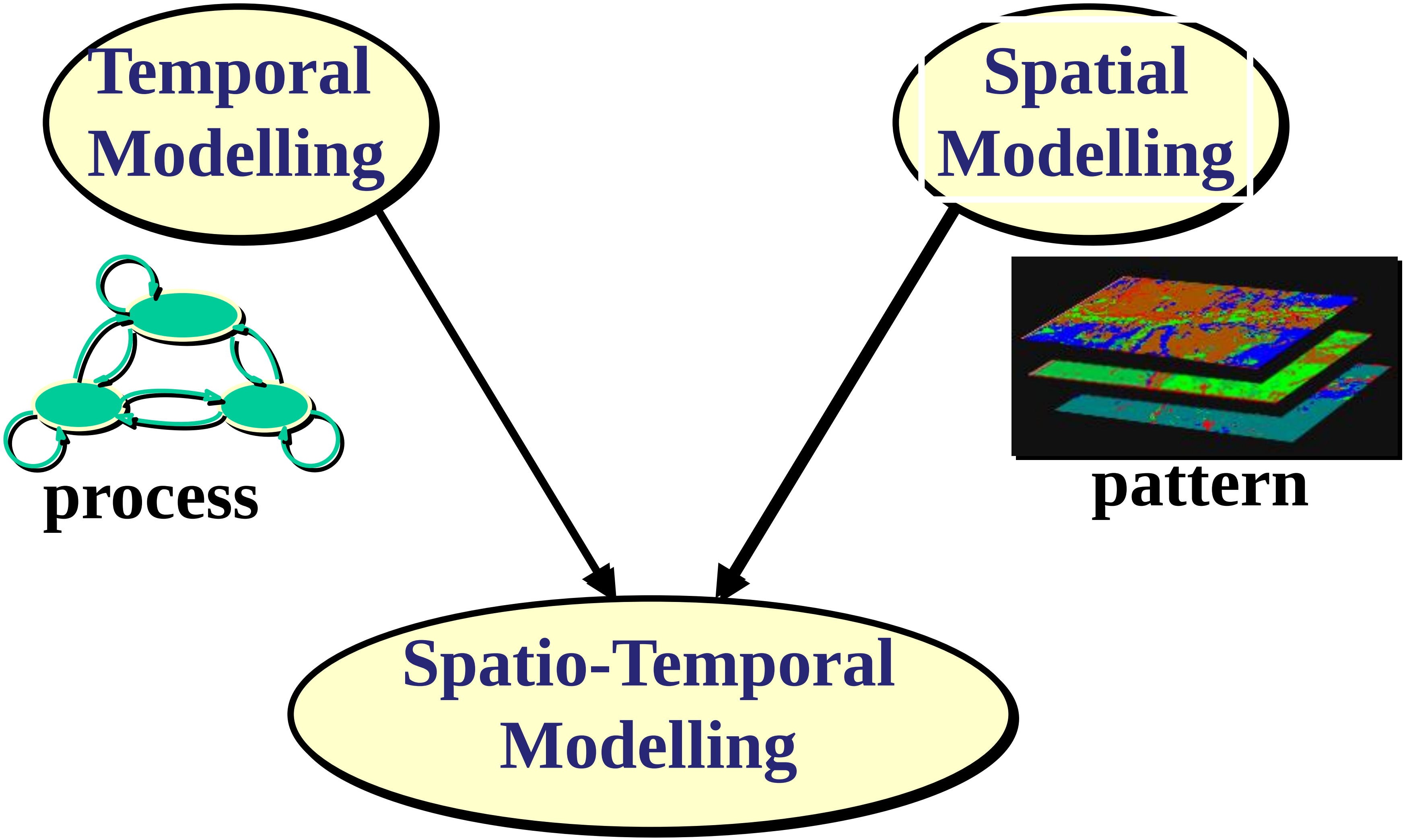
Landscapes as Complex Systems

- Landscape scale problems require a systems approach
  - system complexity limits analytical and statistical methods

- Models are important tools to explore hypotheses and support decisions in complex systems

