OPPA European Social Fund
Prague & EU: We invest in your future.

Agent-based Modeling and Simulation

Michal Jakob and Michal Pěchouček
Agent Technology Center, Dept. of Computer Science and Engineering, FEE, Czech Technical University

AE4M36MAS Autumn 2012 - Lect. 13

OTEVŘENÁ INFORMATIKA

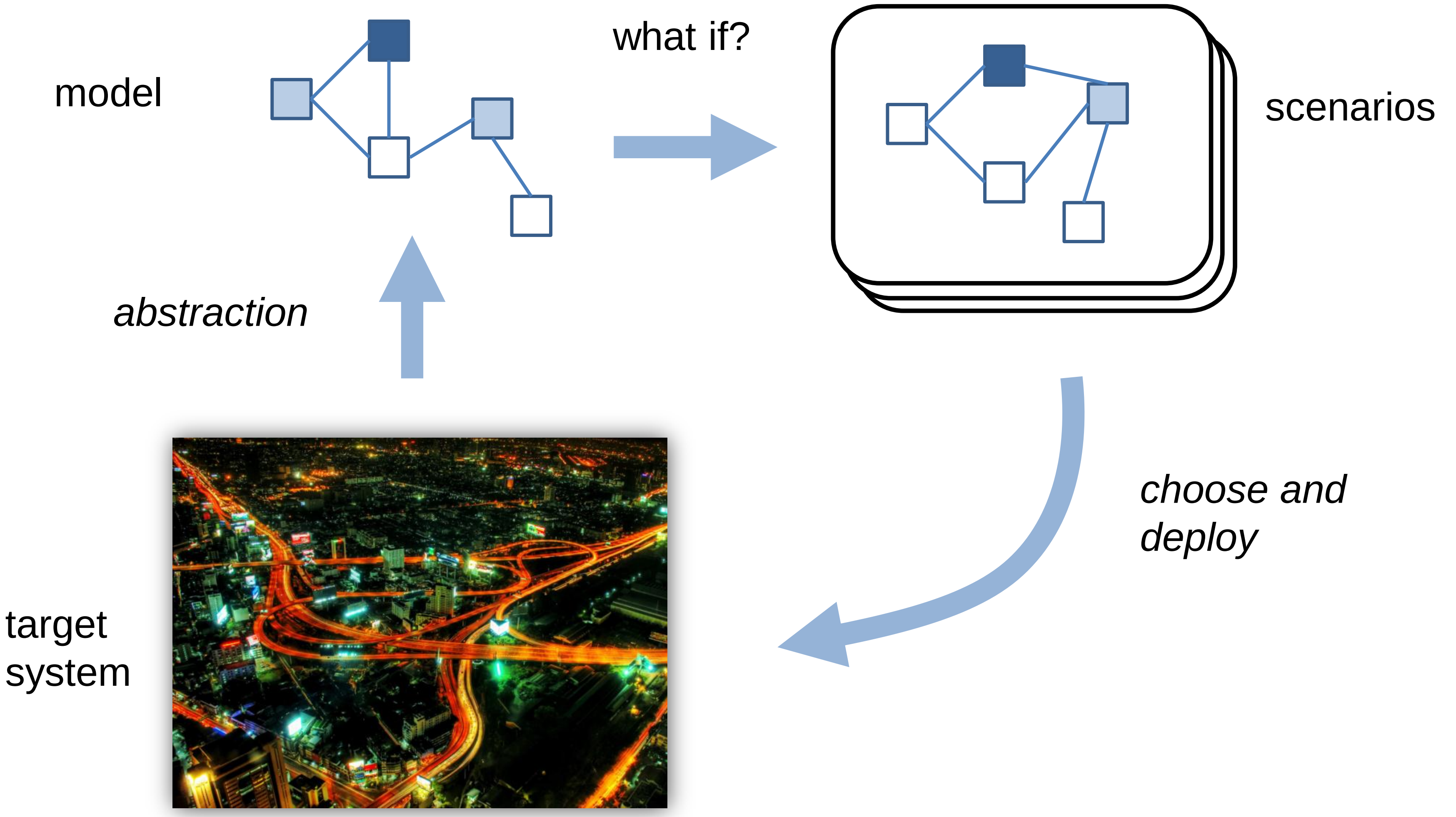
Motivation

- We live in an **increasingly complex world.** Systems that need to be analyzed are becoming more complex
  - Decentralization of Decision-Making: “Deregulated”electric power industry
  - Systems Approaching Design Limits: Transportation networks
  - Increasing Physical and Economic Interdependencies: infrastructures (electricity, natural gas, telecommunications)
- In complex adaptive / interconnected multi-agent systems
  - Extrapolating past does not always work
  - Intuition may be misleading
- ➔ We need computation tools to assist us in understanding and improving the operation of such systems.

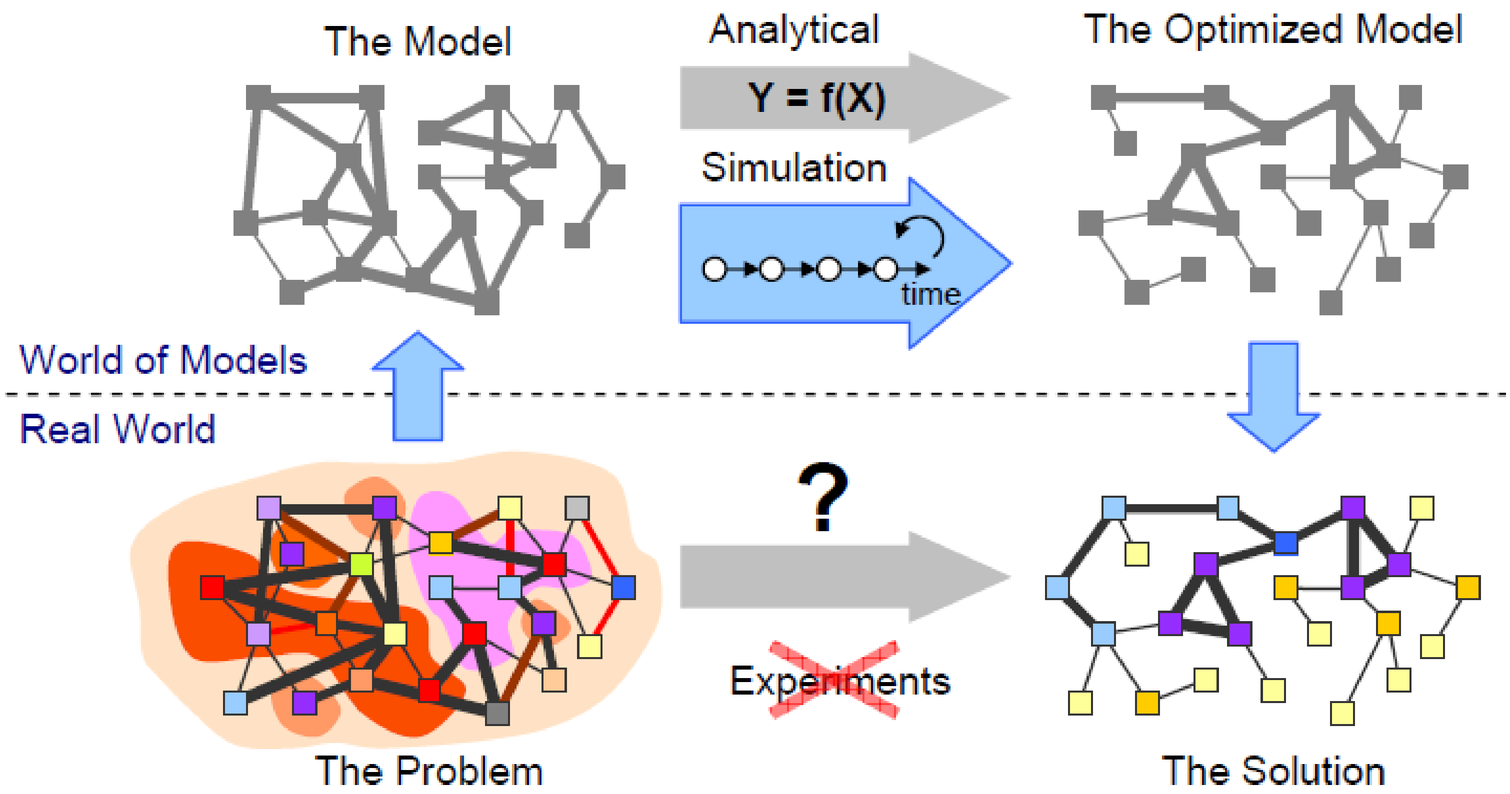
Computational Modeling / Simulation

- Computational modeling / computer simulation is a powerful tool for obtaining **insight** and **foresight** regarding the operation of complicated systems

Modeling Cycle

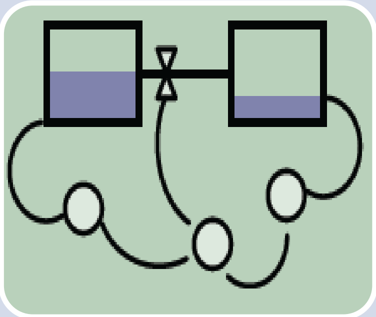


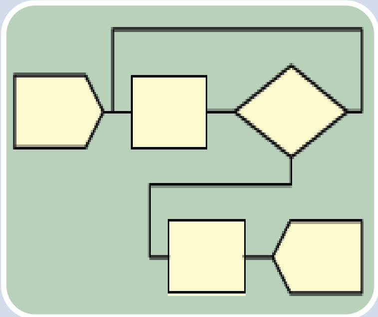
Beyond Insight: Simulation-based Optimization

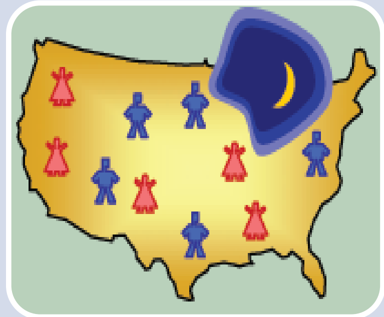


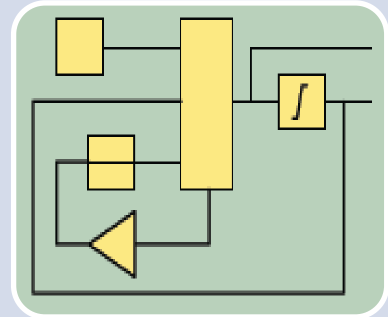
From: Borshchev, A. et al (2004): From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools

S&M Approaches









System dynamic

- states, feedbacks and delay structures
- continuous
- global, aggregate view

Discrete Event

- entities and resources
- discrete, event-based
- global entity processing algorithm

