Agent-based Systems for Intelligent Transport Systems

Lectures on “Intelligent transportation systems (1958)”
Hasselt University - 26 February 2016

dr.habil. Stéphane GALLAND

During this lecture, I will :

1. remain you Intelligent Transport Systems.
2. present the concepts of agent and multiagent systems.
3. detail the principles and advantages of the agent-based simulation.
4. show a simulator of emergency situation on a French highway.
5. show a simulator of a multidimentional urban environment.

Outline

Outline

1. Introduction
2. Intelligent Transport Systems
3. Agent-Based Model
4. Agent-Based Simulation
5. Conclusion

Automation factory in Amberg, Germany

- Worldwide data volume doubles every two years. By 2020, it will have grown to 40 zettabytes — a 50-fold increase within 10 years.
- Worldwide revenue of the IT and communications industries reached a record €2.84 trillion in 2013.
- Revenue from apps alone amounted to €66.2 billion in 2013 and will more than double by 2017.
- Digitalization boosts GDP — a 10% increase in the digitalization level of a country leads to 0.75% rise in per capita GDP.

Urbanization

Commuters in Jakarta

Growth of cities

- 2009: For the first time in history, more than 50% of the world's population lived in cities.
- 2050: 70% of the world's population will live in cities.

Megacities worldwide

- 1970: 2 megacities with more than 10 million inhabitants.
- 2025: 37 megacities; more than 13% of the world's population will live in a megacity.

Demographic Change

Morning gymnastics in Shanghai

World population

- 2012: 7.1 billion people.
- 2050: 9.6 billion people.

Worldwide life expectancy

- 2012: 70 years.
- 2050: 76 years.
- By 2050, the share of the population aged 60 or over will, for the first time, equal the share of the population younger than 15.


Ice sheets off the Icelandic coast

- 2012:
  Highest CO_2 concentration in the atmosphere in 800,000 years.
- 2001 to 2010:
  Warmest decade on record.

Cities

Smart Cities

Intelligent Transport Systems

Environment

Sensors
Communication Infrastructure
Smart Traffic Control

Vehicles

Environment-aware Vehicles
Connected Vehicles
Unmanned Vehicles

Outline

What is an Intelligent Transport Systems?

Definition

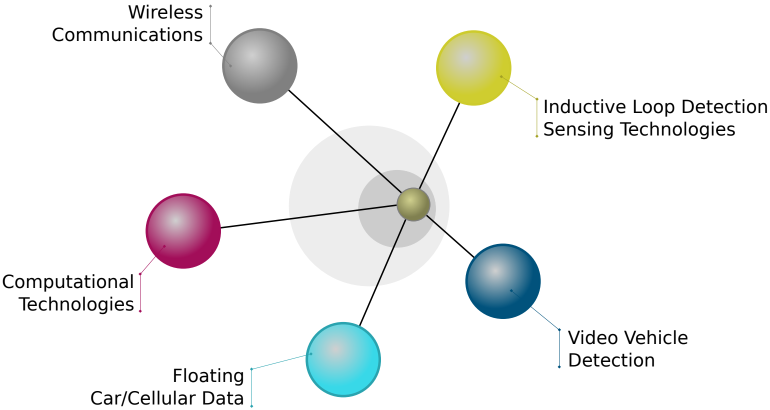
Intelligent Transport Systems (ITS) Systems in which information and communications technologies are applied to transport infrastructure and vehicles.

ITS improves

- Transport safety
- Transport productivity
- Travel reliability
- Informed travel choices
- Social equity
- Environmental performance
- Network operation resilience

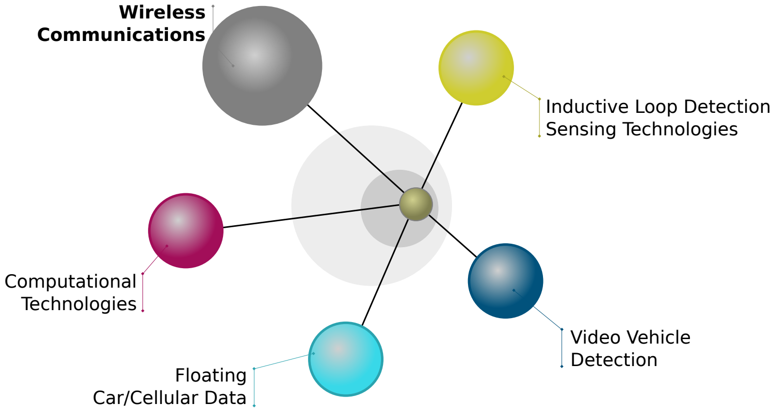
Why ITS?

- Interest in ITS comes from the problems caused by traffic congestion and a synergy of new information technology for simulation, real-time control, and communications networks.
- Traffic congestion has been increasing worldwide as a result of increased motorization, urbanization, population growth, and changes in population density.
- Congestion reduces efficiency of transportation infrastructureand increases travel time, air pollution, and fuel consumption.
- Many of the proposed ITS systems also involve surveillance of the roadways.
- Further, ITS can play a role in the rapid mass evacuation of people in urban centers after large casualty events, such as a result of a natural disaster or threat.