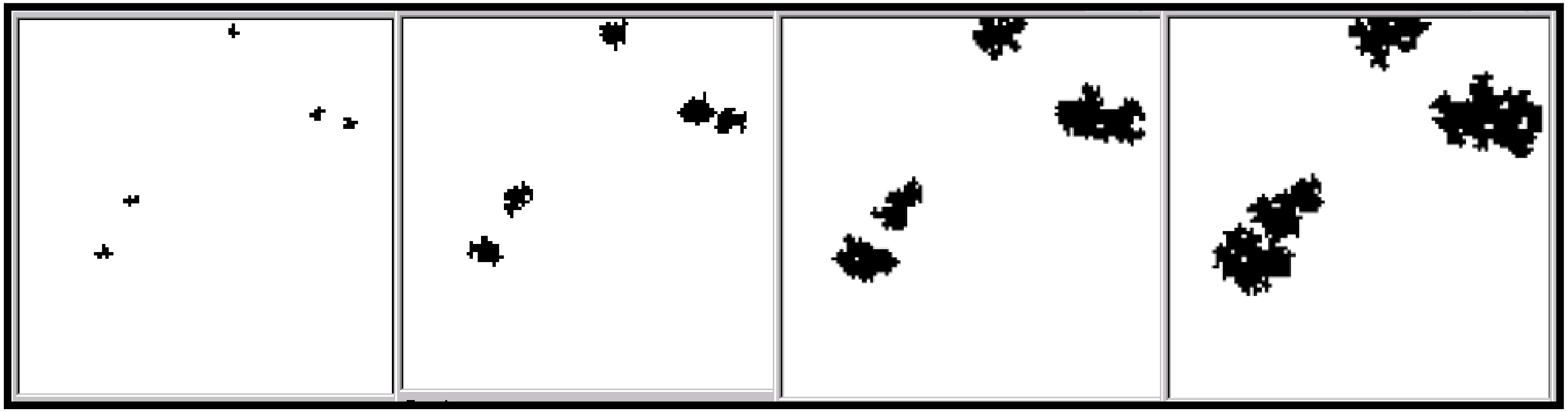
Weinberg's Classification of Systems

- *Organized simplicity*
  - strong, consistent relationships among a few components
  - analytical models are possible (calculus, optimization, LP)
- *Unorganized complexity*
  - many components, each acting fairly uniformly and independently
  - statistical models can describe system behaviour
- ***Organized complexity***
  - many components with a variety of relationships and interactions
  - usually requires systems analysis and simulation


Temporal Modelling
Spatial Modelling
process
pattern
Spatio-Temporal Modelling

Dynamic Spatial Models

- Explicitly simulate changes over time
- Dynamic feedback of state changes
- Output a sequence of models states, not a single map



Stochastic vs. Deterministic Models

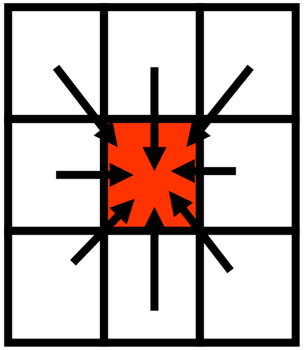
- Deterministic
  - always produce the same results given the same initial conditions

- Stochastic
  - includes a random variable or process
  - produce different output each run

Loaded dice

Cellular Automata

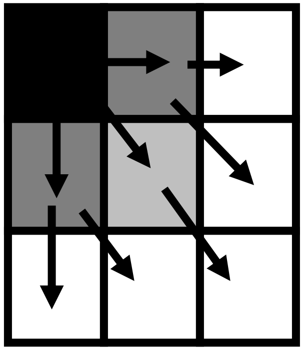
- Each cell runs as an independent sub-model
- State at next time step depends on current state plus state of neighbours



- Patterns: emergent

Diffusion Models

- Spread across space (grid-based or vector-based)
- Need to reconcile spread in Euclidean space with grid
- Example: wildfire spreading, water flow



Event-based Models

- Sequence of events
- Event: discrete process
- Discrete-event simulation

- Example: many natural disturbance models

Population Models

- Represent a population as a *set* of *non-distinct* individuals

1	0	4
0	5	2
0	0	0

- Stage based models
  - Population is divided into life stages (e.g. egg, tadpole, juvenile, sub-adult, adult)
- Matrix population models
  - Formalization of stage based models

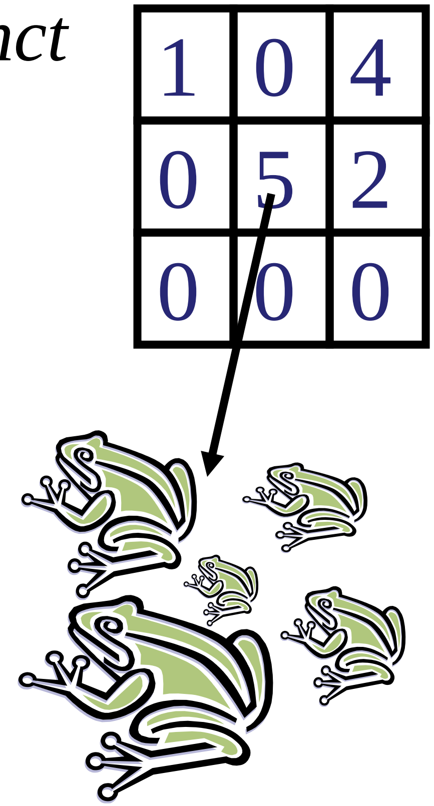
Individual-based Models

- Represent a population as a *set* of *distinct* individuals, each tracked separately
-

  Identity persistence

- Issues: movement rules, mortality, …

- Example: random walkers



Challenges and Rewards of Spatio-temporal Models

+ modelling of *lateral fluxes* (e.g. flow along roads, spatial blocks/fires)
+ interaction between pattern and process (e.g. feedback between fire and salvage logging)