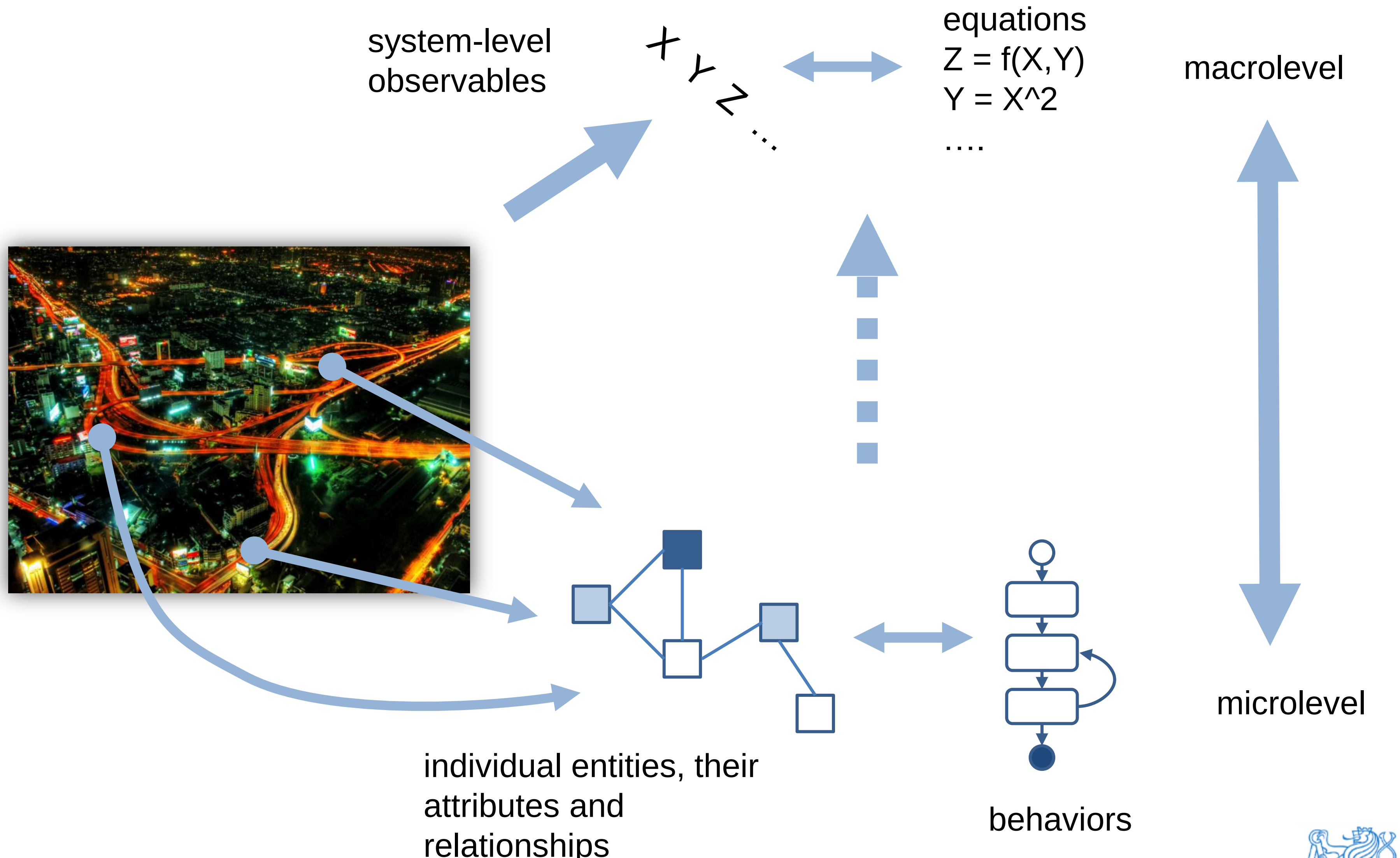
Agent-Based

- active entities within an environment
- decentralized, individual perspective
- **global behavior emerges**

Dynamic Systems

- state variables and differential equations
- direct physical meaning, no aggregation

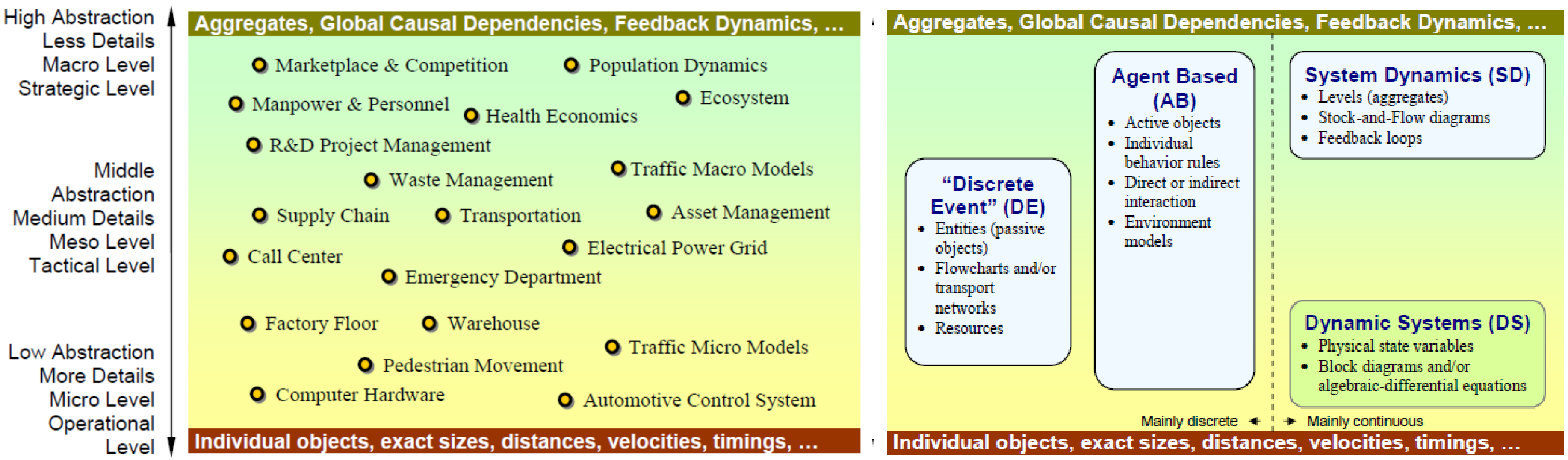
Top-down (Equation) vs. Bottom-up (Agent) –based Approach



Agent-based Simulation

- Based on localized (micro-) behaviours and interactions
- State and state updating is distributed throughout the entities of the model
- No high-level, fixed process structure (but structure can emerge dynamically)

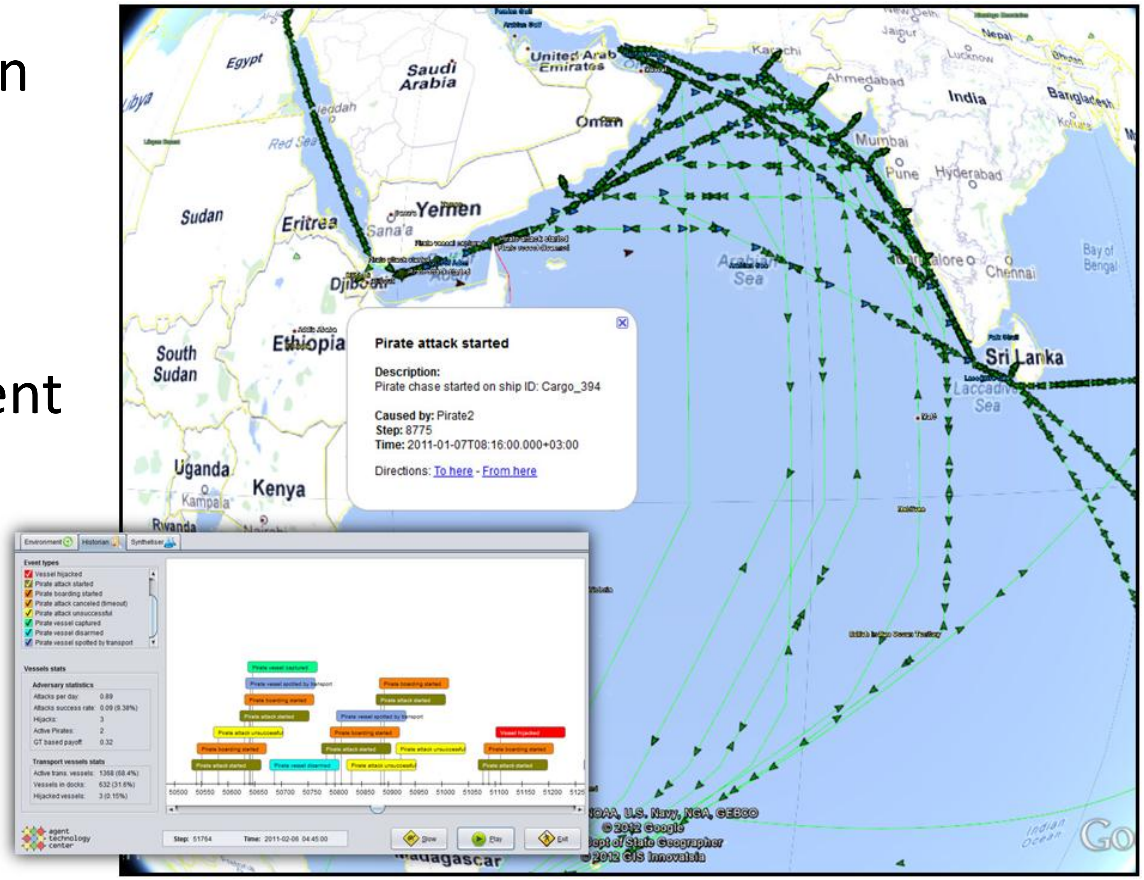
Levels of Abstraction



From: Borshchev, A. et al (2004): From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools

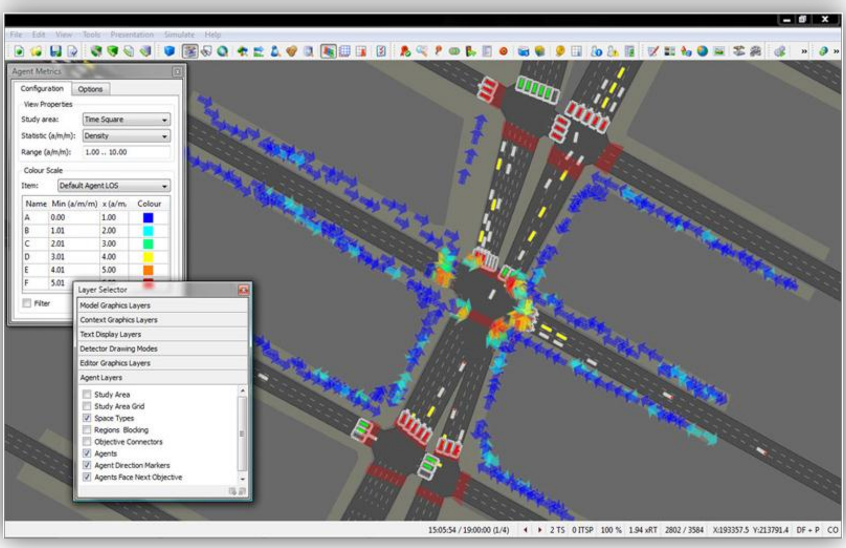
Illustrative Examples: Maritime traffic and piracy modelling

- Modelling movement and activity of vessels in piracy-affected waters
- Allows assessing the efficiency of counter-measures under different circumstances



Illustrative Example: Crowd Modelling

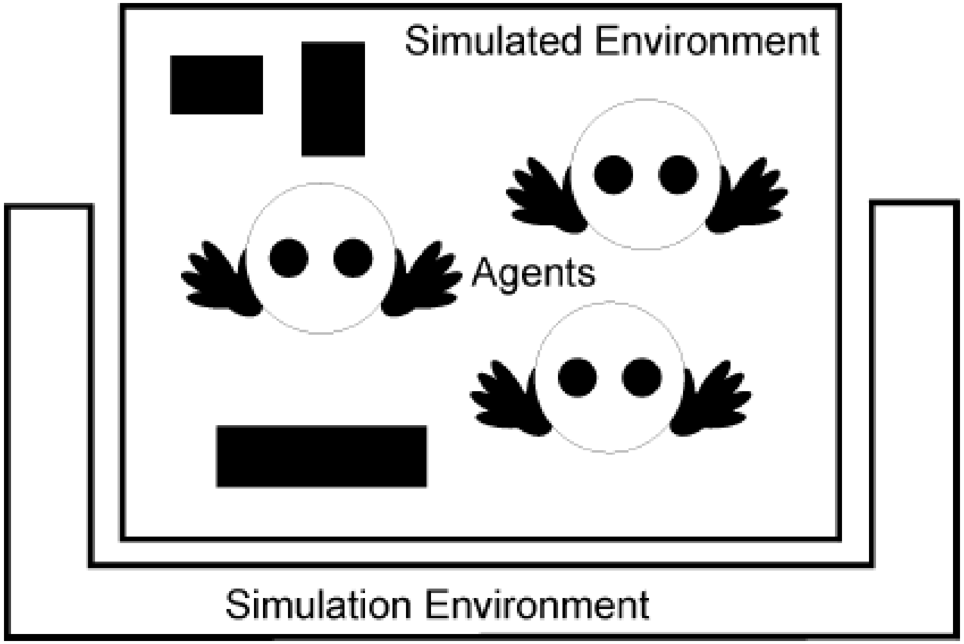
- Pedestrian simulation
  - Each pedestrian modeled as an agent sensing the environment and interacting with other pedestrian agents
- The model allows
  - determining crowd flows and densities under various scenarios
  - optimizing crowded public spaces for capacity, comfort and safety



