Bridging the Gap Between Discrete-Event and Agent-Based Simulation

Luis Gustavo Nardin
National College of Ireland
LuisGustavo.Nardin@ncirl.ie

II Workshop em Modelagem e Simulação de Sistemas Intensivos em Software

19th October 2020

Overview

- Introduction
- Agent-Based Simulation
- Discrete-Event Simulation Paradigms
- Agent-Based Simulation using DES Paradigm
- Adding Agent Concepts to DES Paradigm
- Conclusions and Outlooks

Introduction

- There are different interpretations of **Agent-Based Systems**
  - Disciplines: AI, Robotics, Complexity Science, Economics, Social Science

- These differences derive from the diverse understanding each discipline has of what constitutes an agent

Introduction

“An agent is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuator**.”

(Russel & Norvig, 2010)

Introduction

“An agent is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuator**.”

(Russel & Norvig, 2010)

“an agent is a computer system capable of **autonomous** action in some **environment**, in order to achieve its **delegated goals**.” (Wooldrige, 2009)

Introduction

“An agent is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuator**.”

(Russel & Norvig, 2010)

“an agent is a computer system capable of **autonomous** action in some **environment**, in order to achieve its **delegated goals**.” (Wooldrige, 2009)

“An **autonomous agent** is a **system** situated within and a part of an **environment** that **senses** that environment and **acts** on it, over time, in pursuit of its **own agenda** and so as to effect what it senses in the future.”

(Franklin & Graesser, 1997)

Introduction

- No universal agreement on the concept of Agent
- Basic features of an Agent
  - Autonomous
  - Sense
  - Act
  - Environment
  - Goal
  - Plan

Belief

Introduction

- Broad types of agents
  - **Weak**

    Agents are individual entities that **interact** with their **environment** and with **each other**
  - **Strong**

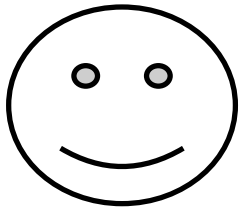
    Agents are individual entities that in addition to **interacting** with their **environment** and with **each other**, have their cognitive states and operations modeled explicitly enabling adaptation

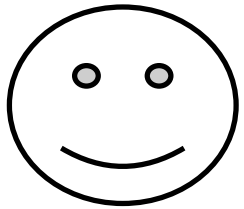
Agent-Based Simulation

- Modeling and simulation are not immune to these differences in the Agent terminology
- The term **agent-based simulation** is used ambiguously
  - **Individual-based simulation** (structure and interaction of individual entities)
    - i.e., weak agent type
  - **Cognitive agent simulation** (+ state and cognitive operations of an agent)
    - i.e., strong agent type