- model complexity
- error propagation
- data limitations

4 Tenets
of
Landscape Modelling

Tenet #1

The modelling process must adapt to fit the decision-making process

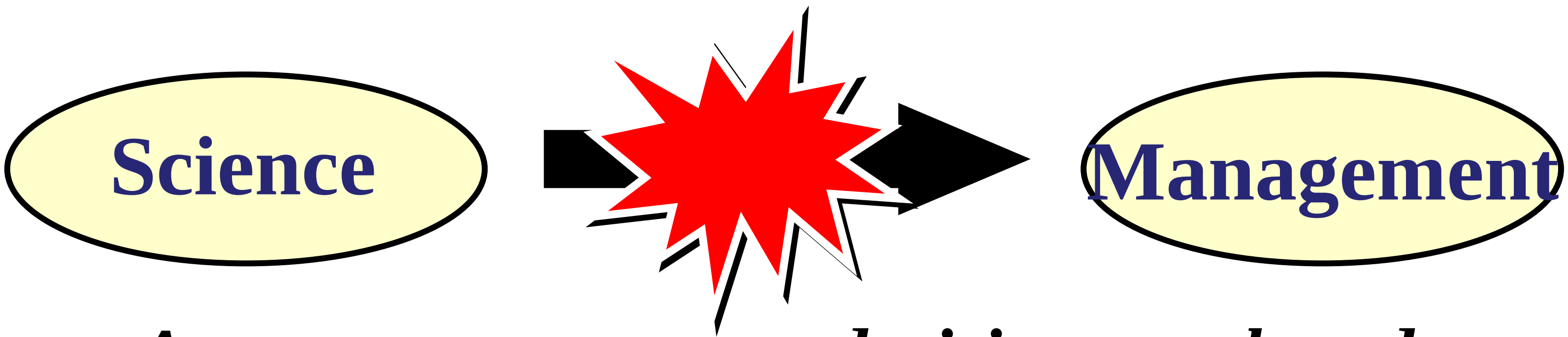
not vice versa

- to be useful, you must be able to understand *and explain* a model

Common Approaches

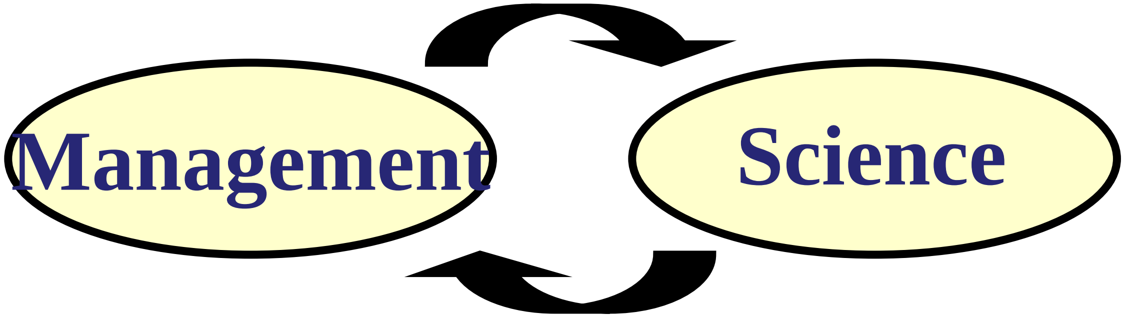
- ***develop a research model***
- ***propose application in a management context***

Common Approaches



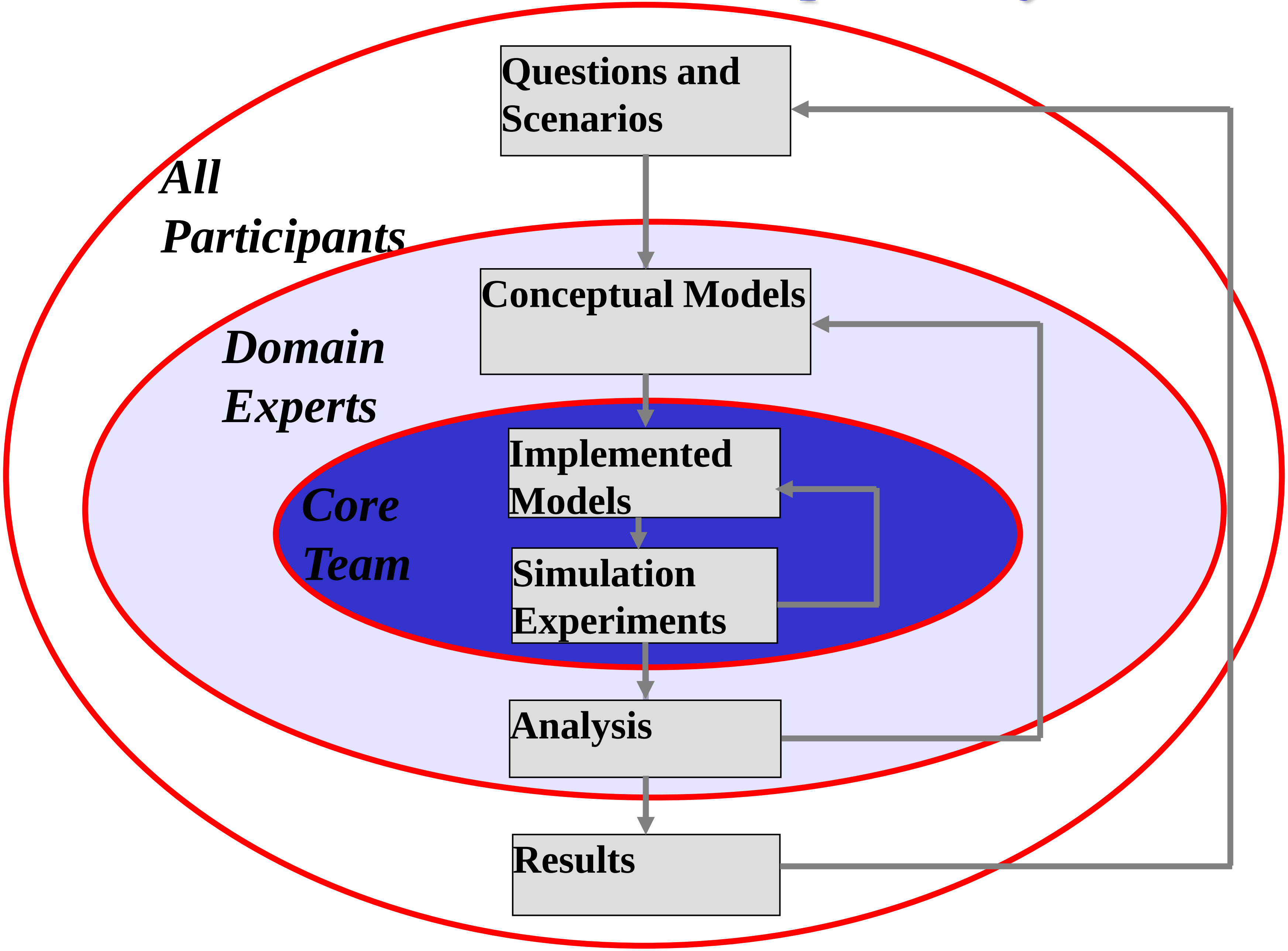
- ***Assumes management decisions are based primarily on improved knowledge of system***
- ***Ignores decision-making process***
  - *social/economic/political objectives*
  - *decision-makers have specific problems and objectives*
  - *how does information enter the process?*