video →

Architecture Agent-based Simulation Models

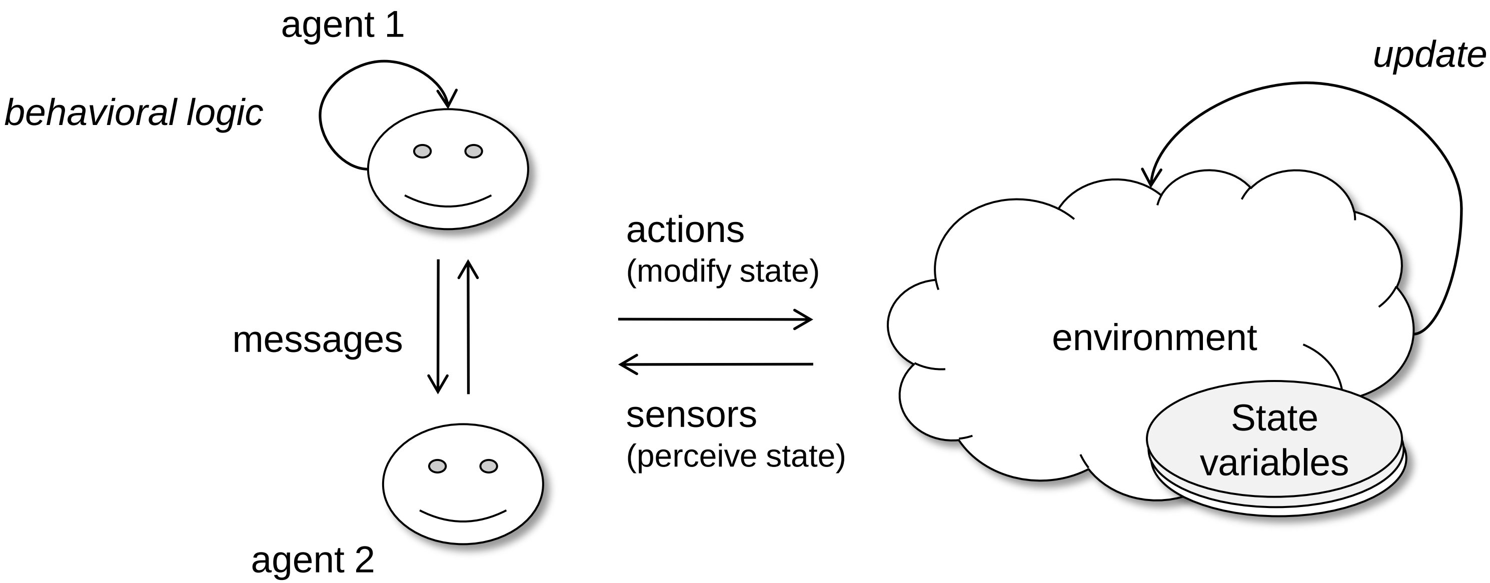
Structure of Agent-based Simulations



Model

Simulator / simulation platform / infrastructure

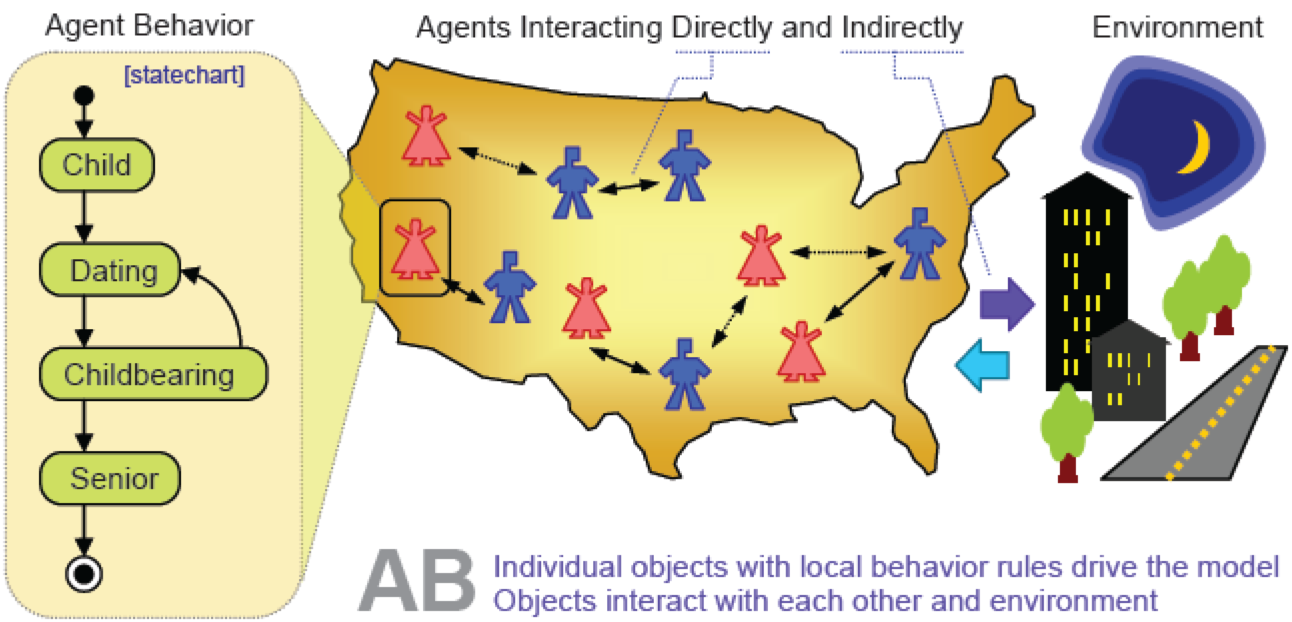
Structure of Agent-based Simulations



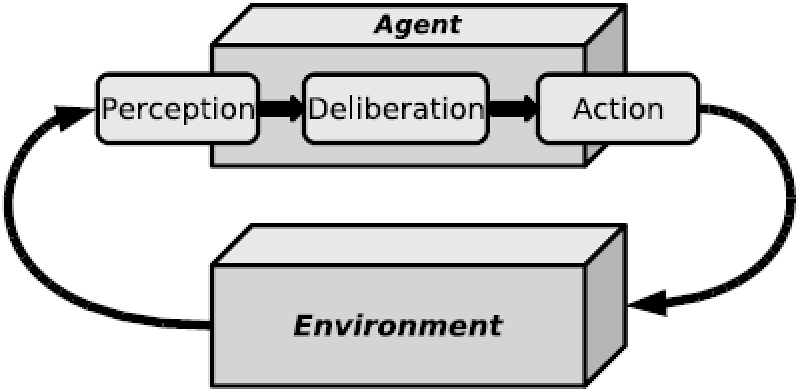
Agents drive the model through local behaviors and direct and indirect interaction with each other and with the environment

Environment state is modified by agent actions and/or agent-independent/passive processes (e.g. weather)

Structure of Agent-based Model



Agent Behaviour Representation



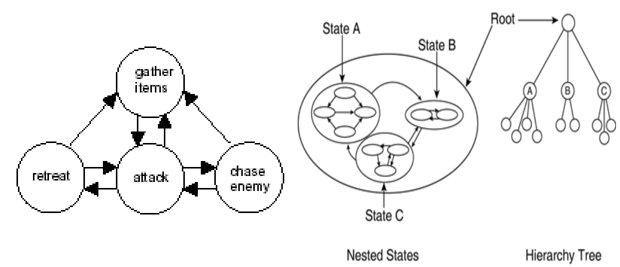
1. Simple / Reactive architecture
2. Complex / Cognitive /Deliberative architecture

Agent Behavior: Simple Approaches

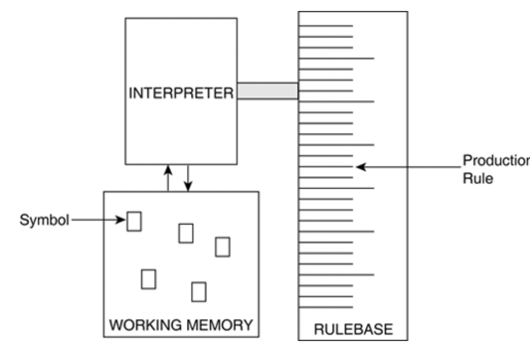
- scripts

```
(if(Said('look'))
    Print("You are in an empty room")
)
(if(Said('take/key'))
    (if(send gEgo:has(INV_KEY))
        Print("You already have it!")
    )(else
        (if(send gEgo:inRect(150 150 170 170))
```

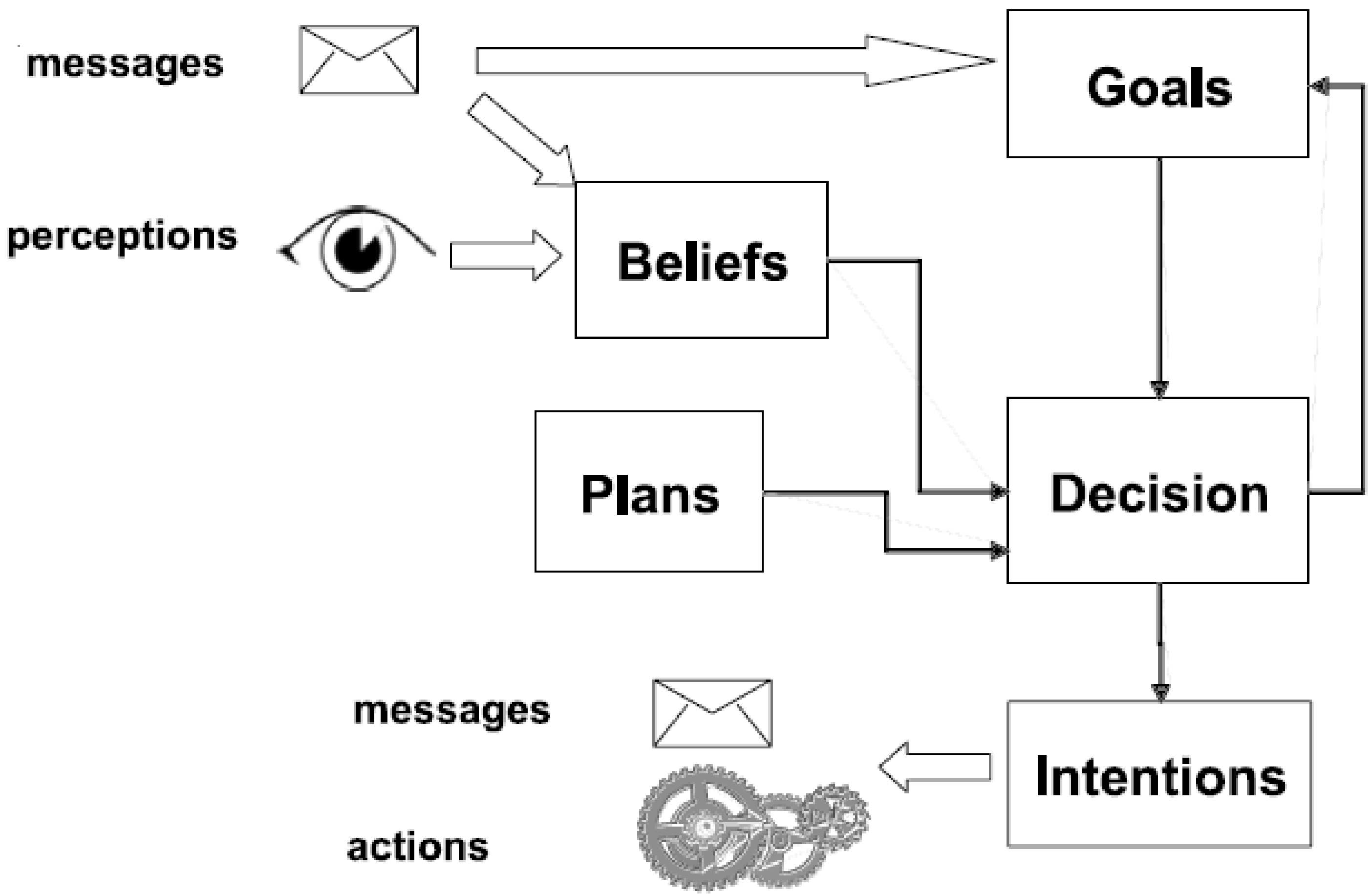
- (hierarchical) finite state machines



- rule engines



Agent Behavior: Complex Approaches



Agent Behavior: Complex Approaches

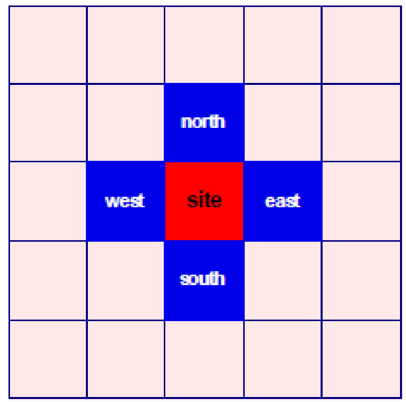
- **Belief Desire Intention (BDI)** Architecture
  - AI-based
  - aims to maximize agent’s performance (utility)
- **Cognitive Agent Architecture**
  - biologically / cognitive science-based
  - aims to realistically replicate human cognitive biases / limitations
- Both computationally very heavy => not suitable for models with many agents