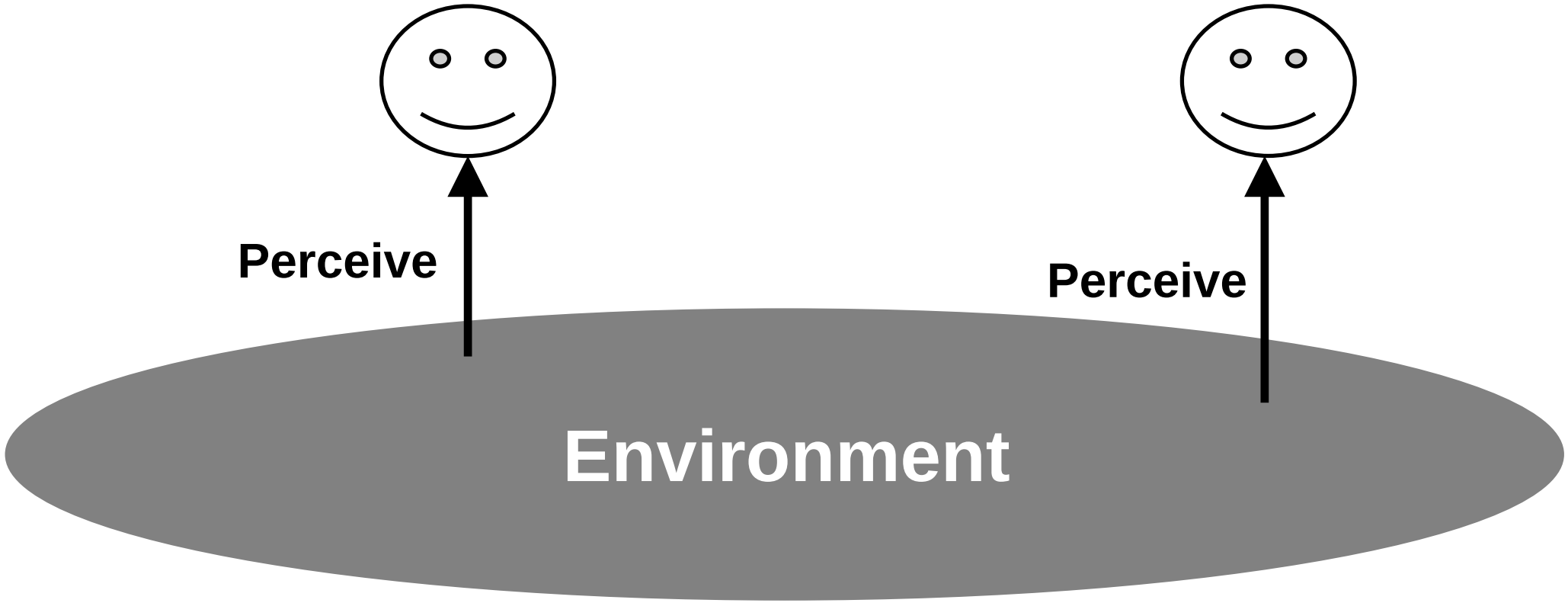
Agent-Based Simulation

Agent-Based Simulation



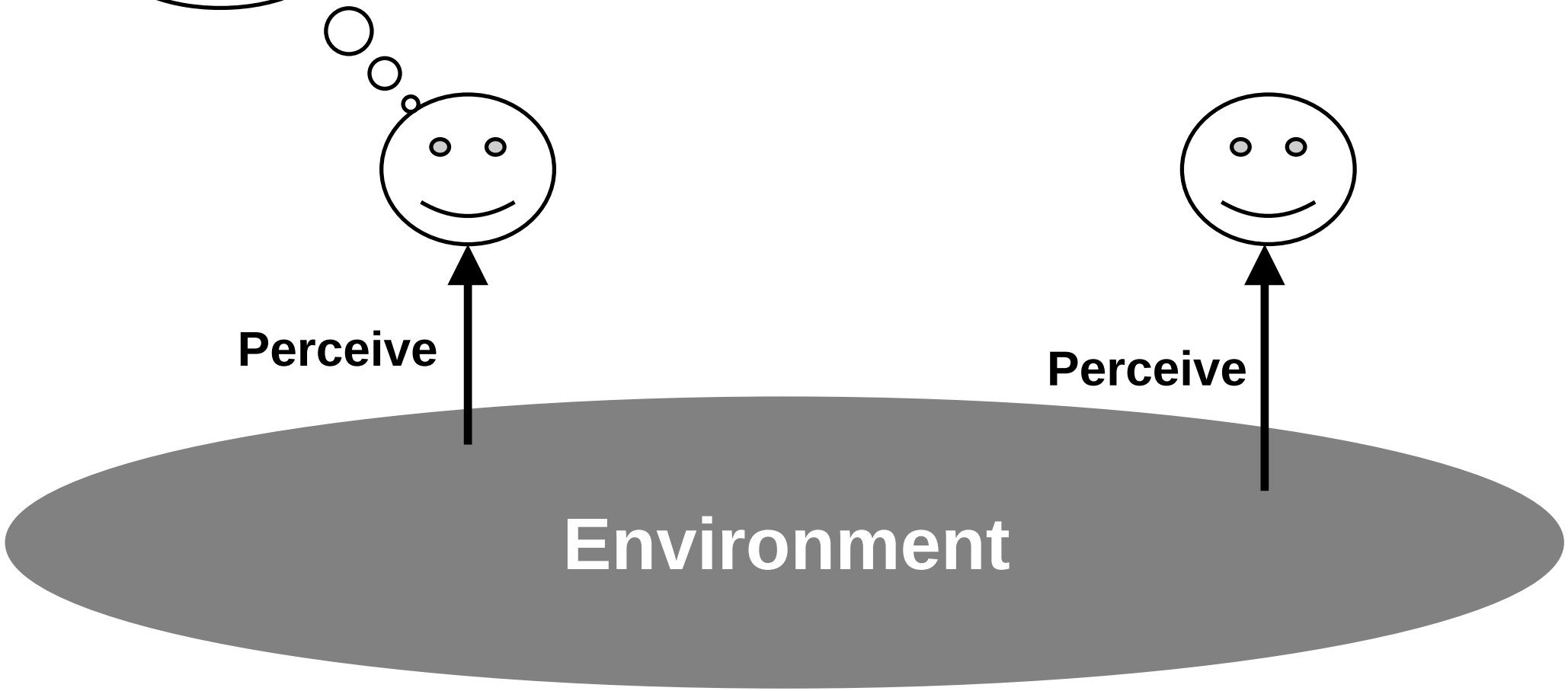


Agent-Based Simulation

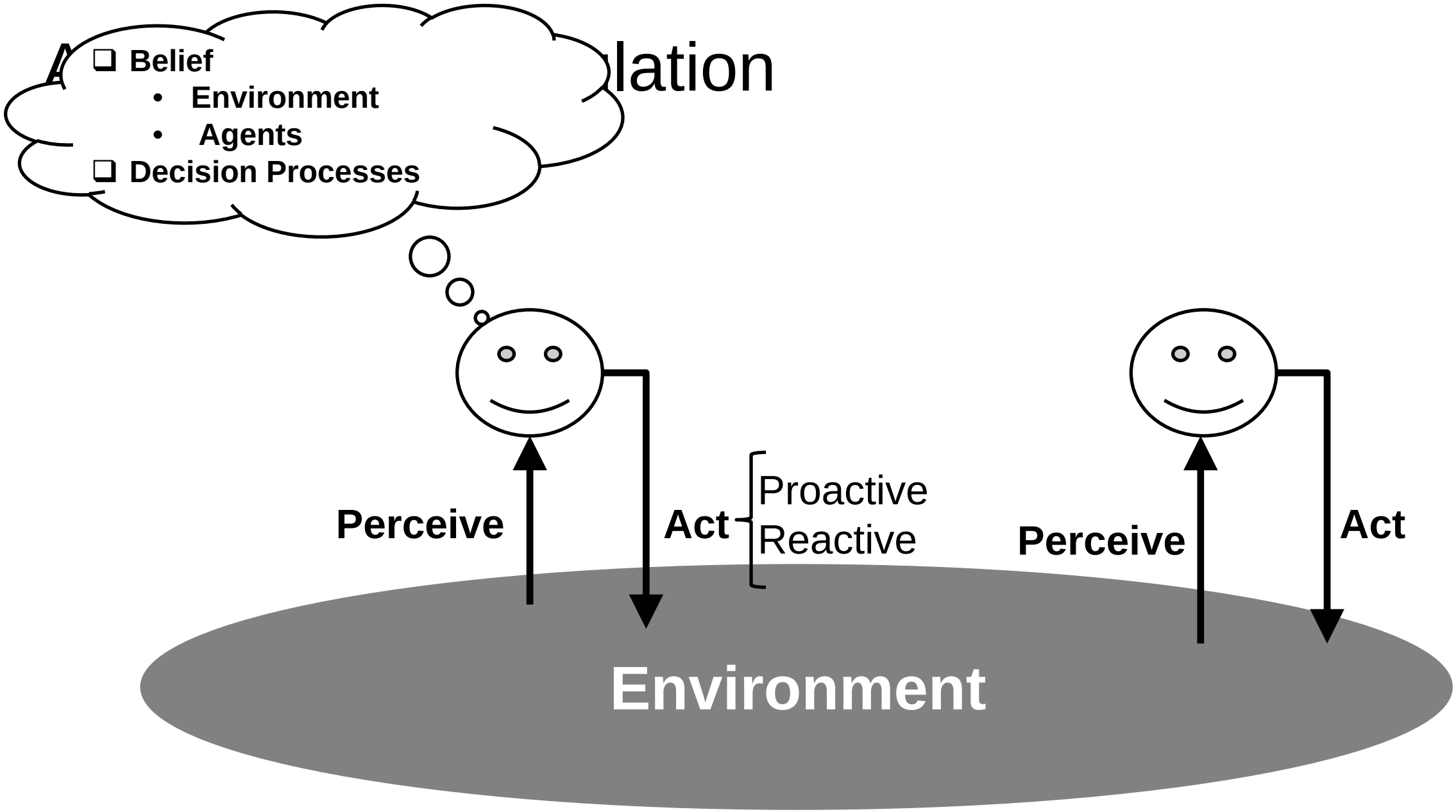


A lation

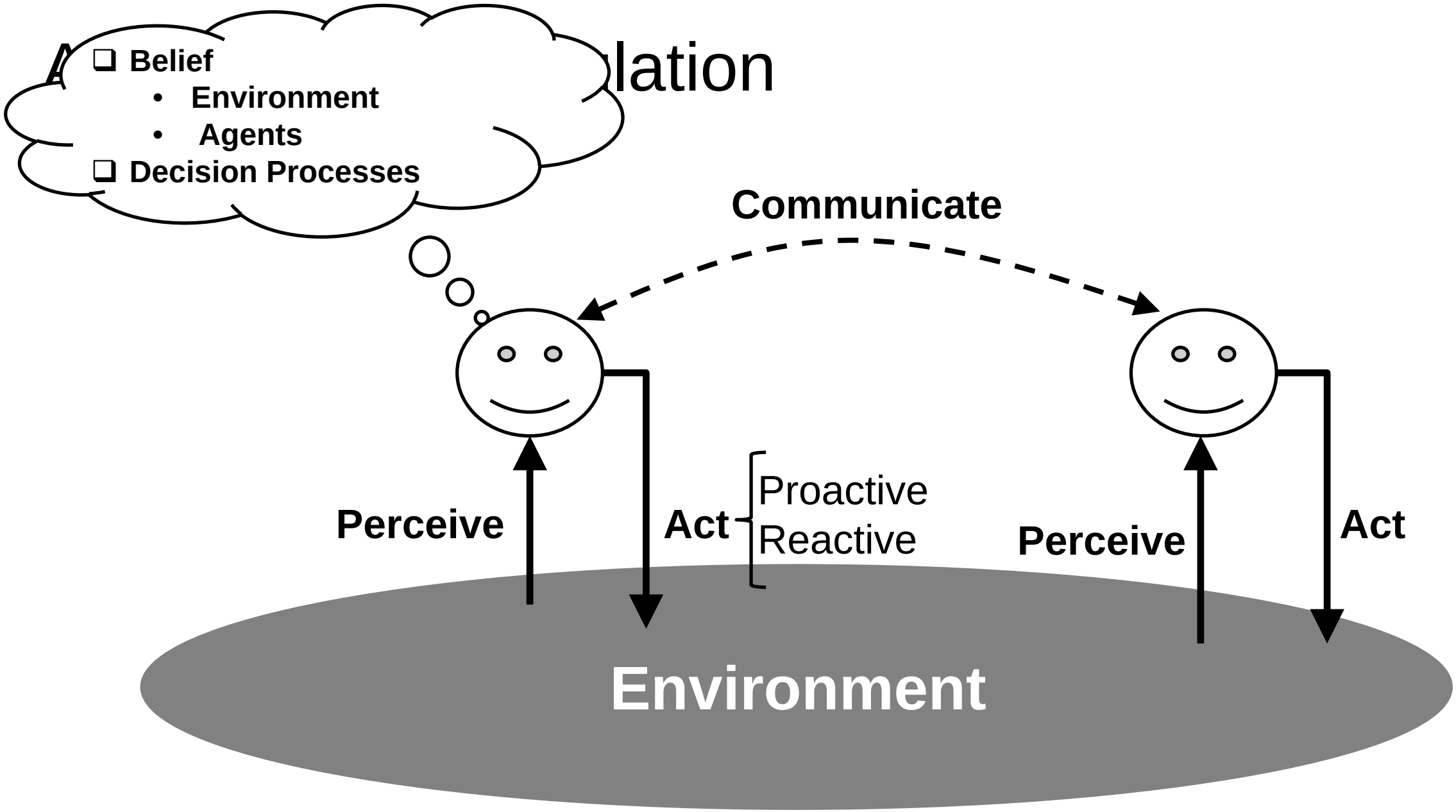
- ❑ Belief
  - Environment
  - Agents
- ❑ Decision Processes



A lation


❑ Belief
• Environment
• Agents
❑ Decision Processes
Perceive
Act
Proactive
Reactive
Perceive
Act
Environment

A lation


Belief
Environment
Agents
Decision Processes
Communicate
Perceive
Act
Proactive
Reactive
Perceive
Act
Environment