A Collaborative Approach



- ***Goal:***
  - *increased understanding by decision-makers*
  - *provide relevant and timely information*
  - *involve and inform stakeholders*
  - *document rationale for decisions*

Collaborative Landscape Analysis



Tenet #2

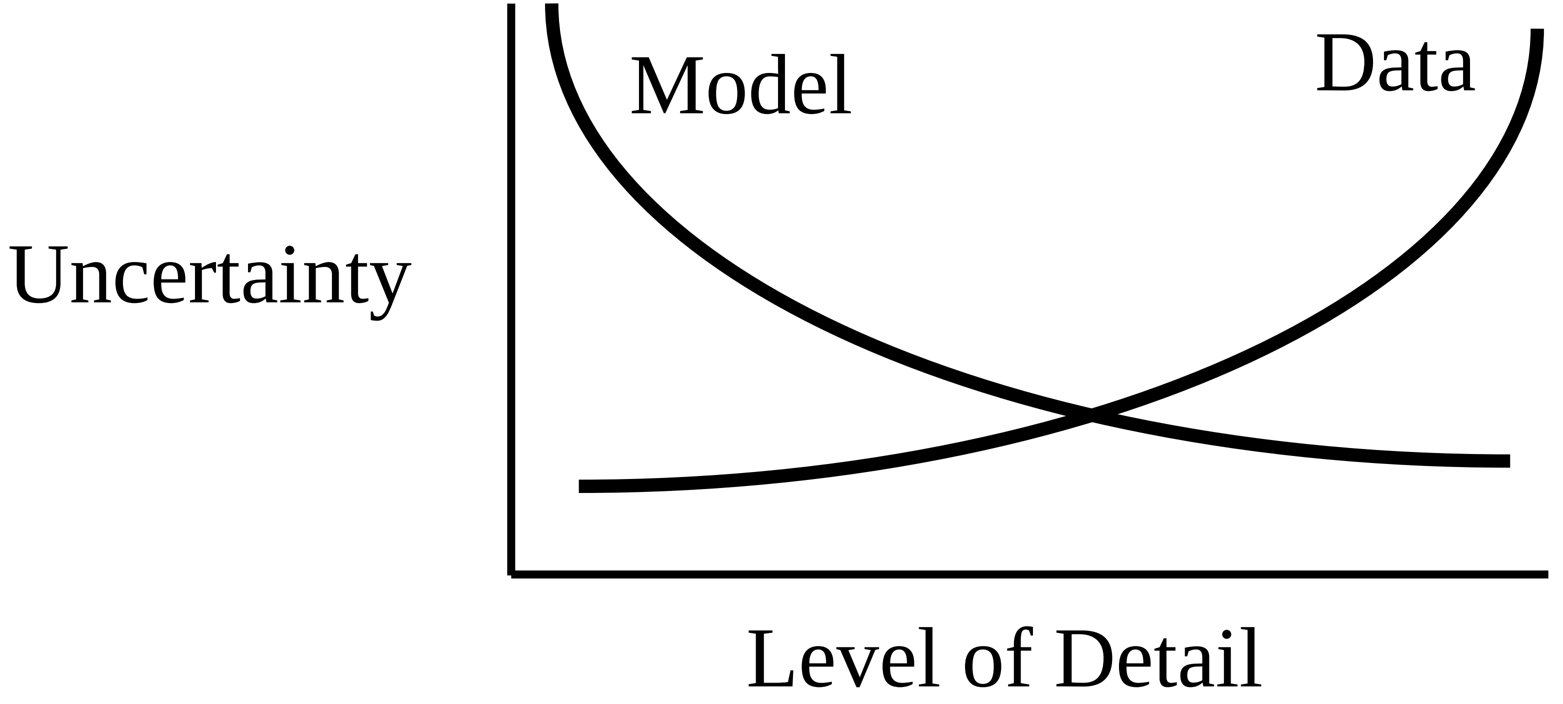
Models should be as simple as possible but not simpler