Interaction Topologies / Spatial Structure

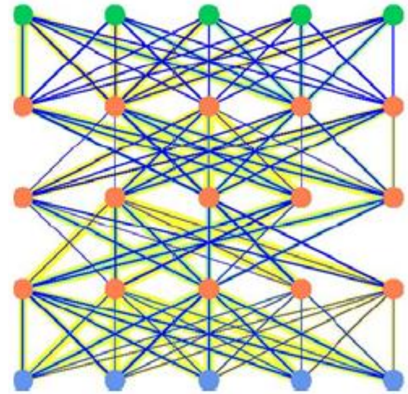
Abstract

Euclidian space (2D, 3D)

Grid

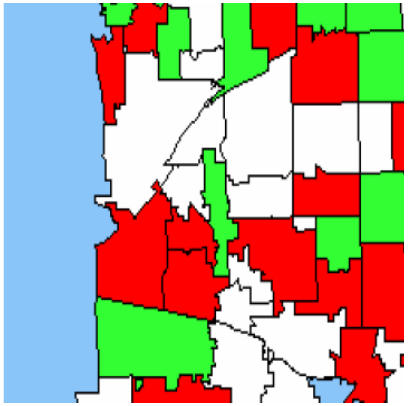


Network

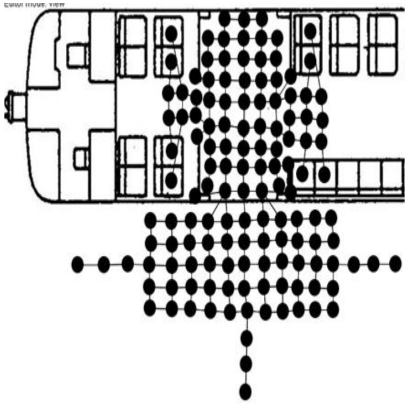


Realistic

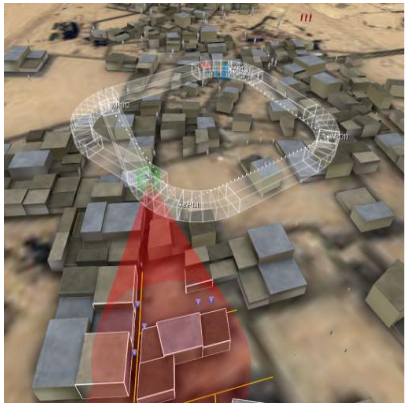
GIS



Structured 2D

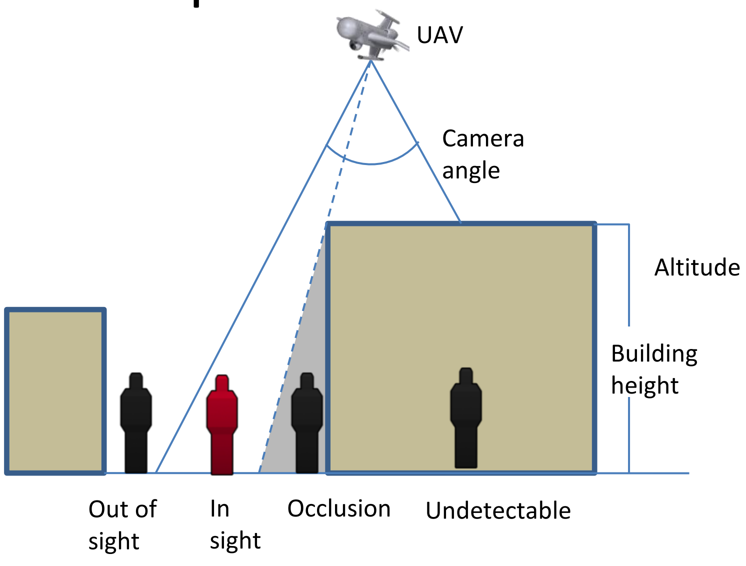


Complex/structured 3D



Sensors

- Enables the agent to access environmental state
  - low-level – direct perception (e.g. image from a camera)
  - high-level – interpreted scene (e.g. walls, people)
- Push vs. Pull sensors
- Efficient implementation crucial in more complex environments
  - partitioning
  - caching
- Examples: Detecting a nearby pirate vessel, observing traffic lights



example sensor model

Actions

- Describe how agents can affect the environment state
- Can be instant or take some time
- Can be deterministic and probabilistic
- Joint-actions also possible
- Examples: hijacking a vessel, boarding a bus, walking to a next junction

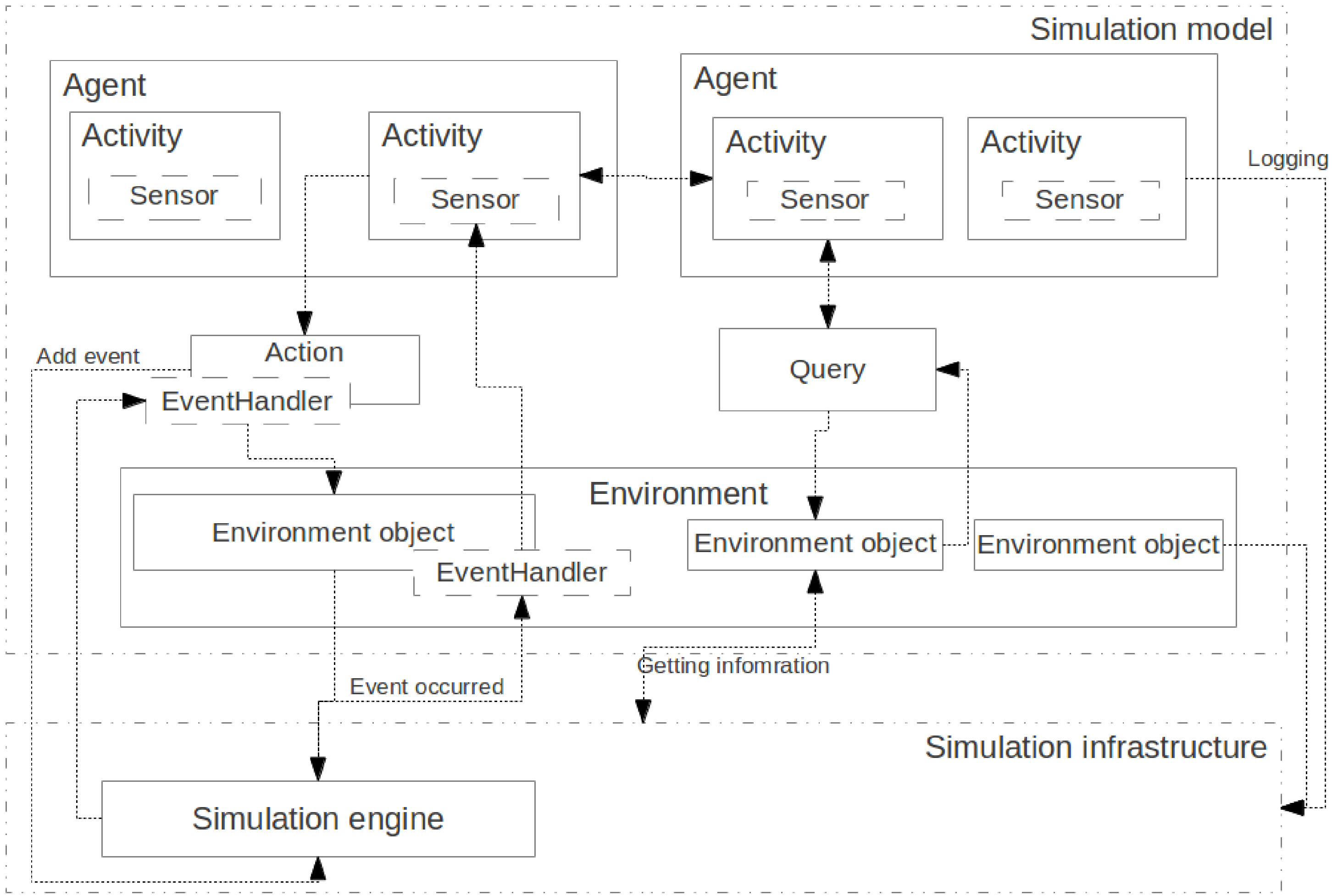
Communication

- Models explicit message-based interaction between agents
  - implicit interaction modelled through actions and sensors
- Two components
  - content
  - protocols
  - can be based on general agent communication languages (ACL) but typically simpler
- Different level of environment-affected mediation possible
  - distance and/or line of sight restrictions
  - noise / unreliable link
- Example: distress call to a navy vessel, ordering a taxi

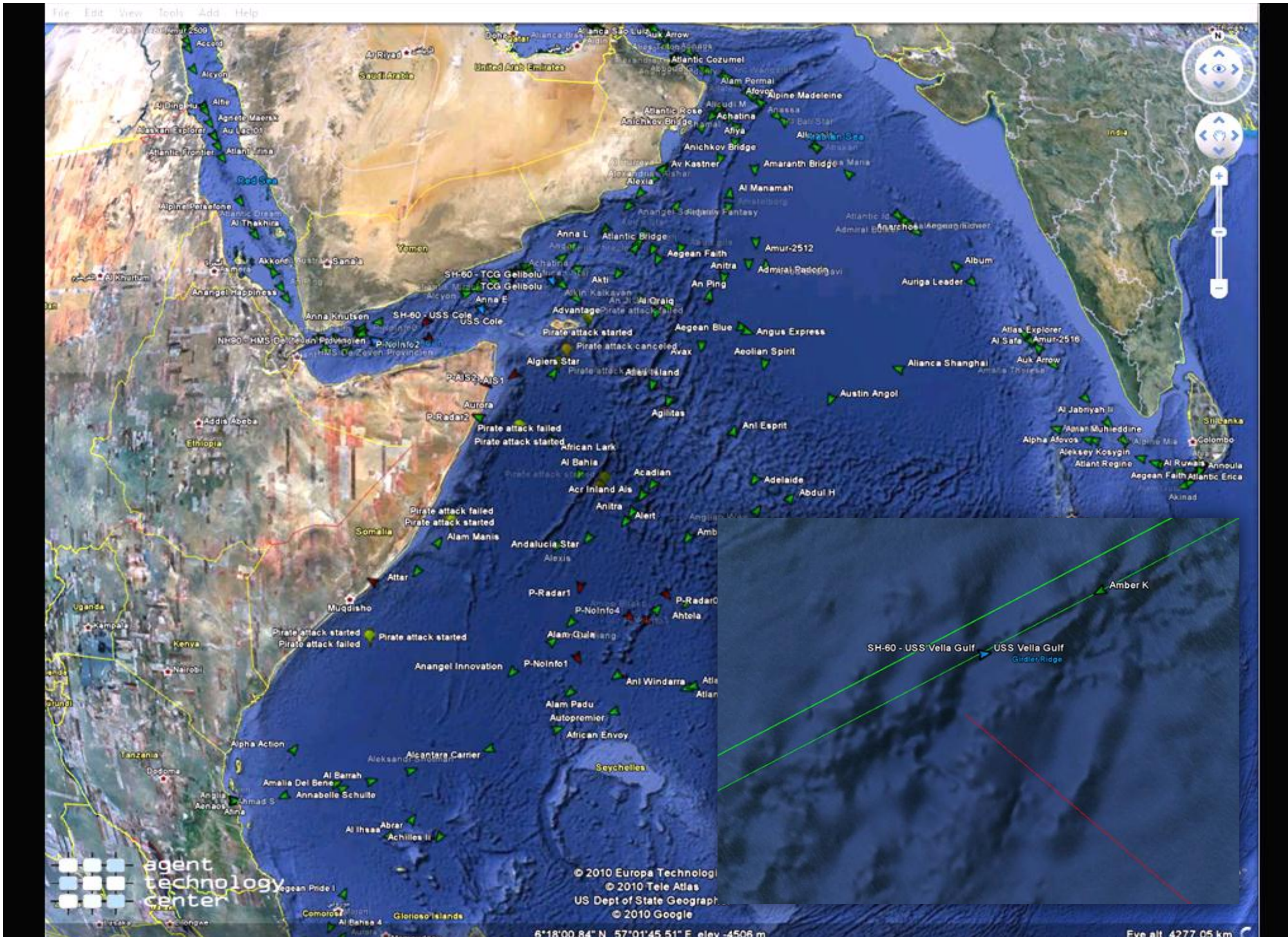
Simulation Platform / Infrastructure

- Initialization
- Scheduling/handling state and sensor updates
- Logging and reporting
- Parallelization / Distribution
- Design of experiments