Agent-Based Simulation

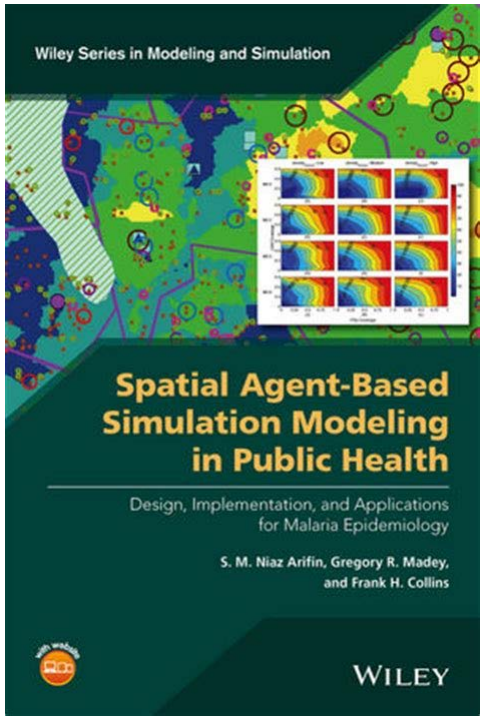
Characteristics

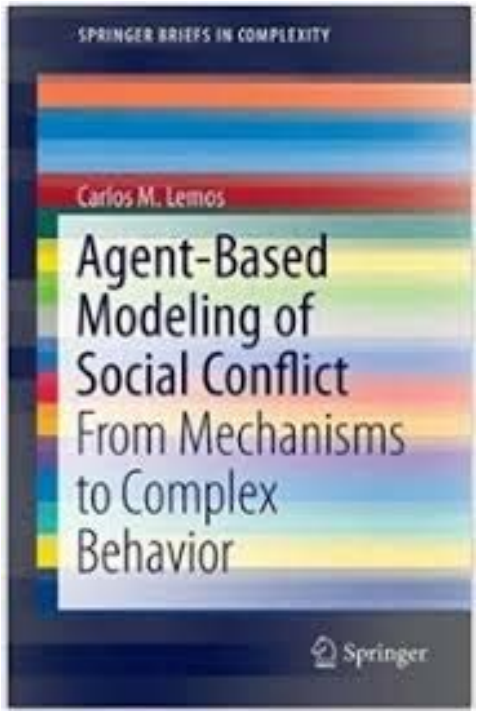
- Complex Systems
- Emergent Behavior
- Dynamic
- Interactive

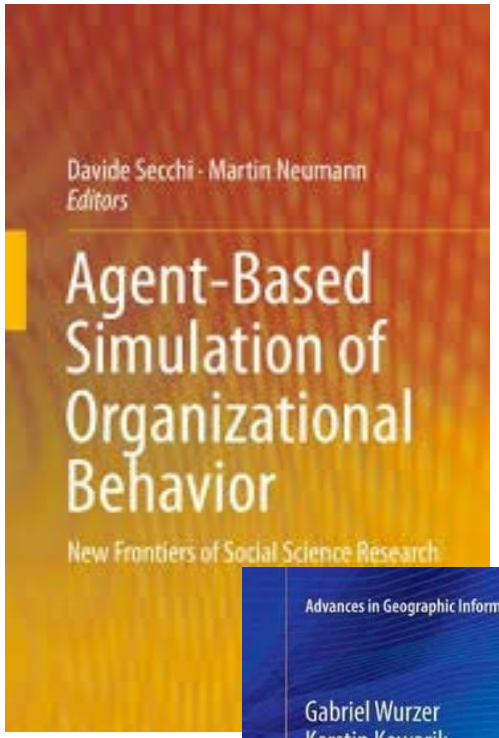
Benefits

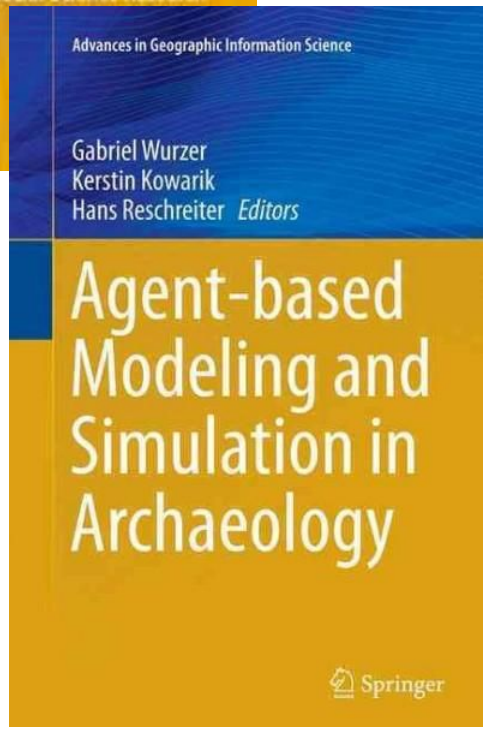
- Isolating prime mechanics
- Interaction of micro & macro
- What if? scenarios
- Finding equilibria

Agent-Based Simulation









Agent-Based Simulation and Discrete-Event Simulation

Because agent **operations** are **discrete**, it is natural to define an agent-based modeling and simulation approach as an extension of a Discrete-Event Simulation (DES) approach

Discrete-Event Simulation

Event-Based Simulation
Event Graphs (Schruben, 1983)

Event Scheduling
Future Event List

SIMSCRIPT (1962)

(Wagner, 2020)

Discrete-Event Simulation

Event-Based Simulation
Event Graphs (Schruben, 1983)

Event Scheduling
Future Event List

SIMSCRIPT (1962)

Process Network Simulation

Activities

GPSS (1961)
Arena
Simio
AnyLogic

(Wagner, 2020)

Discrete-Event Simulation

Event-Based Simulation
Event Graphs (Schruben, 1983)

Event Scheduling
Future Event List

SIMSCRIPT (1962)

Process Network Simulation

Activities

GPSS (1961)
Arena
Simio
AnyLogic

Agent-Based Simulation

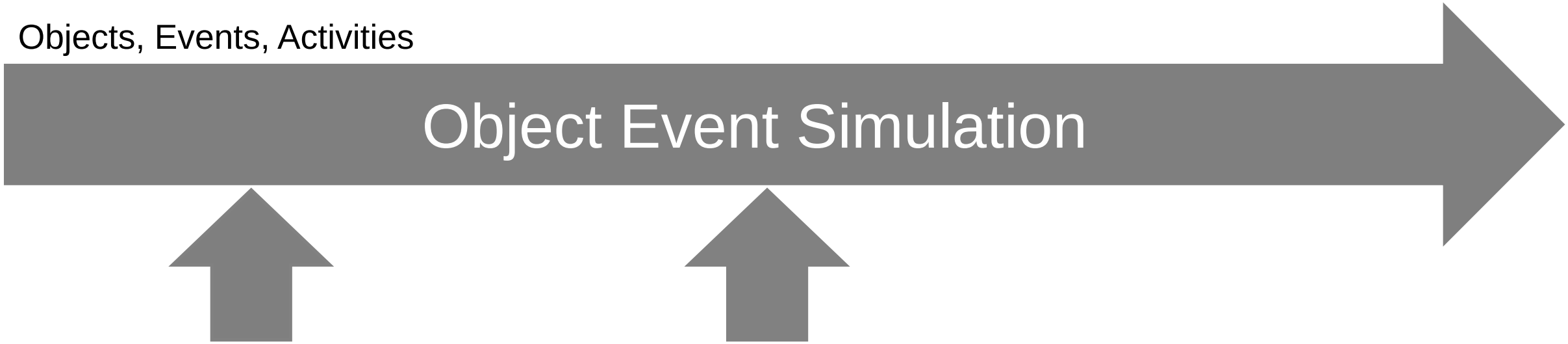
Beliefs
Perception-Action Cycle
Communication

NetLogo
Repast
MASON
Jason

(Wagner, 2020)

Discrete-Event Simulation

Objects, Events, Activities



Event-Based Simulation
Event Graphs (Schruben, 1983)

Event Scheduling
Future Event List

SIMSCRIPT (1962)

Process Network Simulation

Activities

GPSS (1961)
Arena
Simio
AnyLogic

Agent-Based Simulation

Beliefs
Perception-Action Cycle
Communication

NetLogo
Repast
MASON
Jason

(Wagner, 2020)

Object Event Simulation

- Object Event Simulation (OES) represents a general Discrete Event Simulation paradigm based on
  - **object-oriented** state structure
  - **event scheduling** defined by SIMSCRIPT (Markowitz et al., 1962) and **Event Graphs** (Schruben, 1983)
- OES allows the description of DES as a state transition system in terms of
  - **Object types** (e.g., defined in the form of classes of an object-oriented language)
  - **Event types** (e.g., defined in the form of classes of an object-oriented language)
  - **Event rules** (i.e., captures behavioral causal regularities)

(Wagner, 2018)

Causal Regularities

An event e@t causes:

1. **state changes** Δ of affected objects, and
2. **follow-up events** $e_1@t_1$, $e_2@t_2$

According to the dispositions of affected objects, which can be generalized as causal regularities of the form

$$t, O, e@t \rightarrow \Delta, \{e_1@t_1, e_2@t_2,...\} \text{ with } t_i > t$$

with *O* being the set of system's object states at time *t*, such that

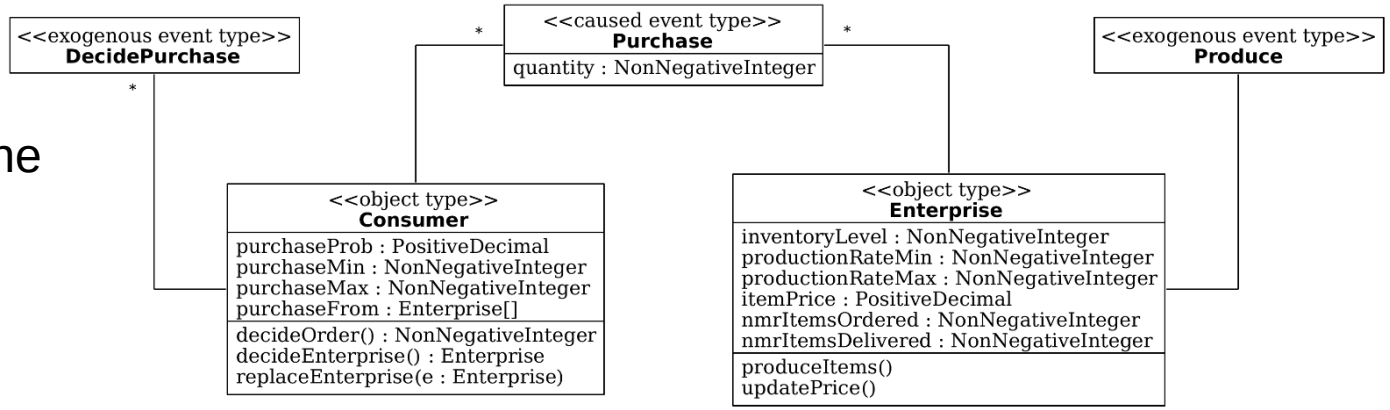
$$O' = \text{Upd}(O, \Delta)$$

is the resulting changed system state.