Wireless
Communications
Inductive Loop Detection
Sensing Technologies
Computational
Technologies
Video Vehicle
Detection
Floating
Car/Cellular Data

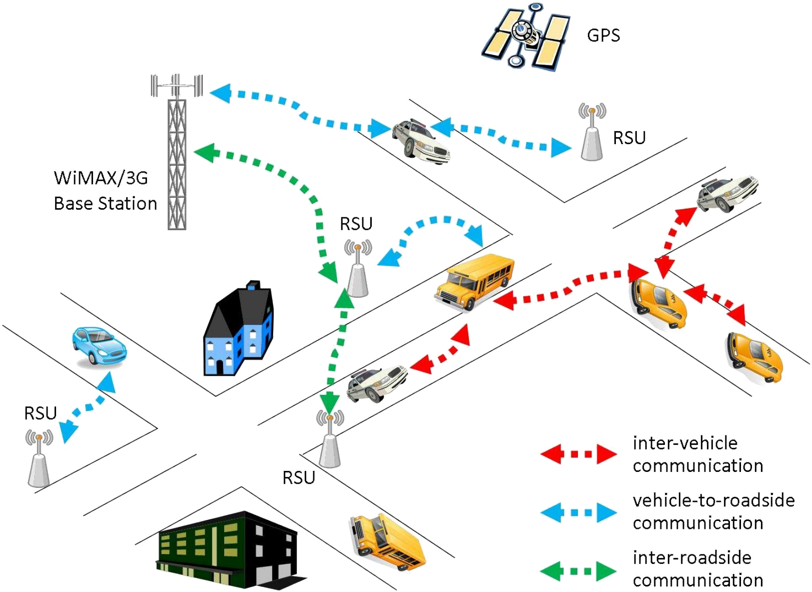
Intelligent Transport Technologies

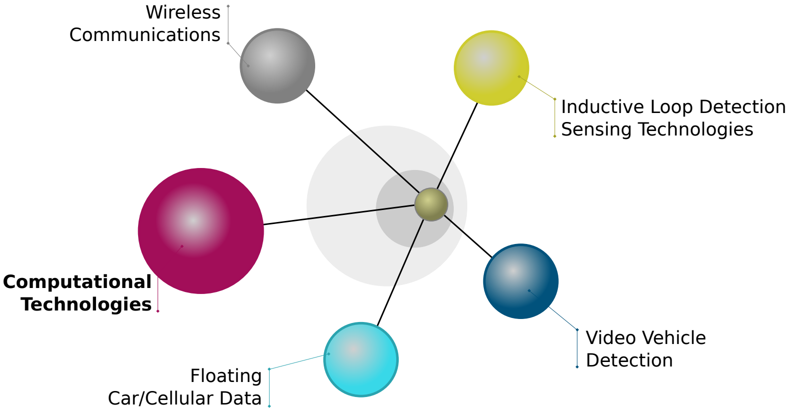


Wireless Communication

- Radio modern communication on UHF or VHF frequencies are widely used for short and long range communication within ITS.
- Short-range communications (less than 450 meters) can be accomplished using IEEE 802.11 protocols. Theoritically, the range of these protocols can be extended using Mobile ah-hoc networks or Mesh networking.
- Longer range communications have been proposed using infrastructure networks such as WiMAX (IEEE 802.16), Global System for Mobile Communications(GSM), or 3G/4G.

Wireless Communication

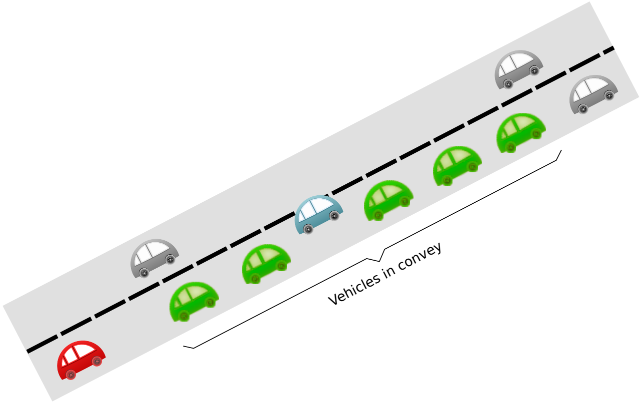



Wireless
Communications
Inductive Loop Detection
Sensing Technologies
Computational
Technologies
Video Vehicle
Detection
Floating
Car/Cellular Data