Simulation Architecture (AgentPolis)

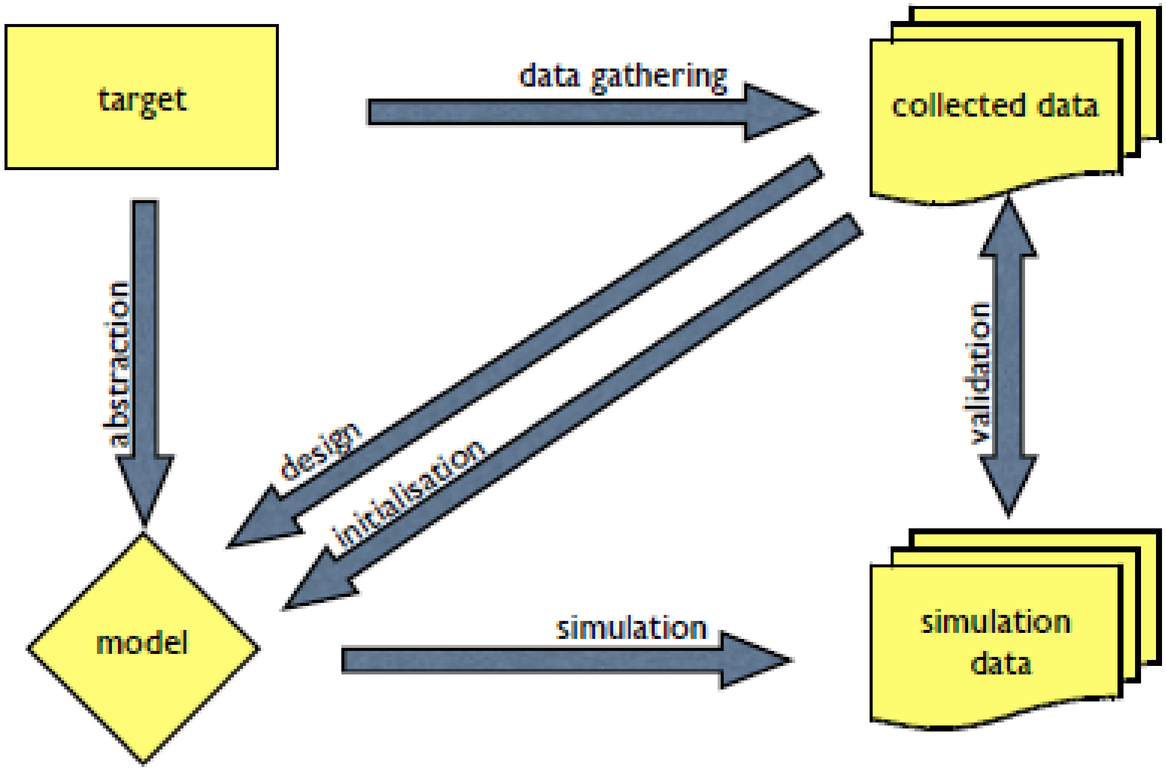


AgentC Example

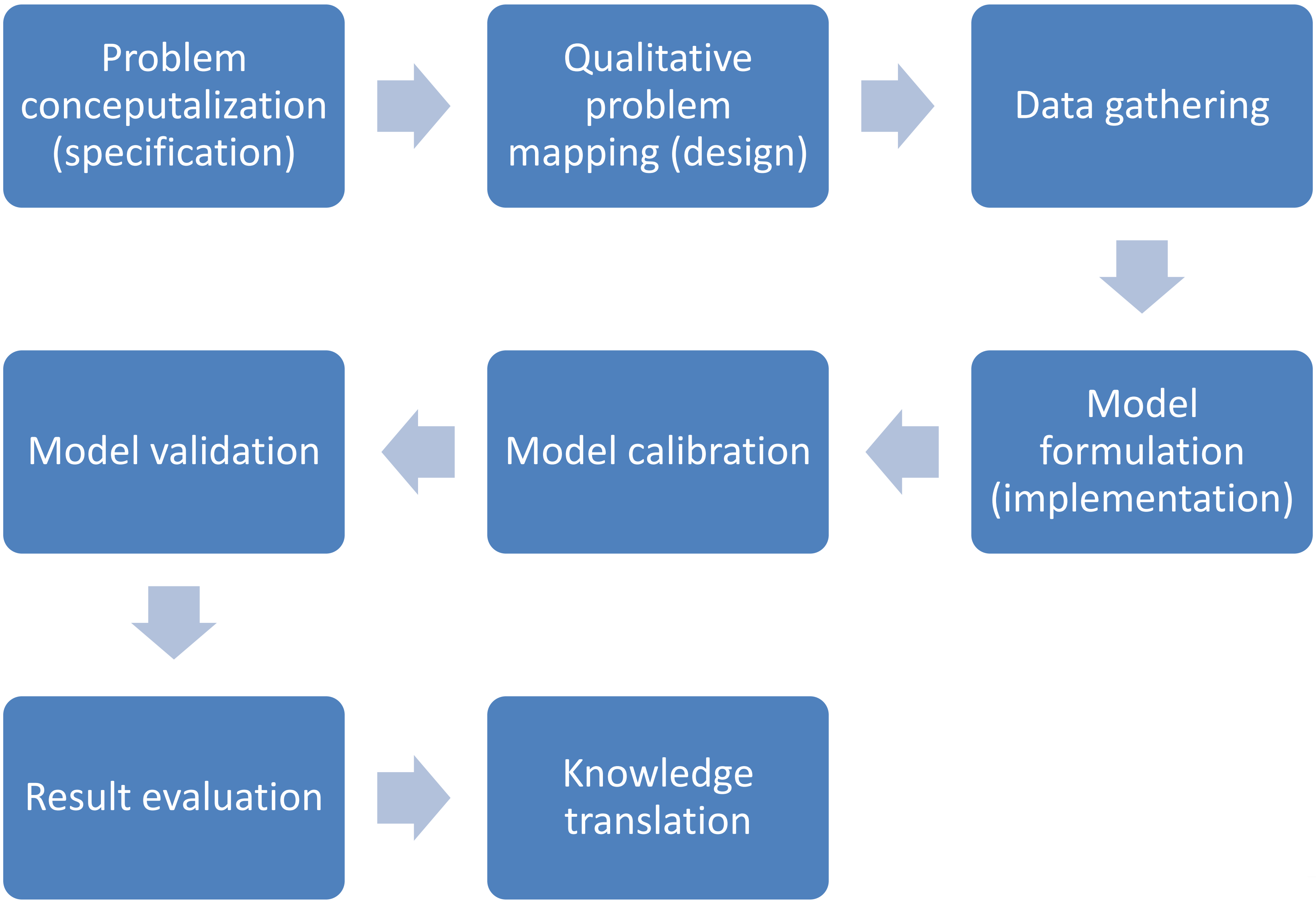


Developing agent-based simulations

(Data-driven) Simulation Process



Model Development Process



Problem conceptualization (model analysis and specification)

- Problem/research question articulation
- Model scope/boundary selection
  - endogenous vs. exogenous vs. ignored
  - purpose is king: only add features to the model if necessary
  - level of detail
- Key entities & their relationships
  - agents (&collectives)
  - environment
  - nesting hierarchy and/or interaction networks
- Model outputs of interest
- Data

Conceptualizaton Example (AgentC)

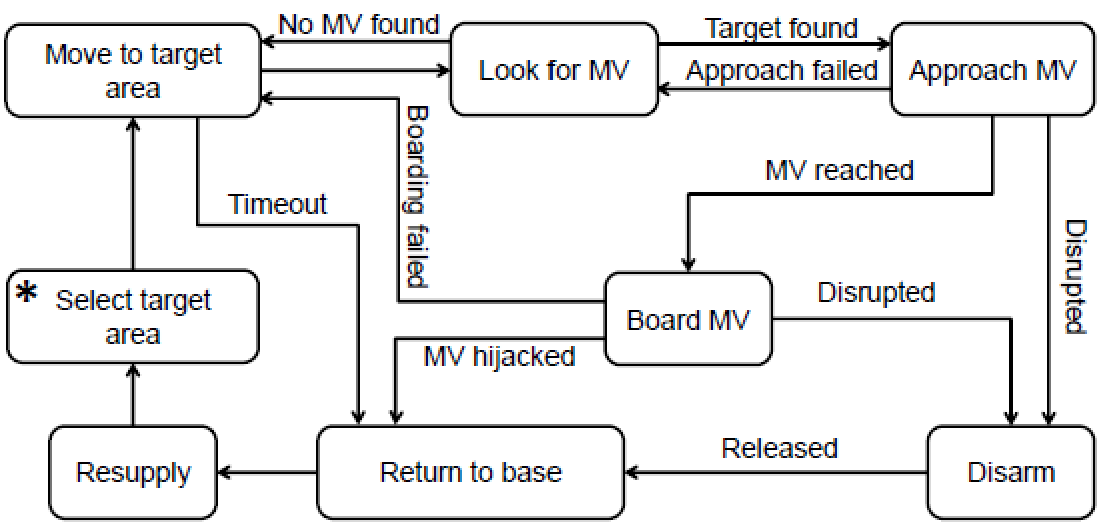
- Scope:
  - area of interest: Gulf of Aden and Indian Ocean
  - time of interest: 2005-now
  - attacks (endogeneous), weather (exogeneous), currents (ingored)
- Key entities
  - vessels: merchant, pirate and navy
  - environment: navigable waters, corridors, ports and anchorages
  - interactions: pirate attack
- Model outputs
  - attack statistics, transit distance and duration
- Data
  - merchant traffic patters, pirate incidence statistics, vessel operational parameters,…

Model Design

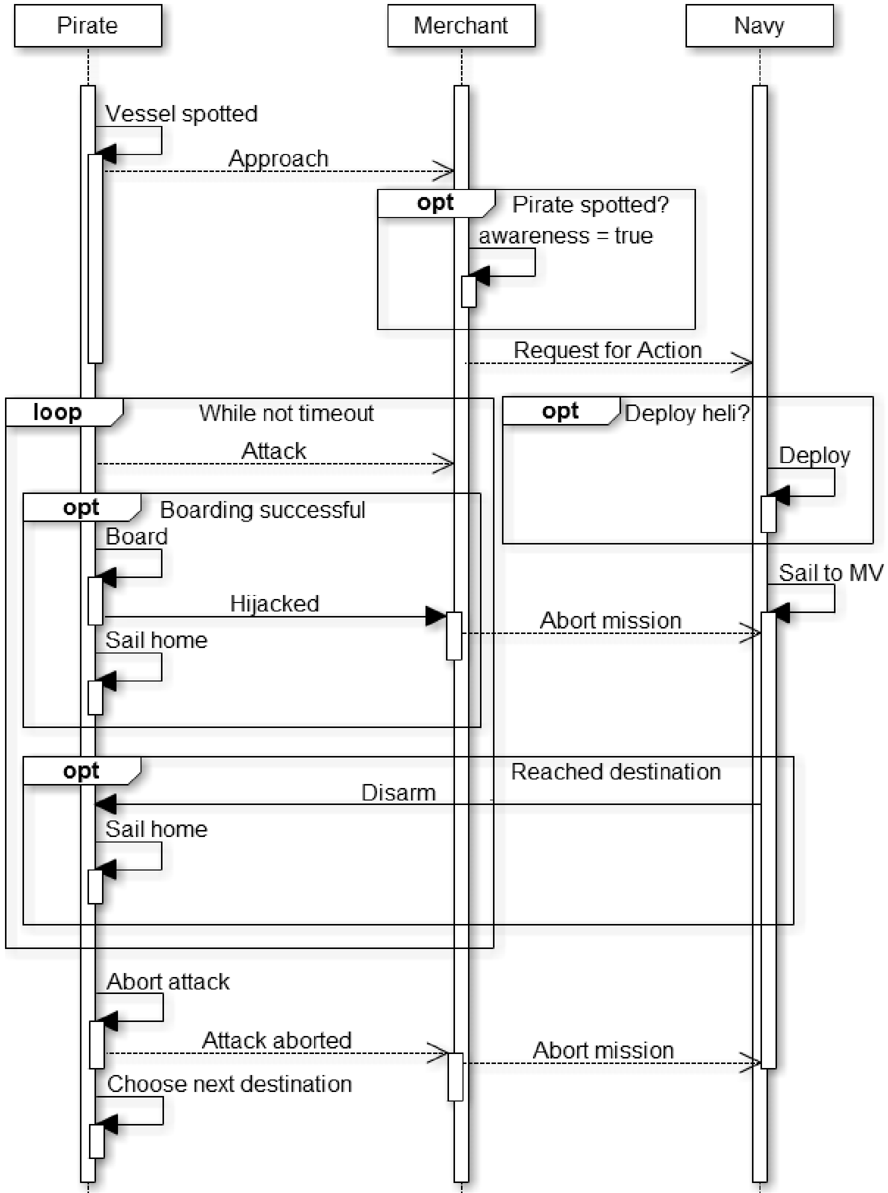
- Parameter & state variables identification
- Behavioral fragments
- Interaction diagrams
- Evnironment objects
- Actions and sensors
- Key events
- Output metrics
- Three approaces
  - agent-driven
  - interaction-driven
  - environmet-driven