(Wagner, 2020)

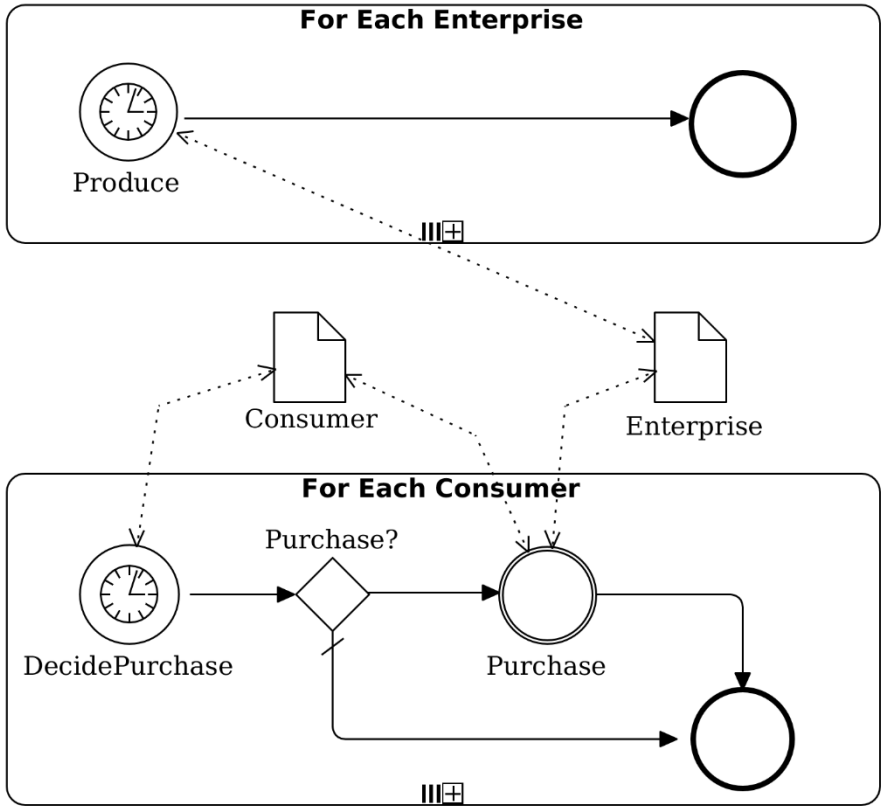
Object Event Simulation

- OES is formed by
  - **Information Model**
    - Describes the state structure of the **object types** and **event types**
    - Modeled as special categories of classes in **UML Class Diagram**



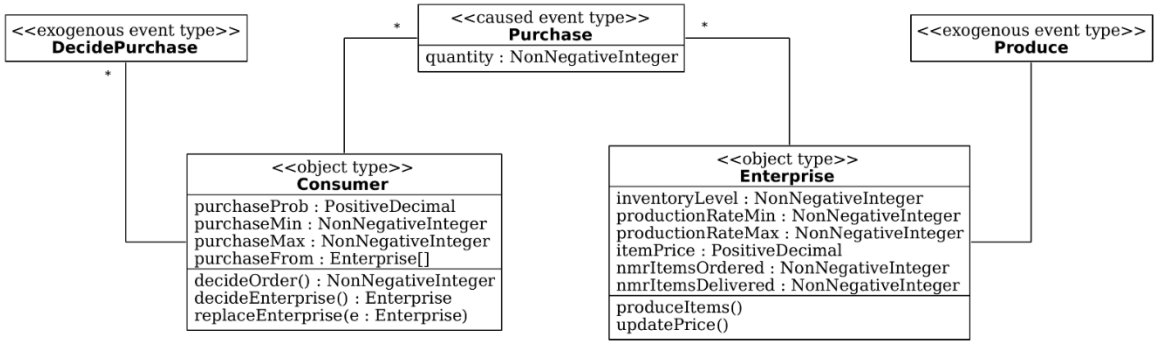
Object Event Simulation

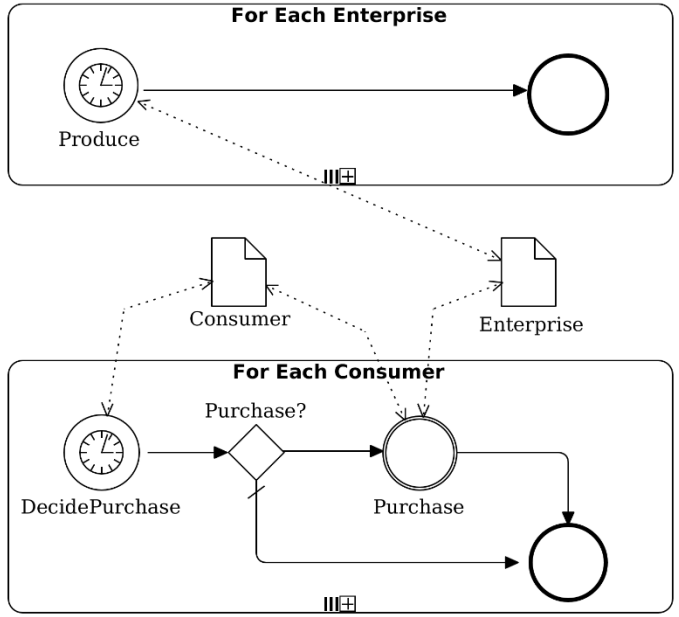
- OES is formed by
  - **Information Model**
    - Describes the state structure of the **object types** and **event types**
    - Modeled as special categories of classes in **UML Class Diagram**
  - **Process Model**
    - Describes the dynamics of the system, i.e., the **event rules**
    - Modeled visually in **BPMN Process Diagram**



Object Event Simulation

- OES is formed by
  - **Information Model**
    - Describes the state structure of the **object types** and **event types**
    - Modeled as special categories of classes in **UML Class Diagram**
  - **Process Model**
    - Describes the dynamics of the system, i.e., the **event rules**
    - Modeled visually in **BPMN Process Diagram**
- A JavaScript-based implementation available at **https://sim4edu.com**





Rebel Groups Protection Racket

https://gnardin.github.io/RebelGroups

Consider an anarchical situation where rebel groups fight over opportunities to extort enterprises.

- Which conditions lead rebel groups to stop fighting, or achieve a stalemate?
- Does the system reach equilibrium based on an initial distribution of strength and enterprise agency?
- Is hegemony achieved? If so, which rebel group(s) achieve it?
- Which conditions may lead to the collapse of the economy?
- What are the fleeing patterns of enterprises?

(Duffy, Klosek, Nardin, & Wagner, 2020)

Objectives

- Investigate the influence of **economic actors** and **external interventions** on the dynamics of civil war

(Duffy, Klosek, Nardin, & Wagner, 2020)

Rebel Groups Protection Racket Model

Object Types

Rebel Group	Rebel Groups are armed groups that maintain dominance over a certain region exploiting and providing services to the population.
Enterprise	Enterprises are economic actors that sell consumption goods, usually powerful actors in the society.