Albert Einstein

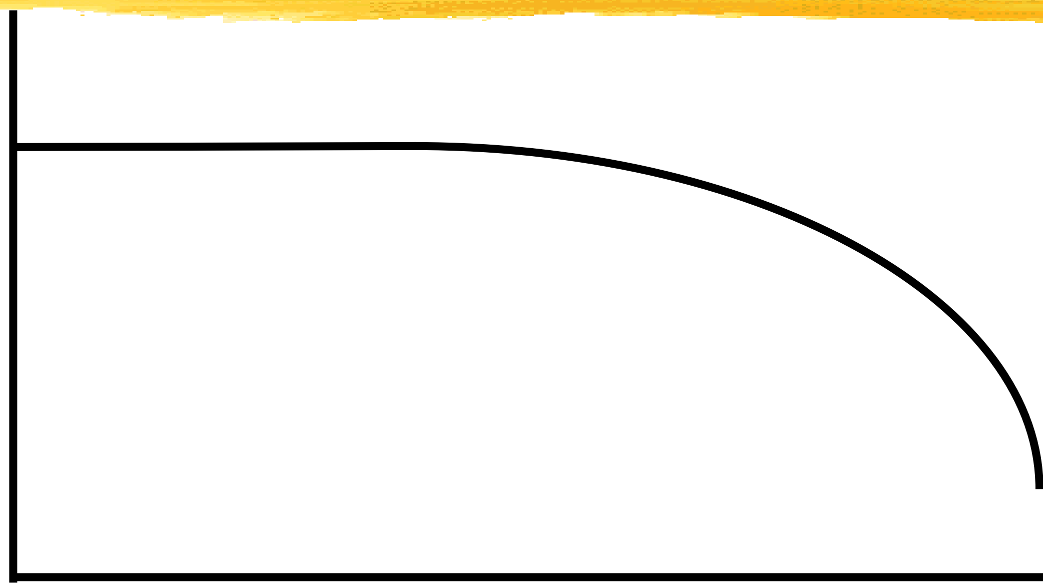
Entia non sunt multiplicanda praeter necessitatem

William of Ockham
(d. 1347)

More Realistic ⇏ Higher Certainty



Higher Precision $\not\Rightarrow$ Higher Accuracy



Representing Π: which is better?

3.14 is very accurate (to 2 decimals), but not very precise

3.145176 is very precise, but not very accurate

Tenet #3

Formal conceptual models are distinct from implemented models

model ≠ program

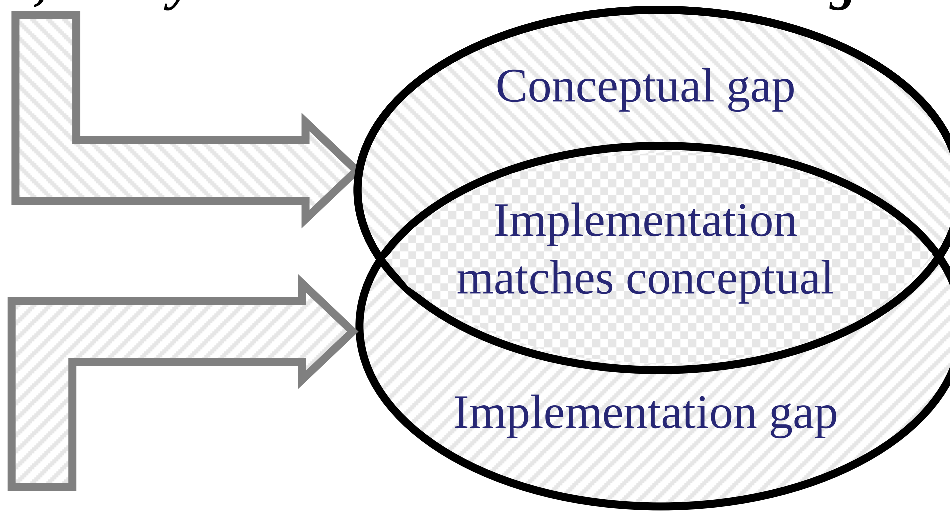
Focus should be on the conceptual models, scenarios and outputs

Beware of the *modelling gap*

Conceptual model ("the model"):

- ***Description and plan for an ascent, including how to deal with fundamental challenges (e.g. knowledge gaps)***
- ***What you want to achieve; how you describe a model in writing***

Modelling gap



Implemented model:

- ***The actual climb, including how unforeseen technical issues are handled and implicit or explicit discrepancies with the plan***
- ***There may be more than one implementation of the same model***