Model Design Example (Pirate vessel)

Parameter	Values
Home anchorage	base id
Cruising speed	[8, 14] kn
Pursuit speed	[25,30] kn
Endurance	[7, 21] days
Visibility radius	[5, 12] nm
Attack time	30 min
Cool-down time	[1, 4] hr
Navy knowledge	[0, 1]
Hijack prob. ρ_u	[0, 1]
Hijack prob. ρ_a	[0, 1]



Model Design Example (Pirate attack)



Parameter	Values
M Cruising speed	[10, 20] kn
M Alertness	[0, 60] hr^{-1}
M Awareness	Y/N
P Visibility radius	[5, 12] nm
P Pursuit speed	[25,30] kn
P Attack time	30 min
P Hijack prob. ρ_u, ρ_a	[0, 1]
N Helicopter	Y/N
N Action radius	[100, 200] nm
N Helicopter speed	[140, 170] kn
N Cruising speed	[20, 30] nm

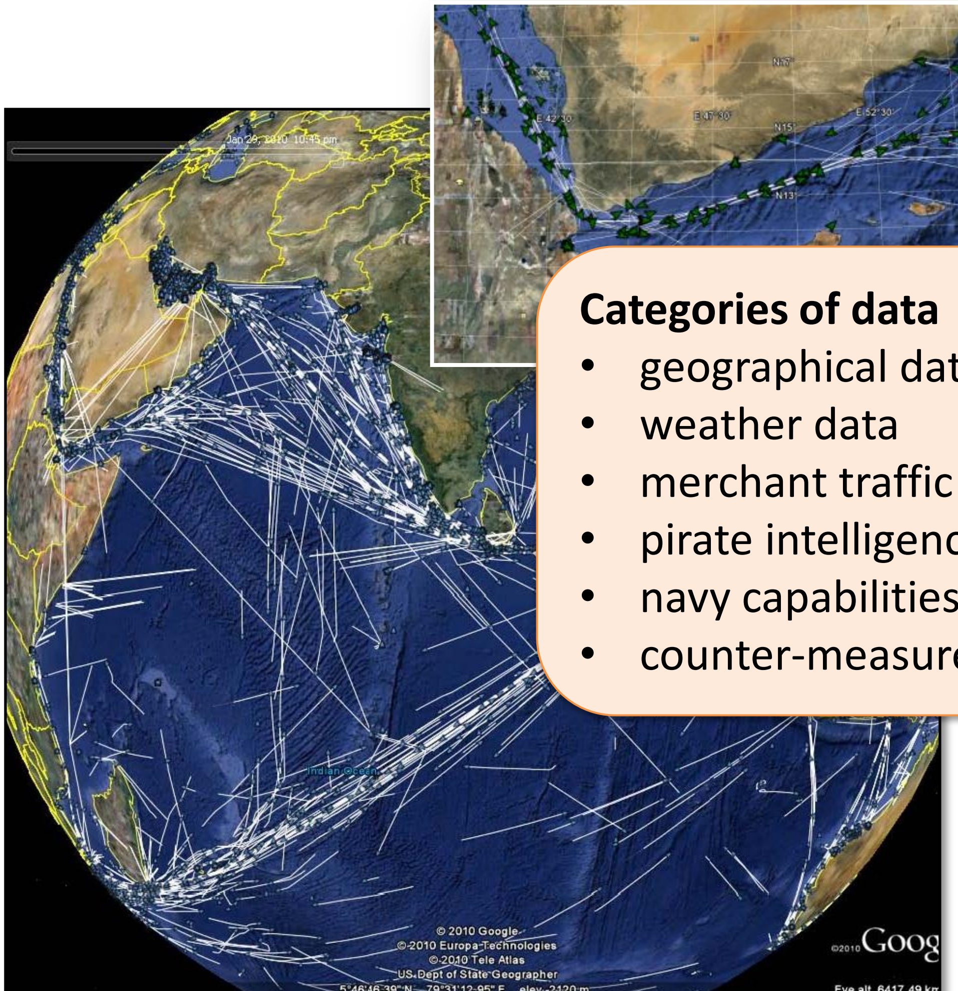
Data Collection and Preprocessing

- Dataset acquisition
- Data selection and filtering
- Data cleaning and quality checking
- Import / format conversion
- Database / data store creation

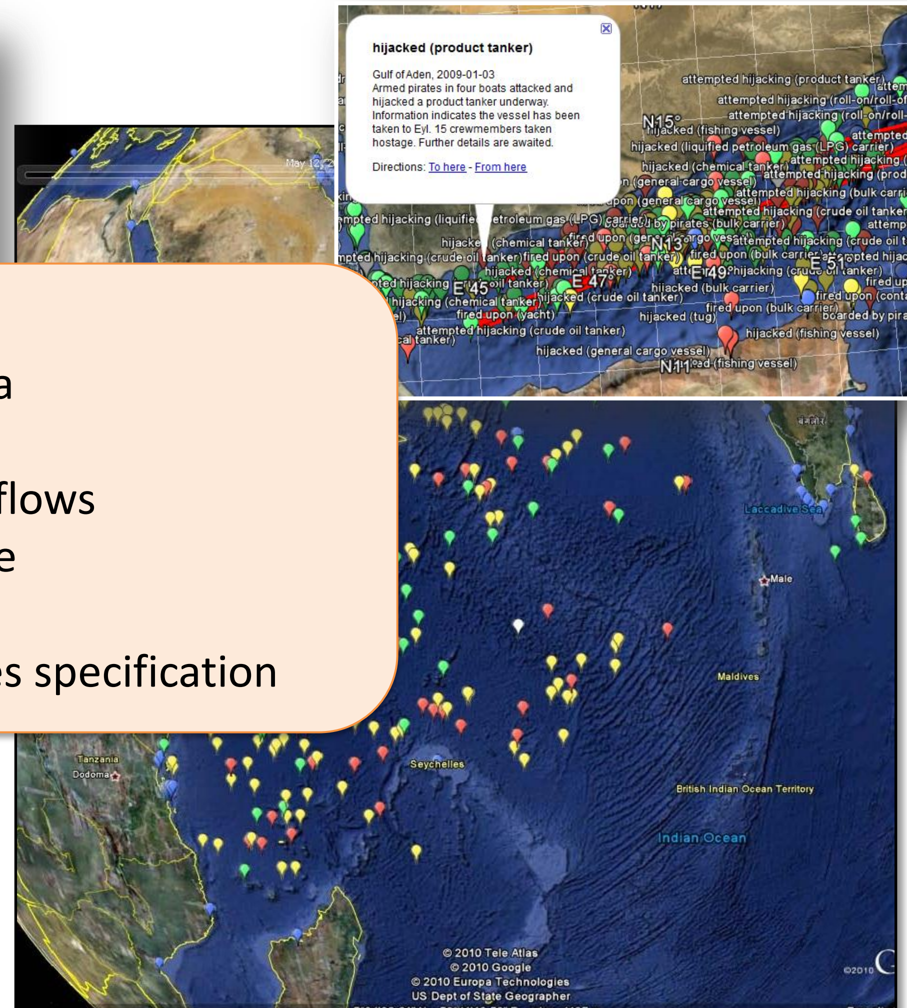
Data Examples (AgentC)

Categories of data

- geographical data
- weather data
- merchant traffic flows
- pirate intelligence
- navy capabilities
- counter-measures specification



Global AIS tracks
(2-day sample 28-29 Jan 2010)



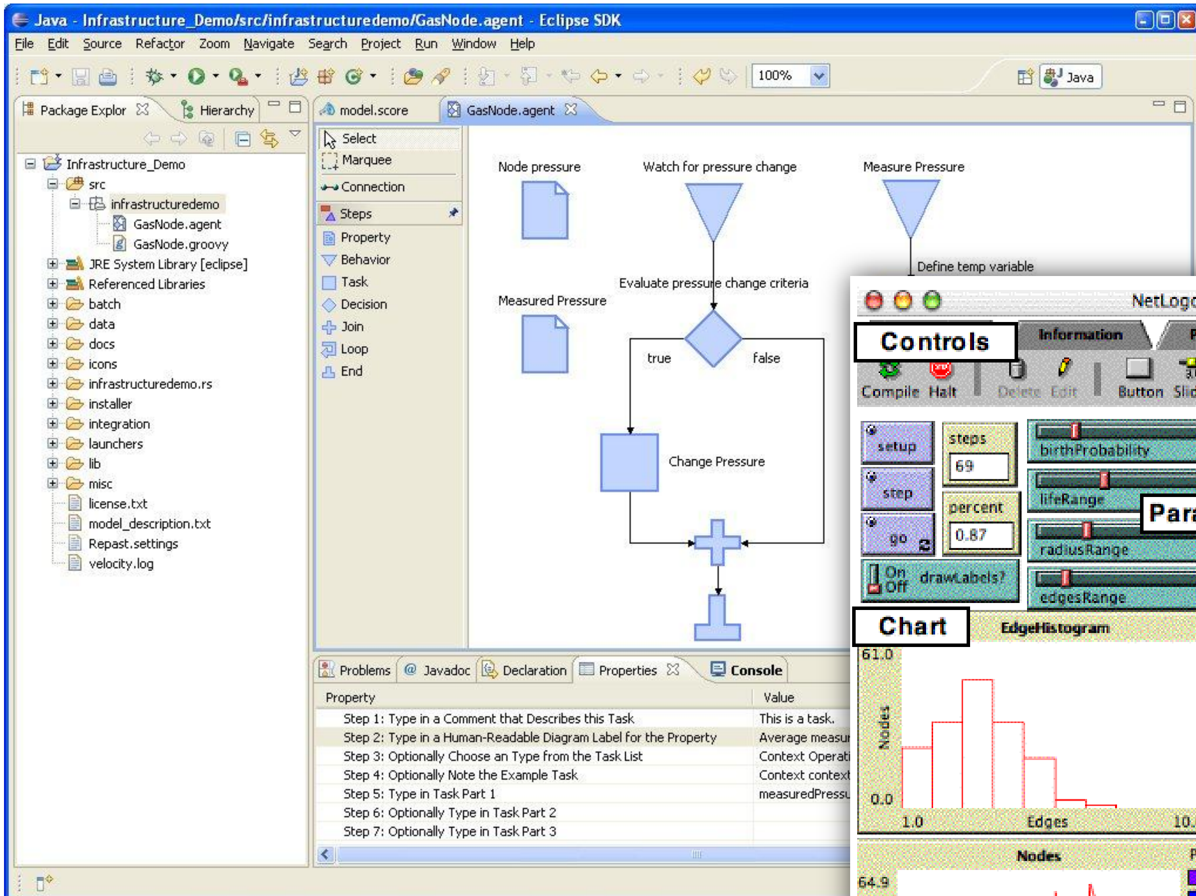
Pirate incidents (2005-2010)