(Duffy, Klosek, Nardin, & Wagner, 2020)

Rebel Group

- **Rational** actors
- **Maximize their utility**: wealth and group size
- Highest priority is to **survive**
- Different decision rules for
  - **cooperation**
  - **inter-group fighting** (offensive, defensive assumptions)

(Duffy, Klosek, Nardin, & Wagner, 2020)

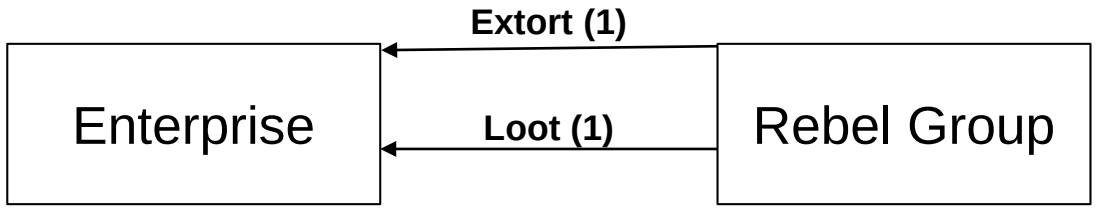
Event Types

Enterprise

Rebel Group

(Duffy, Klosek, Nardin, & Wagner, 2020)

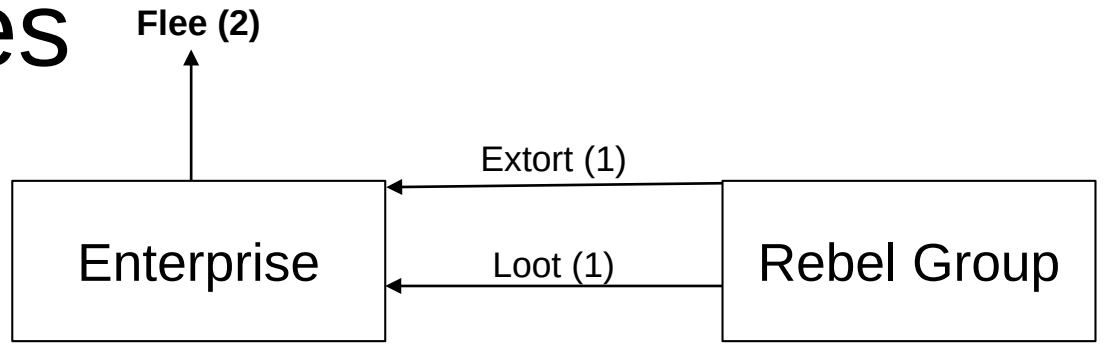
Event Types



1. Rebel Groups extort or loot Enterprises to increase their wealth.

(Duffy, Klosek, Nardin, & Wagner, 2020)

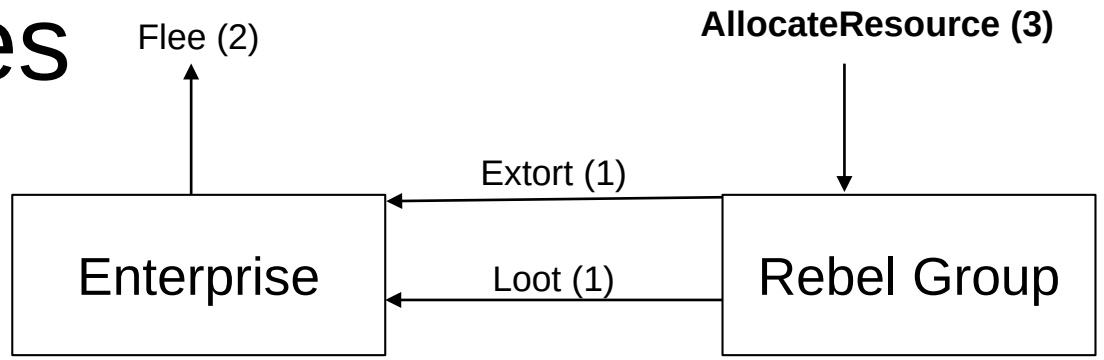
Event Types



1. Rebel Groups extort or loot Enterprises to increase their wealth.
2. Enterprises may flee because of looting.

(Duffy, Klosek, Nardin, & Wagner, 2020)

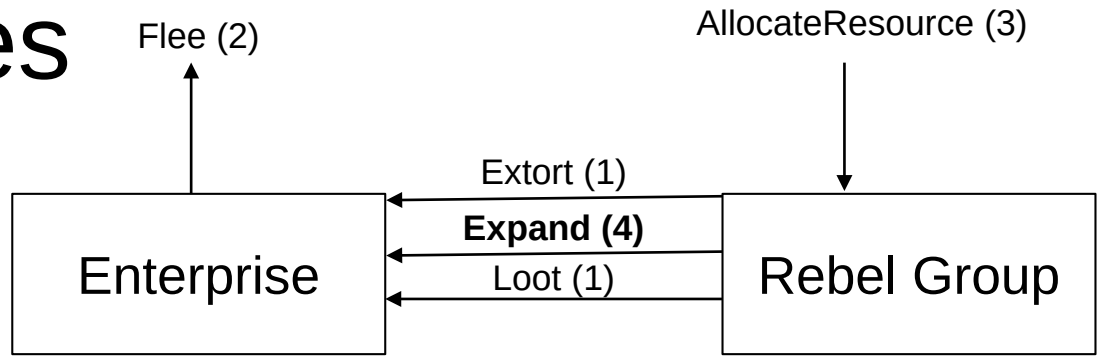
Event Types



1. Rebel Groups extort or loot Enterprises to increase their wealth.
2. Enterprises may flee because of looting.
3. The size and wealth of Rebel Groups influence how they allocate resources for recruiting or expelling members.

(Duffy, Klosek, Nardin, & Wagner, 2020)

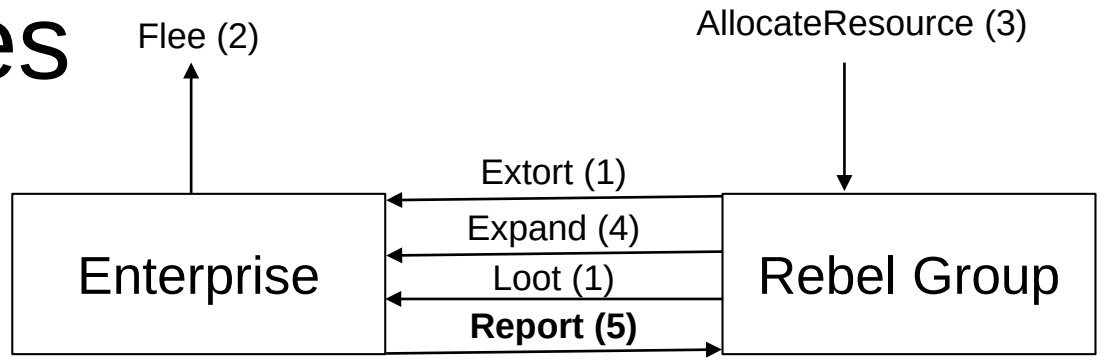
Event Types



1. Rebel Groups extort or loot Enterprises to increase their wealth.
2. Enterprises may flee because of looting.
3. The size and wealth of Rebel Groups influence how they allocate resources for recruiting or expelling members.
4. Rebel Groups decide whether to expand to new territory based on their strength, which is a function of their size relative to the other Rebel Groups.

(Duffy, Klosek, Nardin, & Wagner, 2020)

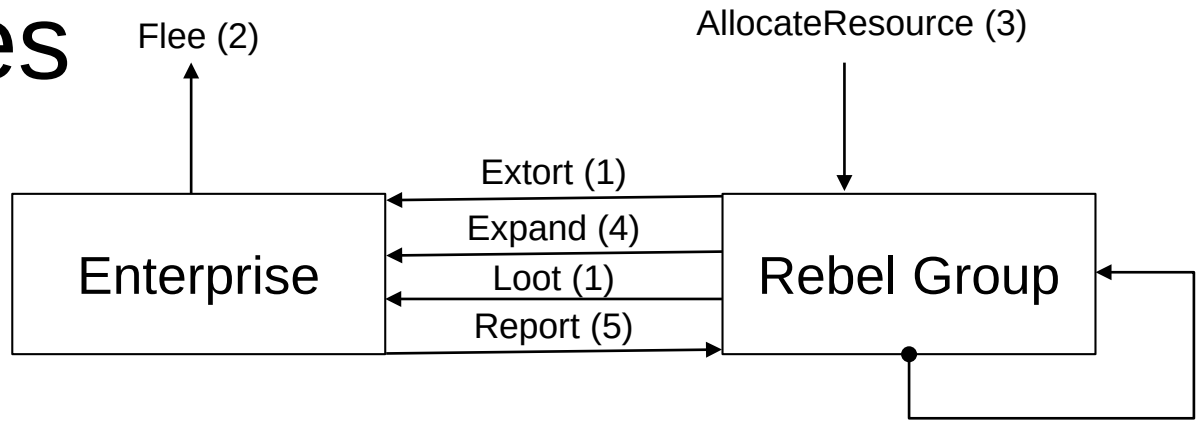
Event Types



1. Rebel Groups extort or loot Enterprises to increase their wealth.
2. Enterprises may flee because of looting.
3. The size and wealth of Rebel Groups influence how they allocate resources for recruiting or expelling members.
4. Rebel Groups decide whether to expand to new territory based on their strength, which is a function of their size relative to the other Rebel Groups.
5. Enterprises may report to their main Rebel Group if another Rebel Group extorts or loots them. The main Rebel Group may initiate a fight against the reported Rebel Group.