Tenet #4

Model development tools should be flexible and transparent

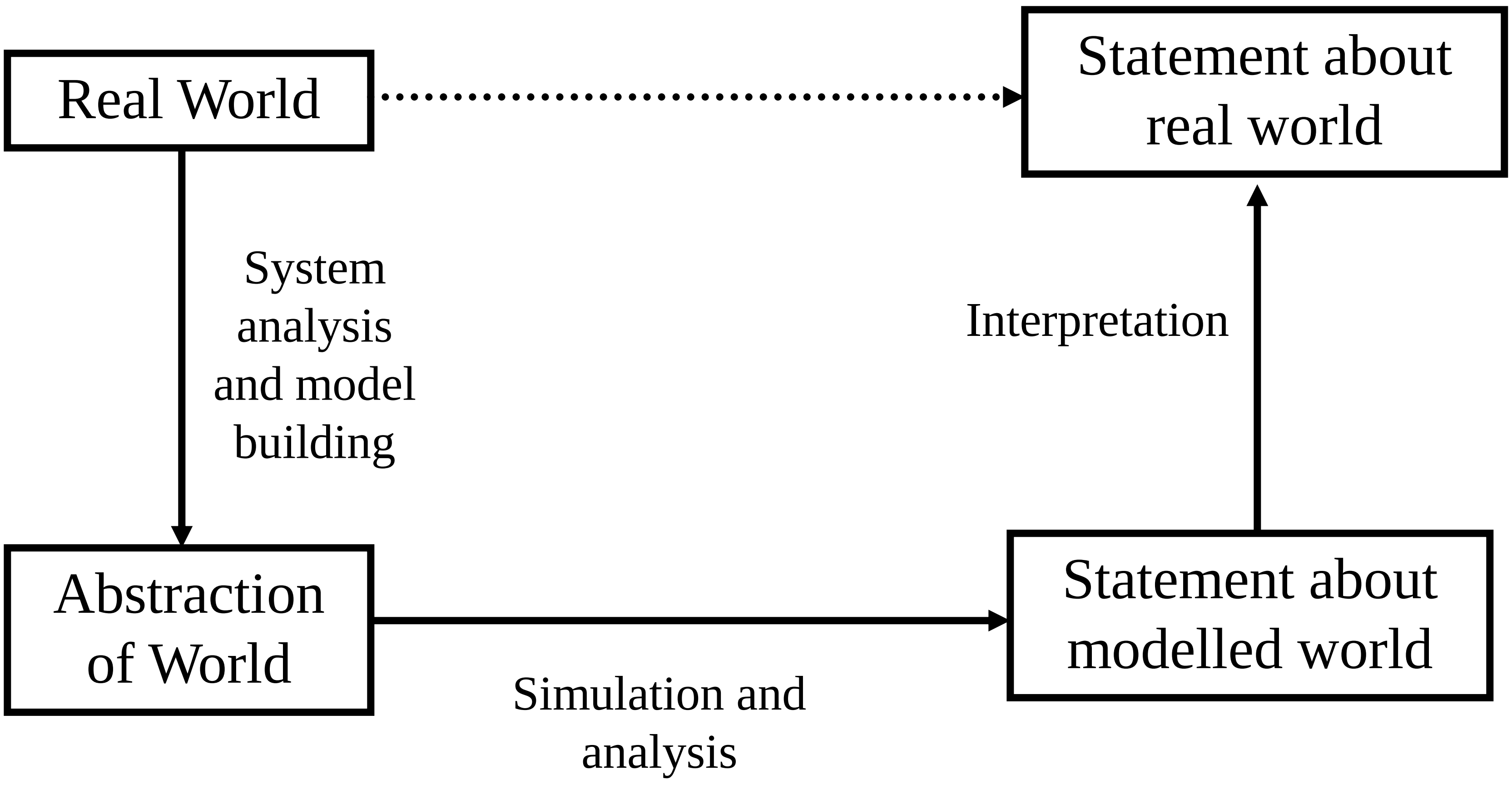
. . . as transparent as possible

"…the [implemented] model … faithfully and faultlessly …forces us to see the implications, true or false, wise or foolish, of the assumptions we have made.

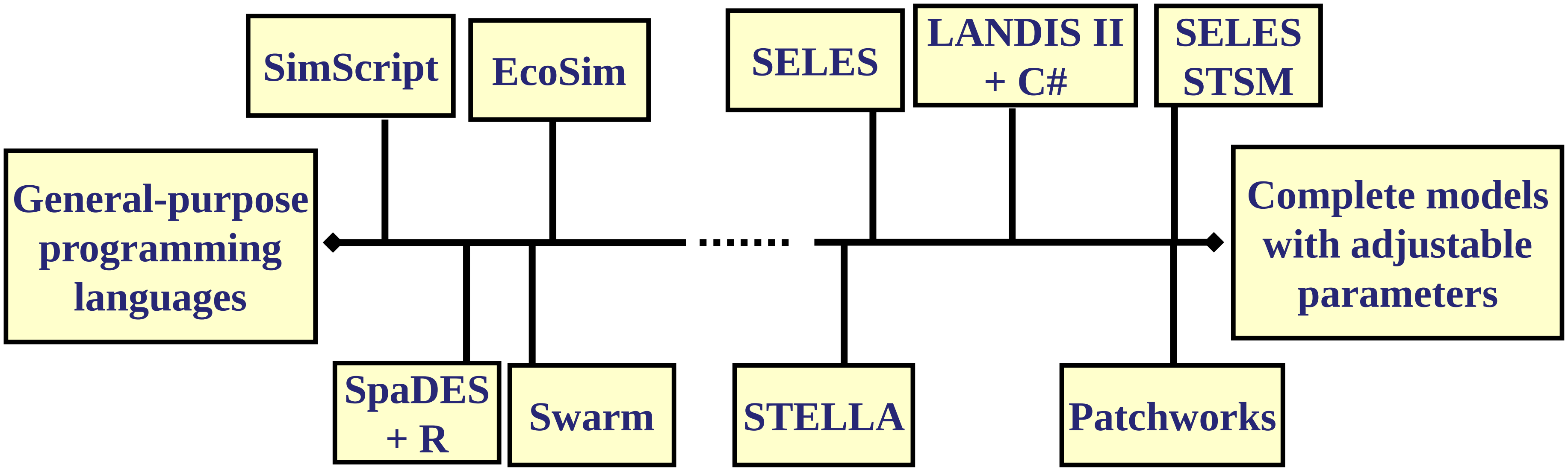
… [a model is] a tool to confront us with the implications of what we think we know"