Model Implementation

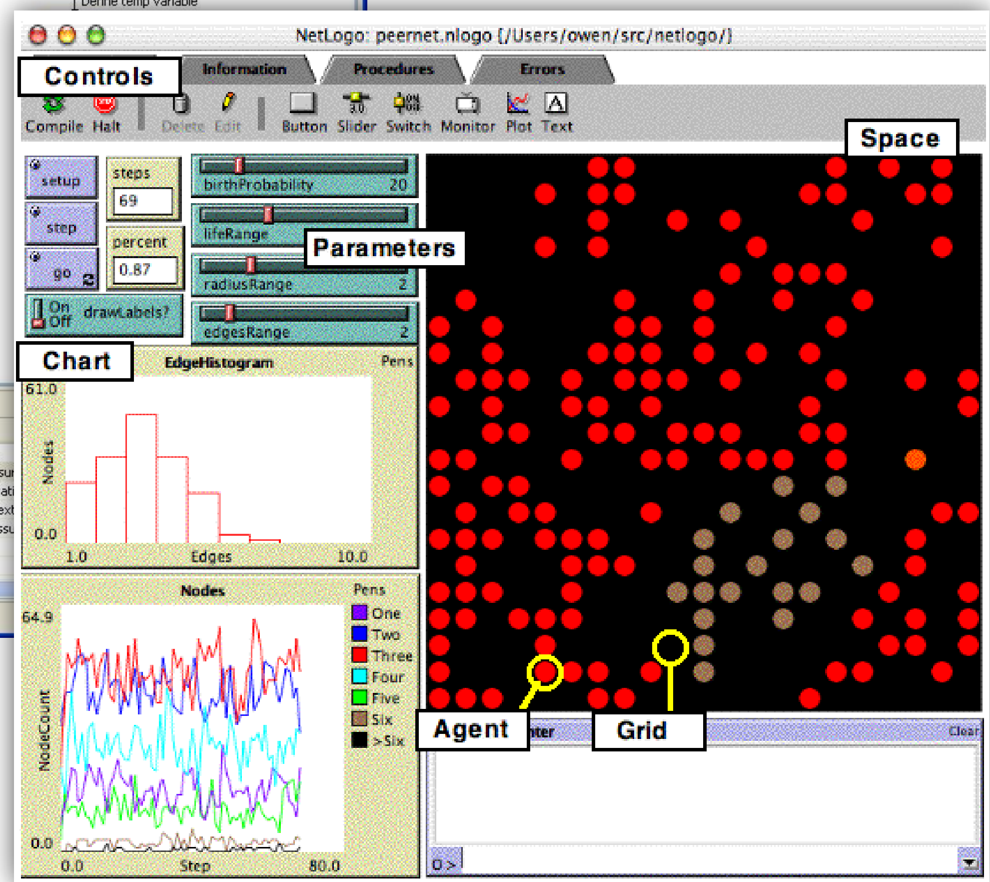
- Implementation of design artifacts into executable code
- General programming languages (Java, C++) or special-purpose
- Import filters implementation
- Reporting scripts

Platforms and Tools

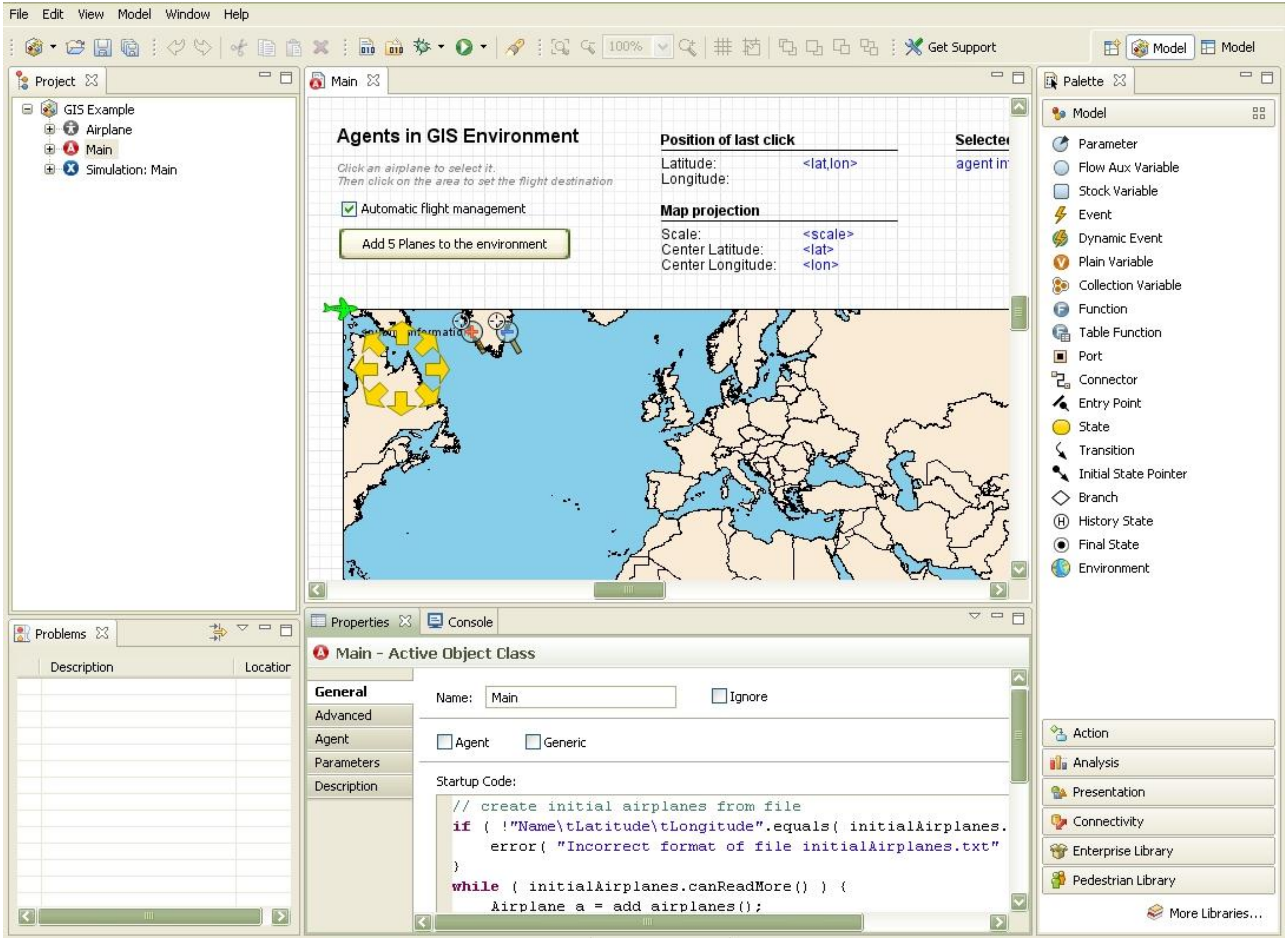
- General platforms still only in an early stage
  - academic/open-source: RePast, NetLogo, AScape
  - commercial: AnyLogic
  - Alite (including the support for distributed simulation)
- Special-purpose platforms more mature
  - traffic modeling: AgentPolis, AIMSUN, Quadstone Paramics
  - pedestrian modeling: LEGION, Pedestrian Simulation
- GIS tools and data sources
  - Google Earth, NASA WorldWind
  - http://www.openstreetmap.org/



Repast Simphony

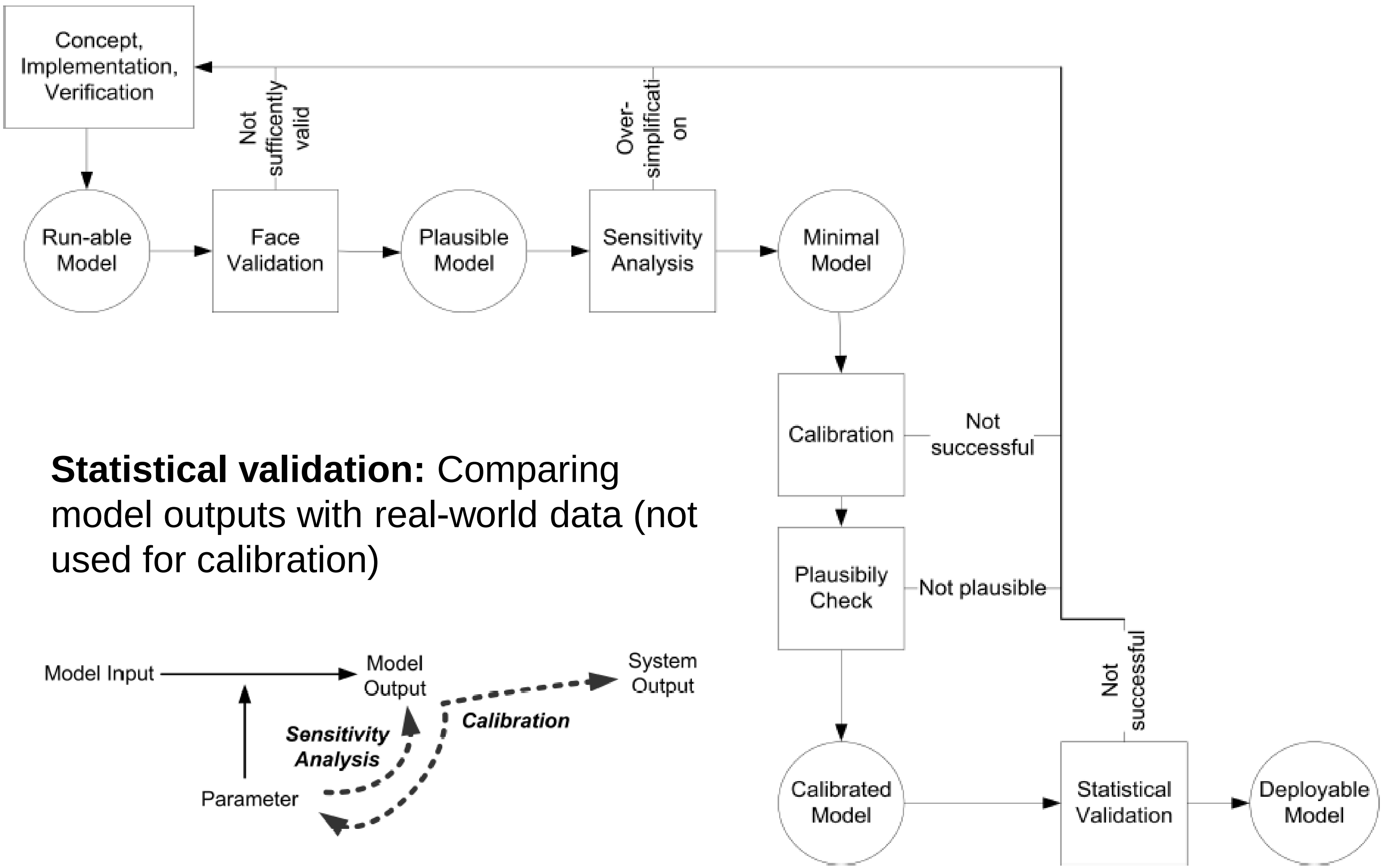


NetLogo


File Edit View Model Window Help
Get Support
Model
Model
Project
GIS Example
Airplane
Main
Simulation: Main
Main
Agents in GIS Environment
Click an airplane to select it.
Then click on the area to set the flight destination
Automatic flight management
Add 5 Planes to the environment
Position of last click
Latitude:
Longitude:
<lat,lon>
Map projection
Scale:
Center Latitude:
Center Longitude:
<scale>
<lat>
<lon>
Palette
Model
Parameter
Flow Aux Variable
Stock Variable
Event
Dynamic Event
Plain Variable
Collection Variable
Function
Table Function
Port
Connector
Entry Point
State
Transition
Initial State Pointer
Branch
History State
Final State
Environment
Action
Analysis
Presentation
Connectivity
Enterprise Library
Pedestrian Library
More Libraries...
Problems
Description
Properties
Console
Main - Active Object Class
General
Advanced
Agent
Parameters
Description
Name:
Main
Ignore
Agent
Generic
Startup Code:
// create initial airplanes from file
if (!"Name\tLatitude\tLongitude".equals(initialAirplanes.
error("Incorrect format of file initialAirplanes.txt"
}
while (initialAirplanes.canReadMore()) {
Airplane a = add airplanes();