(Duffy, Klosek, Nardin, & Wagner, 2020)

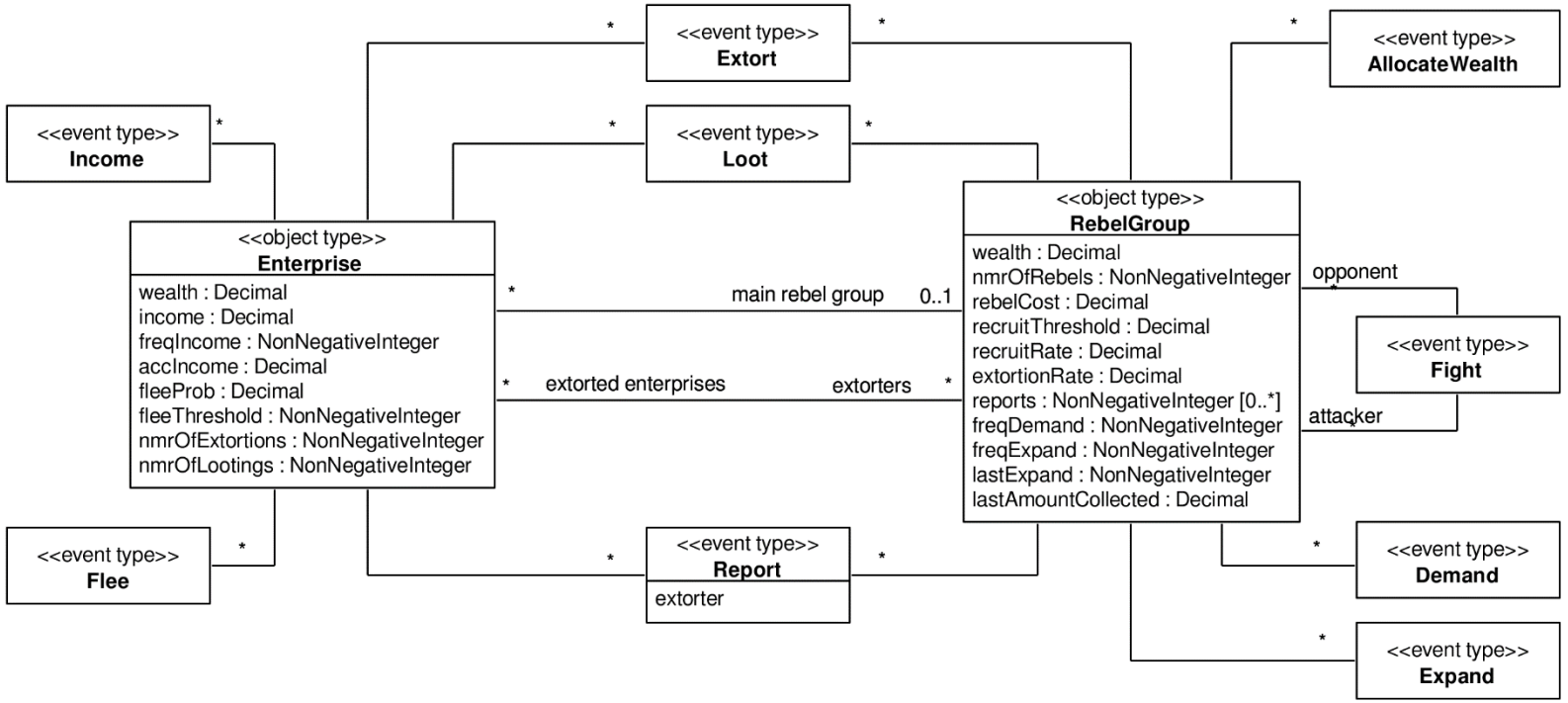
Event Types



1. Rebel Groups extort or loot Enterprises to increase their wealth.
2. Enterprises may flee because of looting.
3. The size and wealth of Rebel Groups influence how they allocate resources for recruiting or expelling members.
4. Rebel Groups decide whether to expand to new territory based on their strength, which is a function of their size relative to the other Rebel Groups.
5. Enterprises may report to their main Rebel Group if another Rebel Group extorts or loots them. The main Rebel Group may initiate a fight against the reported Rebel Group.
6. The size of Rebel Groups increases their probability of winning fights against other Rebel Groups.

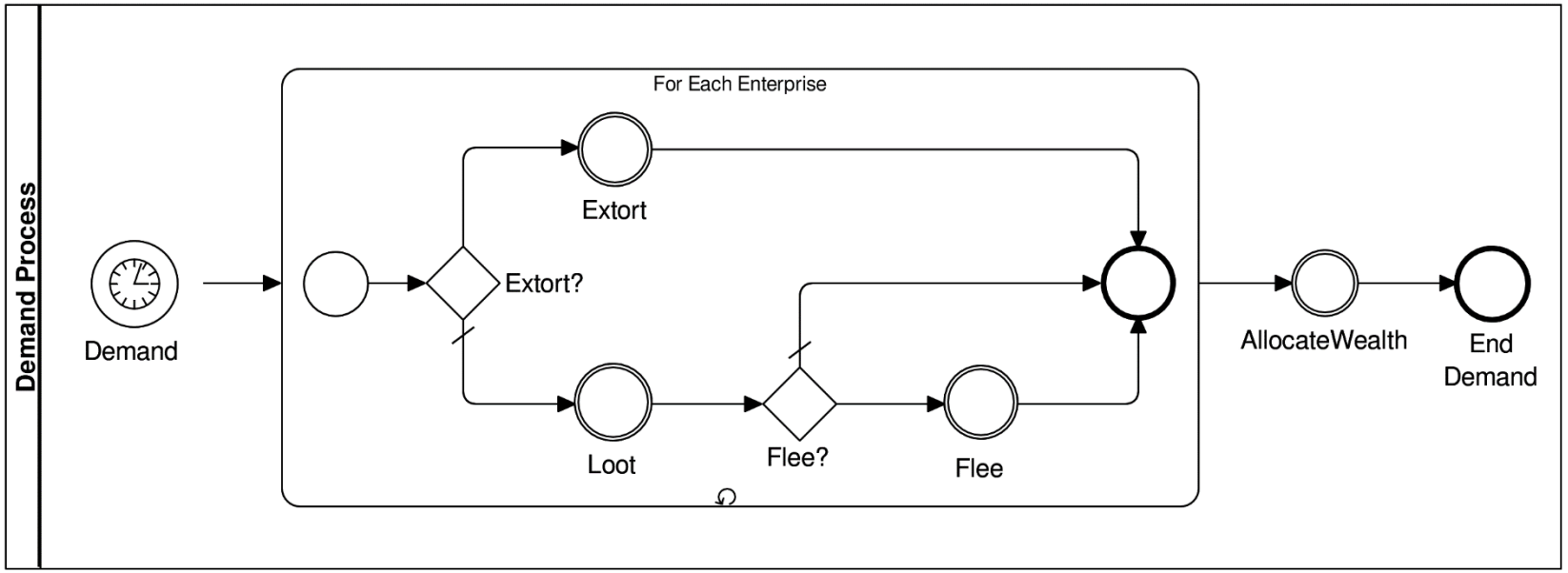
(Duffy, Klosek, Nardin, & Wagner, 2020)

Information Design Model



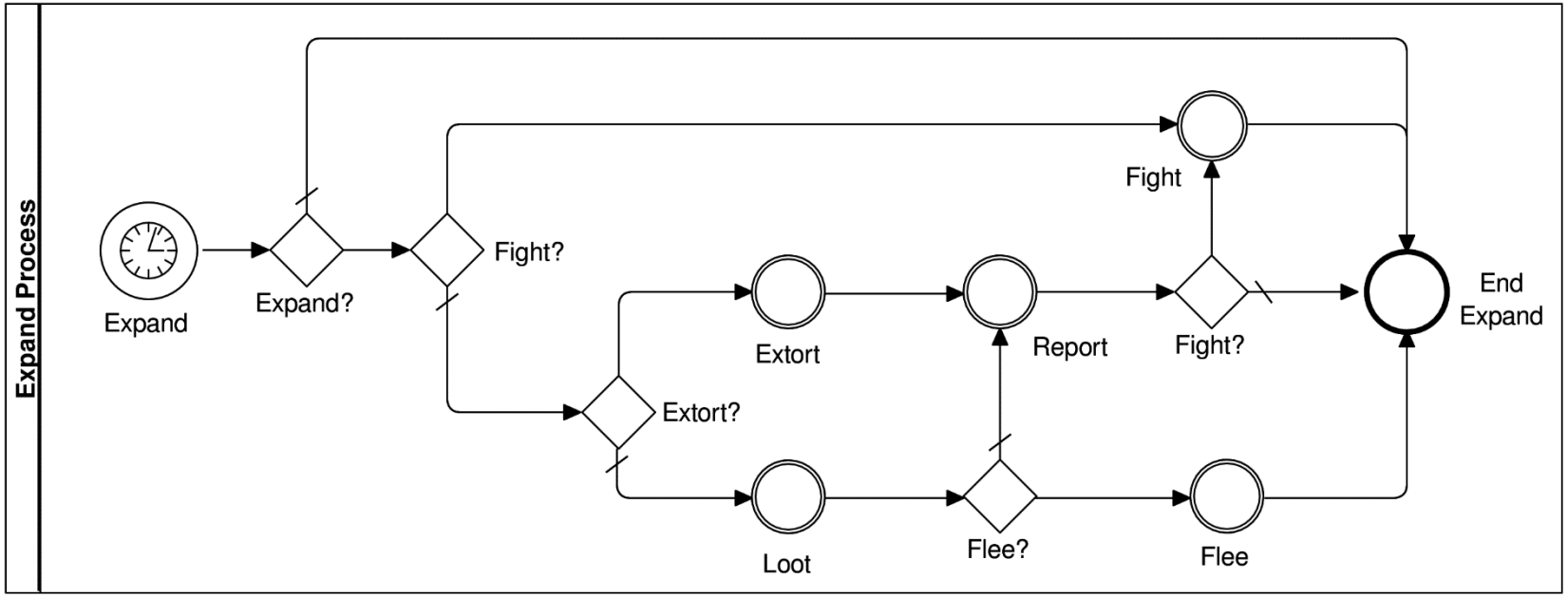
(Duffy, Klosek, Nardin, & Wagner, 2020)