Botkin (1977)

Making Inferences from Models



Model Development Tools



Program-level support:
- general-purpose
- very flexible class of models
- high implementation cost
- difficult to modify
- usually procedural specification

Model-level support:
- special-purpose
- restricted class of models
- simpler and faster construction
- more declarative specification

Extensible Models vs. Programming Support

Tools that allow for extensibility via "*plug-in*" modules challenge the distinction between a "*highly adjustable model*" and "*programming support for modelling*"

Models and meta-models define a *state space and/or general process behaviour* while programming support does not

- Process sub-models are limited to respond to the available state *and* responsible for making appropriate changes to the available state
  - E.g., if the state-space includes the number of trees/ha, then all processes that modify the forest state must account for changes to this state variable

- tools that specify substantial aspects of the state space are a form of extensible model (e.g. LANDIS II + C#), as are tools that specify general process behaviour with a meta-model (e.g. SELES)
- tools that leave the state-space and all model behaviour to the user (e.g. SpaDES + R) provide programing support for modelling

Goals of Domain-Specific Modelling Tools

• Simple	support rapid model implementation
• Flexible	handle a variety of model types
• Capable	knowledge should be the constraint
• Modular	model decomposition
• Transparent	assumptions should be explicit
• Efficient	able to process large, complex models
• Adaptable	modify components to other projects

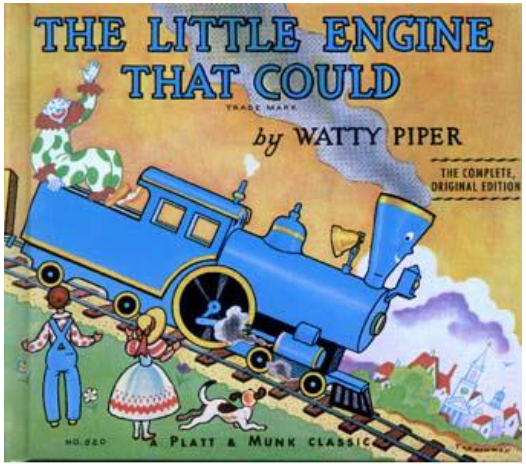
SELES: Spatially Explicit Landscape Event Simulator

A general tool for building models of landscape dynamics

– a language for specifying models of landscape dynamics

and

– a simulation engine for running these models



SELES Paradigm: Spatio-temporal Contexts

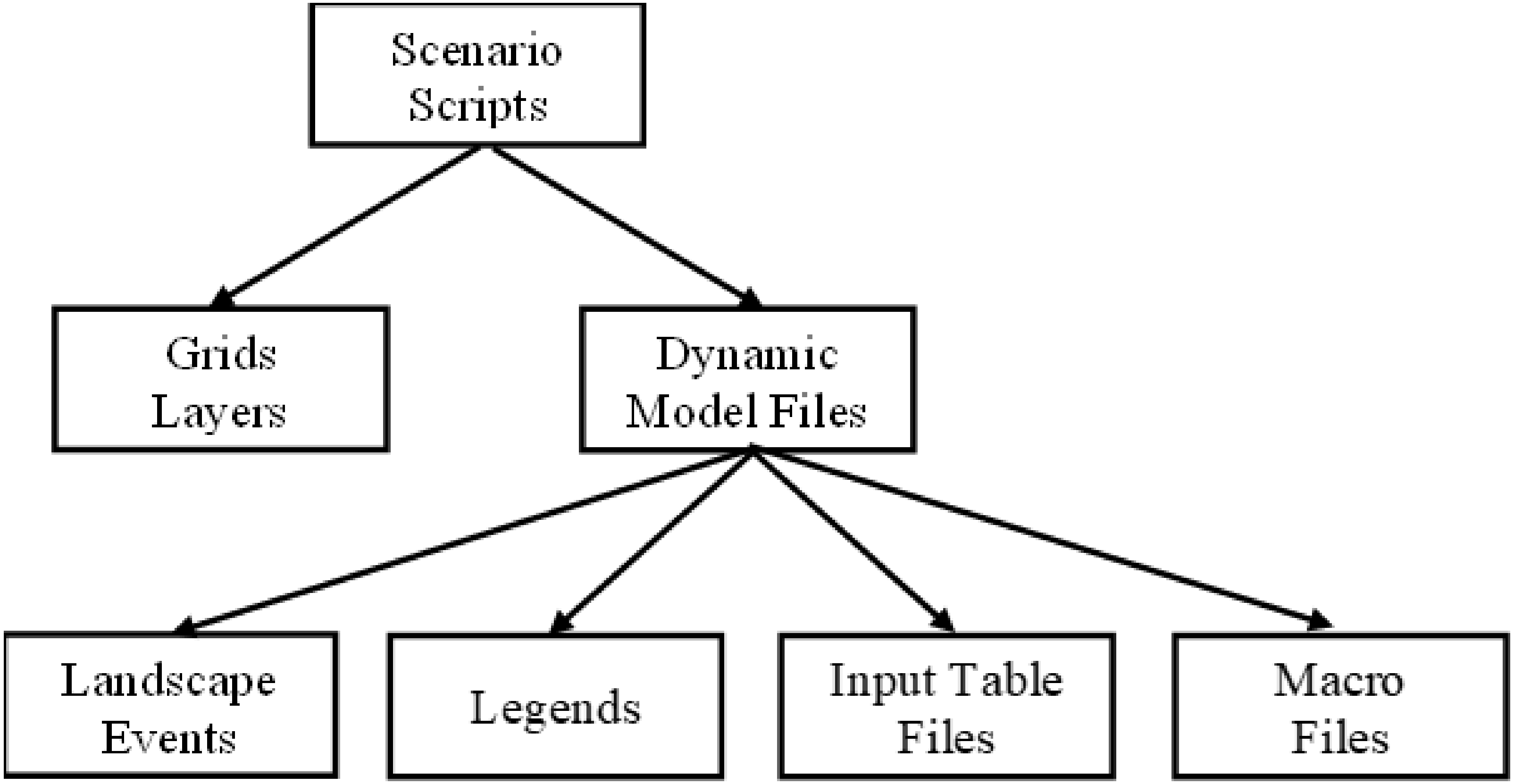
- landscape change arises as the result of feedback between system state and definable *processes* or *entities*
- as *agents of landscape change*, processes react to and modify the landscape state in *spatio-temporal contexts*
- a spatio-temporal context is the set of information (i.e. state variables) available at a particular time and place
- contexts provide a general hierarchical framework for describing landscape dynamics