AnyLogic

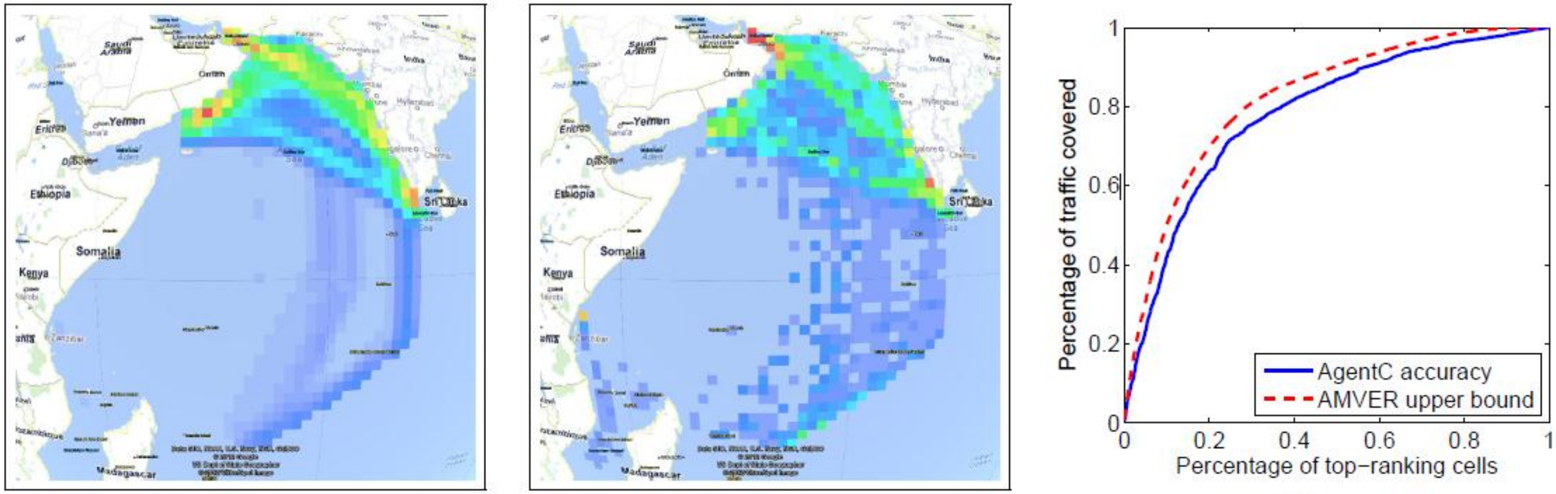
Calibration and Validation



Statistical validation: Comparing model outputs with real-world data (not used for calibration)

AgentC Calibration Example

Parameter	Attack Distribution	Attack Frequency	Hijack Ratio
#N	**0.15**	0.24	0.32
#P	0.046	**0.74**	0.041
P Visibility radius	0.052	**0.26**	0.11
M Alertness	0.053	0.075	**0.20**
P Hijack prob. ρ_a, ρ_u	0.057	0.078	**0.16**
P Navy knowledge	**0.1**	0.085	0.14

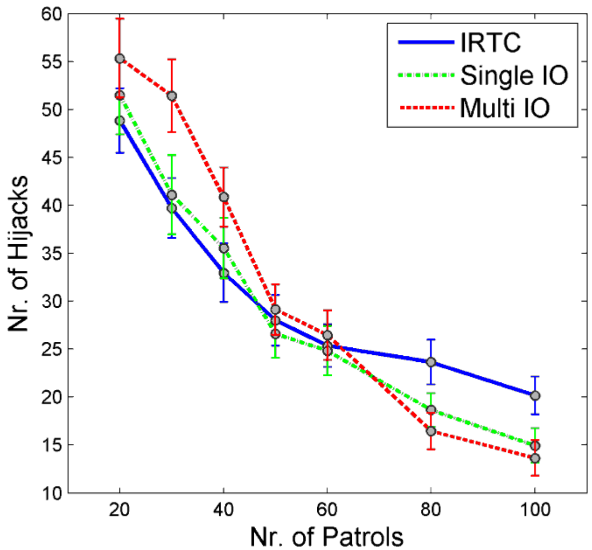


(a) (b) (c)

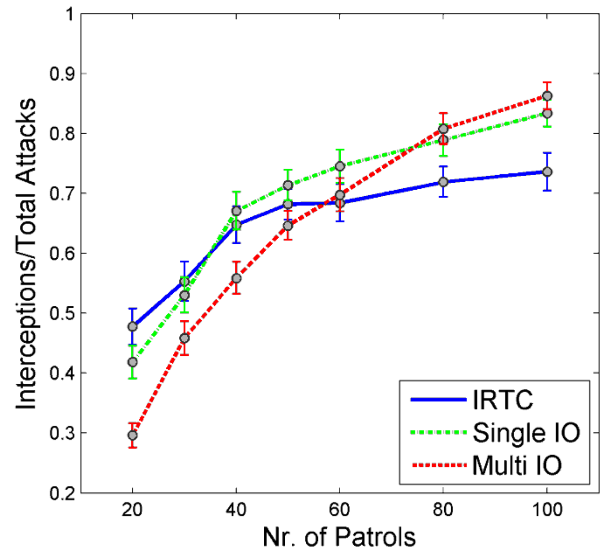
Fig. 8. Merchant traffic sub-model calibration. (a) Density map for merchant traffic sub-model. (b) Reference AMVER 2011 traffic density map. (c) SR curves for the merchant traffic sub-model (blue) and the AMVER density map (red). The red SR curve of the AMVER model captures the theoretical upper-bound achievable for a given spatial resolution of the model: 20% of the AMVER top ranking cells cover 70% of the AMVER traffic; 20% of the AGENTC merchant traffic sub-model top ranking cells cover approximately 64% of the AMVER traffic.

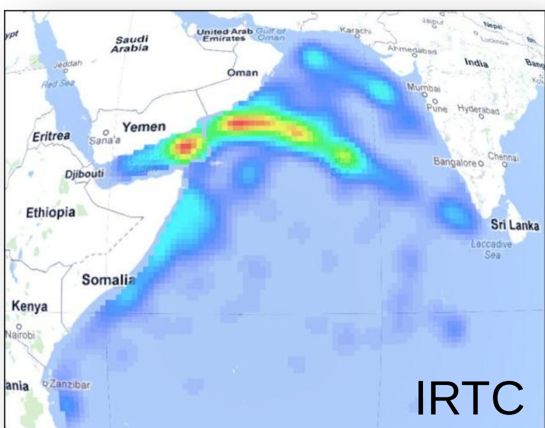
Results Evaluation Example (AgentC)

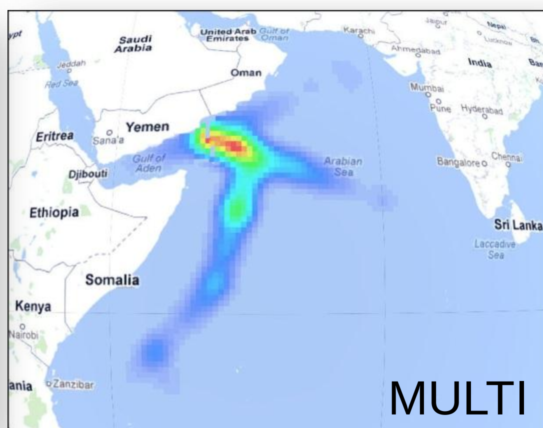
Nr. of hijacks



Patrol Efficiency







Corridor layout	Transit distance	Transit duration
NONE	2153 nm	141 H
SINGLE	2162 nm	142 h
MULTI	2213 km	145 h

Under 60 deployed patrols, randomized transit is more secure. Over 60 patrols, corridor extensions provide better protection and boost patrol efficiency.

Discussion

Advantages of ABM

- Higher **expressivity / modeling power**
  - some behaviors cannot be expressed using equations
- **Natural description** with direct **correspondence**
- Easier **deployment** / translation back to practice
- Ability to capture **adaptivity, emergence** and **heterogeneity**
- Additional level of **validation**
  - individual level in addition to global
- Facilitates integration of **multiple models**

> ABMs give **more realistic results** than EBMs for **manageable levels** of representational detail

Barriers and Enablers

- High **computational cost** ← *cloud deployment*
- Large amounts of **calibration data** required ← *instrumentation*
- Lack of industry-strength **platforms and tools** ← *further R&D*
- (Paradigm shift)

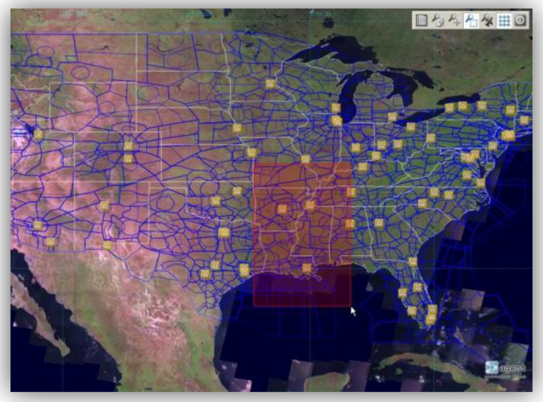
When to Use ABMS

- Agents exhibit complex behavior, including learning and adaptation,
- Agent’s behavior has non-smooth/discrete dynamics with thresholds, if-then rules etc.
- Interactions between agents are context-dependent, nonlinear, discontinuous, or discrete; network-effects apply
- Topology of the interactions is heterogeneous, complex and dynamic
- Population of agents is heterogeneous
- Space is crucial and the agents' positions are not fixed
- System-level equation are not known

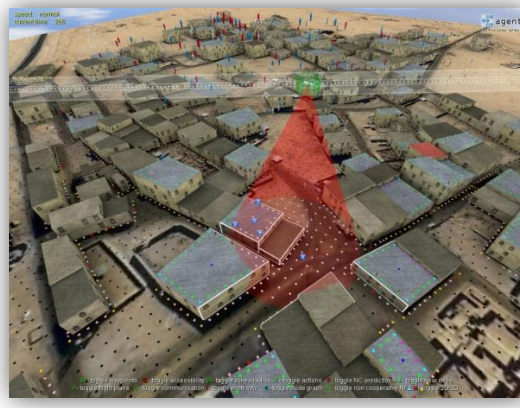
Application Areas

- Infrastructures
  - traffic and transport: development of traffic networks, understanding and eliminating congestion , increasing safety
  - electricity markets
- Crowds
  - pedestrian modeling
  - capacity optimization, evacuation procedures
- Organizations
  - organization design optimization, operation risk estimation
- Markets and economies
  - supply chains and logistics
- Computer networks
  - bandwidth usage estimation, worm infection modeling
- Security
  - crime modeling, vulnerability estimation

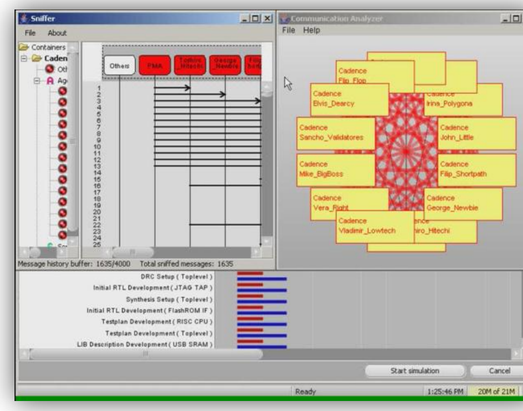
Simulations in ATG



Air traffic

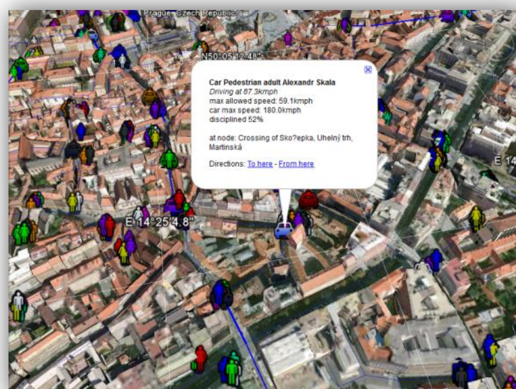


Unmanned aerial vehicles



Business processes

Maritime traffic



Urban life

Highway

Conclusions

- Most recent addition to modeling and simulation toolbox
- **Bottom-up** approach (micro to macro)
- Most suitable for **complex systems** composed of autonomous, interacting entities
- Allows **high-fidelity** models at the expense of **high-computational costs**
- Mature tools exist for specific domains (e.g. transport, crowds); General purpose platforms and tools still under development

OPPA European Social Fund
Prague & EU: We invest in your future.