Demand Process



(Duffy, Klosek, Nardin, & Wagner, 2020)

Expand Process



(Duffy, Klosek, Nardin, & Wagner, 2020)

Limitations

- DES formalisms are enough to represent individual-based simulation (i.e., weak agents)
- But lack several features required to properly represent cognitive agent-based simulations
  - Belief System
  - Environment (Perceive-Act Cycle)
  - Communication

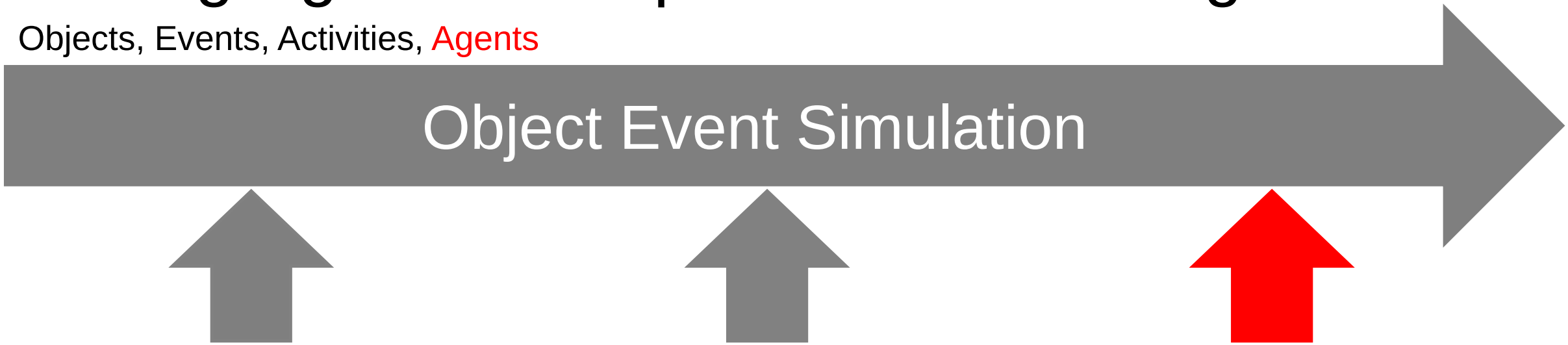
Limitations

Platform	Perceive-Act	Fact/Belief	Belief Type	Communication
NetLogo	Yes	**No**	property-value	-
Repast Symphony	Yes	**No**	-	-
MASON	Yes	**No**	-	-
GAMMA	Yes	**No**	-	FIPA
AnyLogic	Yes	**No**	-	-
Jadex BDI	Yes	**No**	property-object	FIPA
Jason	Yes	Yes	first-order predicate	partial FIPA/KQML
2APL	Yes	Yes	first-order predicate	FIPA

(Wagner & Nardin, 2018)

Adding Agent Concepts to DES Paradigm

Objects, Events, Activities, Agents



Event-Based Simulation
Event Graphs (Schruben 1983)

Event Scheduling
Future Event List

SIMSCRIPT (1962)

Process Network Simulation

Activities

GPSS (1961)
Arena
Simio
AnyLogic

Agent-Based Simulation

Beliefs
Perception-Action Cycle
Communication

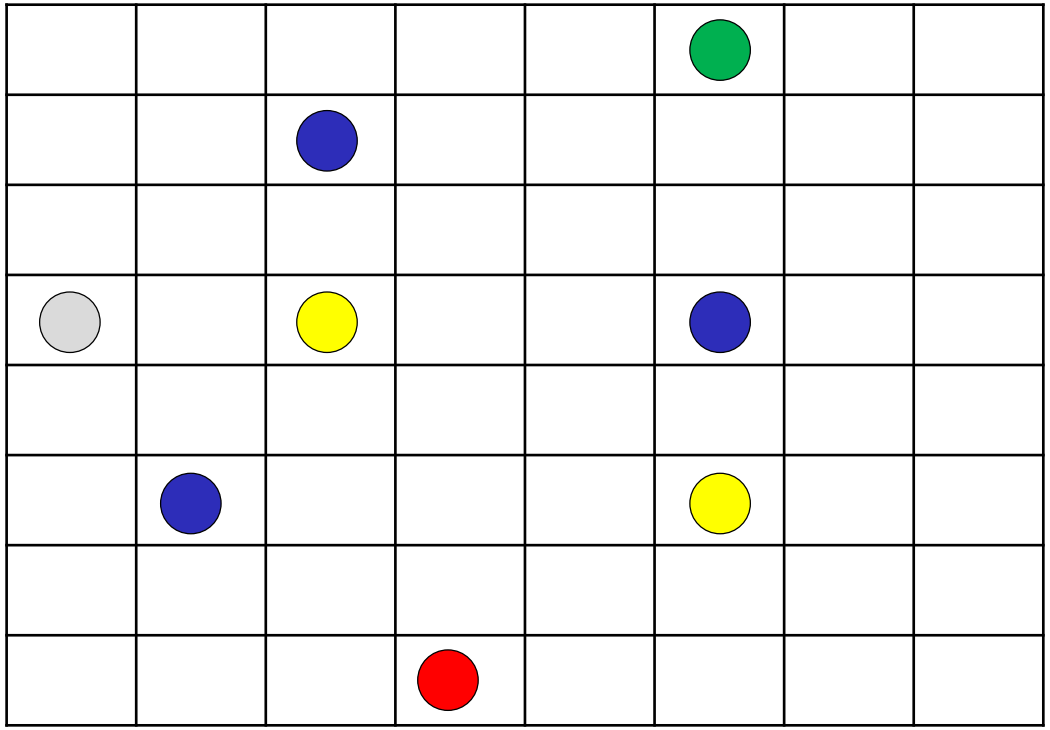
NetLogo
Repast
MASON
Jason

Agent/Object Event Simulation

- Agents are a special type of object
- Agents are represented as Object Types with minimum additional features
  - Belief
  - Perceive
  - Act
  - Communicate

(Wagner & Nardin, 2020)

Scenario



Prince

Knight

Treasure

Castle

Castle with the Princes

- Environment: Island = 2D Grid
- Prince has the goals of
  - Collect treasures
  - Find the castle with princes
- Actions
  - Move on the environment
  - Interact with knights who have information about nearby treasures and the castle with princes
- All objects and agents have a position in the form (x, y)

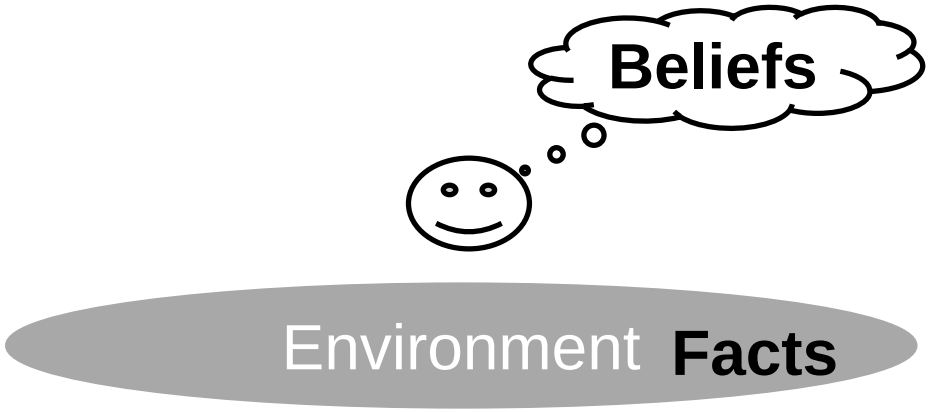
(Wagner & Nardin, 2020)

Beliefs

- There are two kinds of information items
  - **Facts**

    True (**objective**) state of the environment
  - **Beliefs**

    Typically partial and sometimes incorrect (subjective) information of agents and the environment
- Belief Discrepancies
  - Different vocabularies
  - Completeness of beliefs
  - Actual and believed value of a property



(Wagner & Nardin, 2020)