SELES Paradigm: Spatio-temporal Contexts

- SELES is a language for:
  - creating a spatio-temporal state-space
  - defining behaviours to navigate through this state-space
  - specifying state changes along the way

- By managing contexts appropriately, models of various forms can be created, including:
  - natural disturbance models
  - habitat supply models
  - timber supply models
  - spatial population and meta-population models
  - individual-based models

SELES Model and File Structure



SELES Framework and Training Overview

- Module 2: Conceptual basis (SELES paradigm)
- Module 3: User interface (SELES the program)
- Module 4: Running models (driving)
- Module 5: Reading models (transparency)
- Module 6: Exploring the landscape event meta-model
- Module 7 Writing models – case study (constructing)
- Module 8: Designing models

  - Modules assume that prior modules have been completed

SELES Facets

Facet 1 (module 3): as a data exploration tool (a GIS perspective)

- user interface and visualization

Facet 2 (module 4): as a simulation tool (a driver perspective)

- setting up and running existing models (scenarios/experimentation)

Facet 3 (modules 5): as a modelling platform (a mechanic perspective)

- understanding existing models
- conceptual basis and language (landscape events/agents)

Facet 4 (modules 6 & 7): as a modelling platform (a constructor perspective)

- modifying and building models
- debugging and verification

Facet 5 (module 8): as a model factory (a designer perspective)

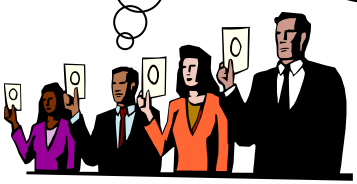
- designing models that fit the SELES paradigm
- landscape systems ecology

Modelling roles

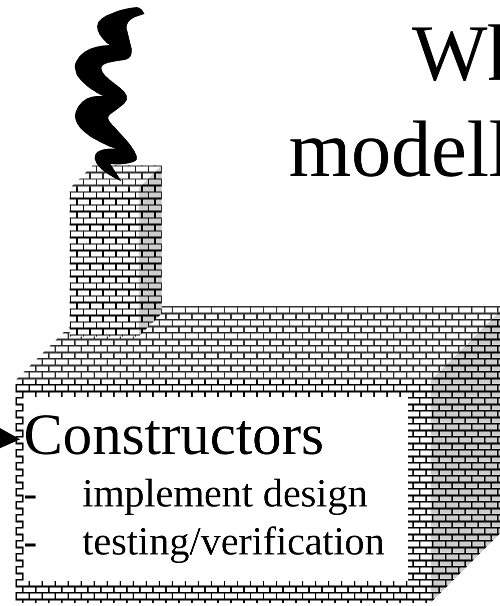
Who are you?

What are your modelling objectives?

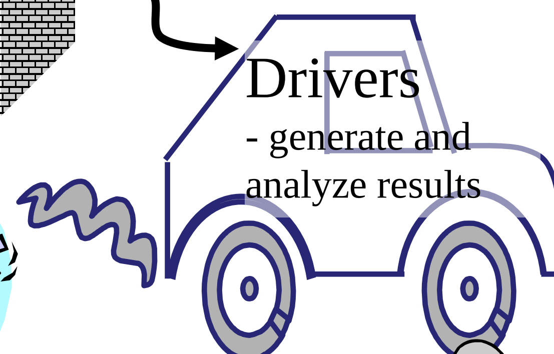
Clients
-set objectives
-use results



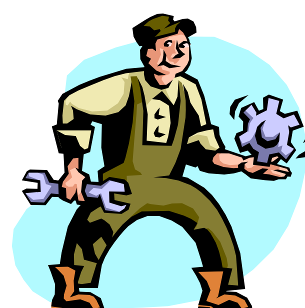
Designers
- model and domain knowledge
- create blueprint (conceptual model)
- evaluate/interpret results



Constructors
- implement design
- testing/verification



Drivers
- generate and analyze results



Mechanics
- adapt and modify existing models