Beliefs

- Beliefs are represented by *object-property-value* triples

Example

1. Fact statement of the wealth of the Prince with ID 17 is 500

   [It's a fact that] 17 wealth 500

2. The Prince's belief statement that his wealth is 650

   [Agent 17 believes that] 17 wealth 650

3. The Prince's belief that the Princess (agent ID 23) is at (x=35, y=42)

   [Agent 17 believes that] 23 x 35 y 42

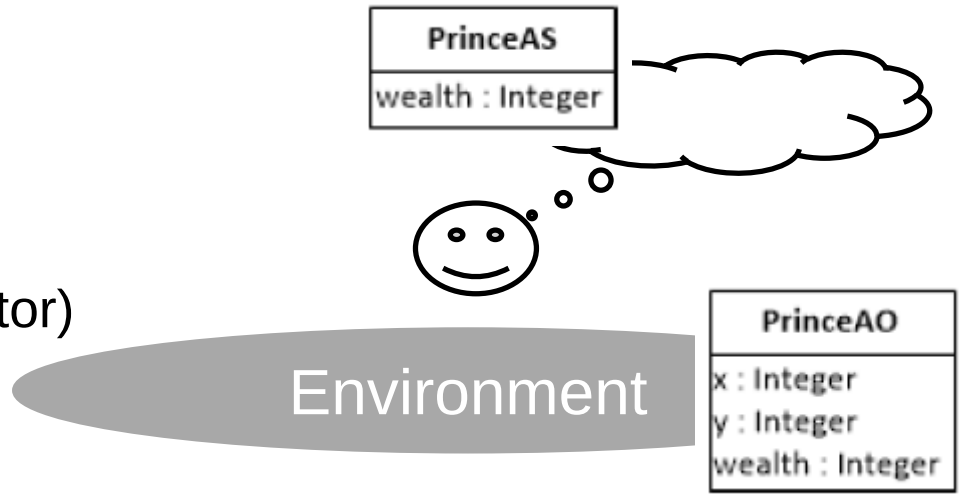
(Wagner & Nardin, 2020)

Beliefs

- **Prince agent**

```
var Prince = new aGENTtYPE({
    name: "Prince",
    supertype: "GridSpaceObject",
    properties: {"wealth": {range: "NonNegativeInteger"}},
    beliefObjectTypes: ["Treasure", "Princess"],
    ...
});
```

- Distinction between facts and beliefs
  - Agent Object (managed by the environment simulator)
  - Agent Subject



(Wagner & Nardin, 2020)

Beliefs

- **Perfect Information Prince agent**

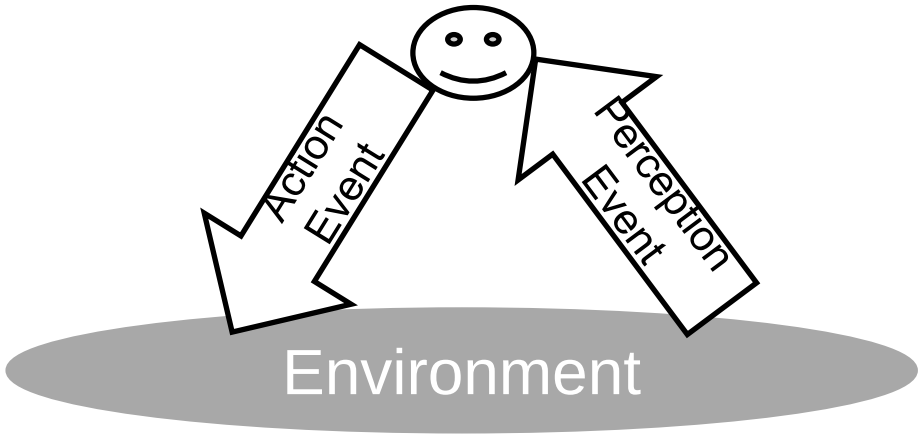
```
var Prince = new aGENTtYPE({
    name: "Prince",
    supertype: "GridSpaceObject",
    properties: {"wealth": {range: "NonNegativeInteger"}},
    hasPerfectInformation: true,
    ...
});
```

- Implies that
  - All objective properties are **duplicated** as self-belief properties
  - All self-belief properties have the **same values** as the corresponding fact properties

(Wagner & Nardin, 2020)

Perception-Action Cycle

- Agents may **perceive** objects and events, and **act** in their environment
- There is two kinds of perceptions
  - Passive perception
    - e.g., robots whose perceptions are automatically created by a sensor
  - Active perception
    - e.g., robots querying the measurement value of a sensor containing a quality detector
- Represented as events
  - Perception Events
  - Action Events



(Wagner & Nardin, 2020)

Perception-Action Cycle

- **Pick-up Treasure action**

```
var PickUpTreasureObject = new aCTIONeVENTtYPE({
    name: "PickUpTreasureObject",
    properties: {"treasureObject": "TreasureObject"},
    onEvent: function (e) {
        e.performer.wealth += e.treasureObject.value;
        sim.objects.remove( e.treasureObject.id);
    }
});
```

- This action event is created and sent to the environment simulator every time that the Prince perceives a treasure.

(Wagner & Nardin, 2020)

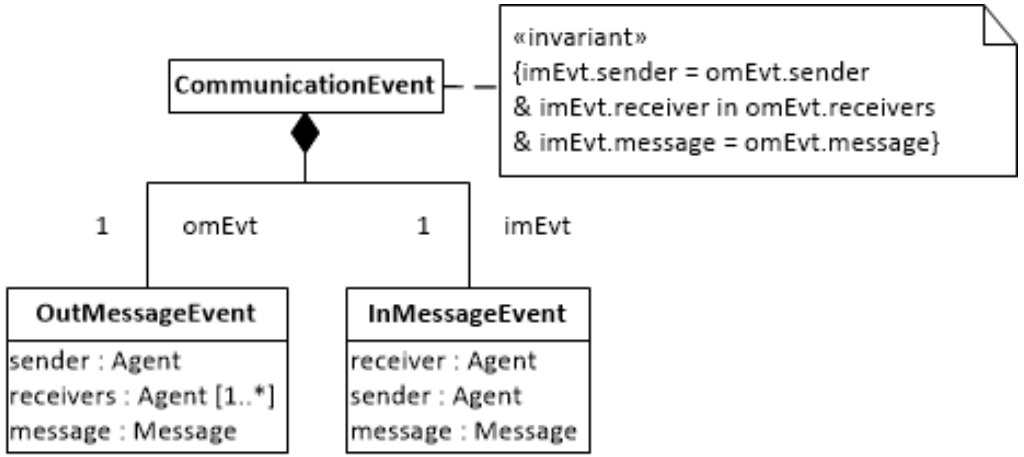
Communication

- The most fundamental type of a conventional message exchange is *tell-ask-reply* communication
- Message types
  - Ad-hoc (i.e., structure and semantics defined for a specific simulation model)
  - Generic (i.e., general built-in messages as Tell/Untell and Ask/Reply)

(Wagner & Nardin, 2020)

Communication

- **Communication event** is a composite event



- **Out-message event** is an external agent event transmitted by the sender's agent to the environment simulator
- **In-message event** is an internal agent event transmitted by the environment simulator to the receiver's agent simulator

(Wagner & Nardin, 2020)

Agent/Object Event Simulation

- The Agent/Object Event Simulation enables the properly representation of features required in cognitive agent-based simulations
- The AOES is still under development and we still do not have any example to demonstrate

Conclusions and Outlooks

- The agent-based simulation approach provide a more flexible form of representing complex systems composed of multiple (semi-)autonomous and interactive entities
- The agent-based simulation can naturally be represented using a Discrete-Event approach because agents' operations are discrete
- DES formalisms have limitations in providing constructs to support all agent-based simulations features
  - Belief
  - Perception-Action Cycle
  - Communication
- DES formalisms can be extended by adding specific agent features

Thank You

Questions?

References

- Duffy F., Klosek K.C., Nardin L.G., Wagner G. (2020). Rebel Group Protection Rackets: Simulating the Effects of Economic Support on Civil War Violence. In: Deutschmann E., Lorenz J., Nardin L., Natalini D., Wilhelm A. (eds) Computational Conflict Research. Computational Social Sciences. Springer, Cham. https://doi.org/10.1007/978-3-030-29333-8_11
- Franklin S., Graesser A. (1997). Is It an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: Müller J.P., Wooldridge M.J., Jennings N.R. (eds) *Intelligent Agents III Agent Theories, Architectures, and Languages*. ATAL 1996. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1193. Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0013570
- Markowitz, H., B. Hausner, and H. Karr. (1962). *SIMSCRIPT: A Simulation Programming Language*. Memorandum RM-3310-PR, The RAND Corporation, Santa Monica, California.
- Russel, S. & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*, 4th Ed. Pearson.
- Schruben, L. (1983). Simulation Modeling with Event Graphs. *The Communications of the ACM*, 26, 957–963.
- Wagner, G. (2020). *Object Event Simulation*. Keynote Presentation SIMULTECH 2020.
- Wagner, G. (2018). Information and Process Modeling for Simulation – Part I: Objects and Events. Journal of Simulation Engineering, 1:1. https://articles.jsime.org/1/1
- Wagner, G. & Nardin, L. G. (2018). Adding Agent Concepts to Object Event Modeling and Simulation. In *Proceedings of the 2018 Winter Simulation Conference* (pp. 893–904). Pistacaway, NJ: IEEE. https://doi.org/10.1109/WSC.2018.863238
- Wooldrige, M. (2009). *An Introduction to MultiAgent Systems*, 2nd Ed. John Wiley & Sons.