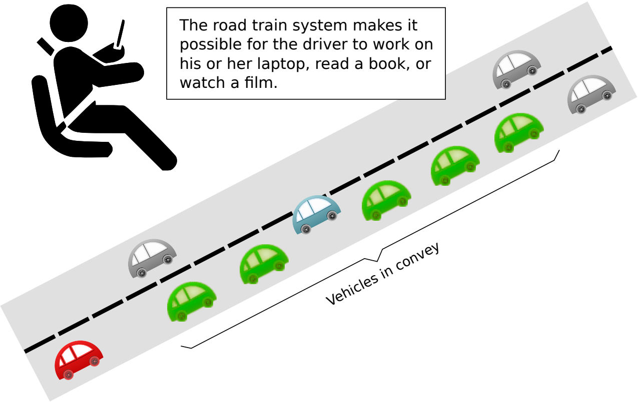
- A typical vehicle in the early 2000s would have between 20 to 100 individual networked microcontroller/programmable logic controller modules with non-real-time operating systems.
- The current tend is toward fewer, more costly microprocessor modules with hardware memory management and real-time operating systems.
- The new embedded system platforms allow for more sophisticated software applications to be implemented, included model-based process control, artificial intelligence, and ubiquitous computing.

- **Perhaps the most important of these for ITS is artificial intelligence.**

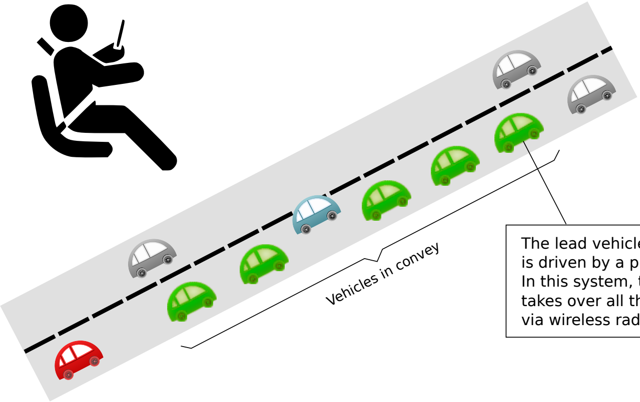
Example: Join a road train



Example: Join a road train

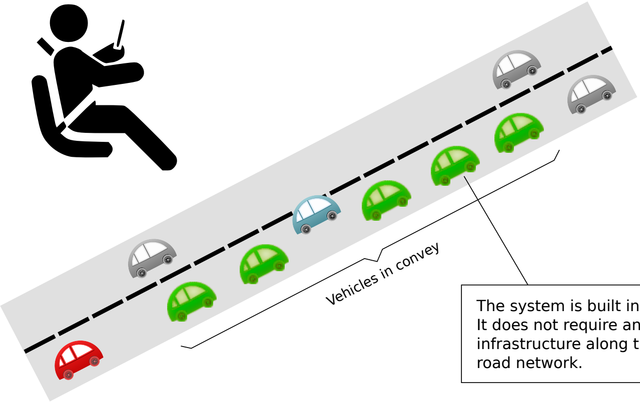


Example: Join a road train



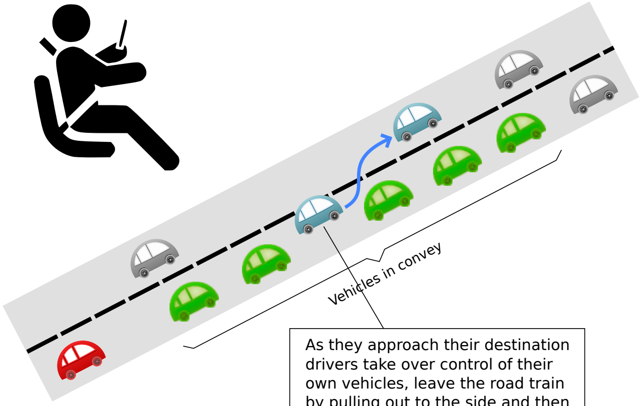
The lead vehicle, for instance a bus, is driven by a professional driver. In this system, the lead vehicle may takes over all the following vehicles via wireless radio communication.

Example: Join a road train



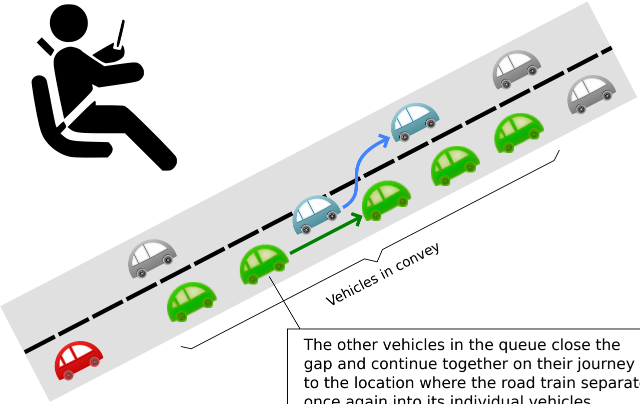
The system is built inside the cars.
It does not require any extended infrastructure along the existing road network.

Example: Join a road train



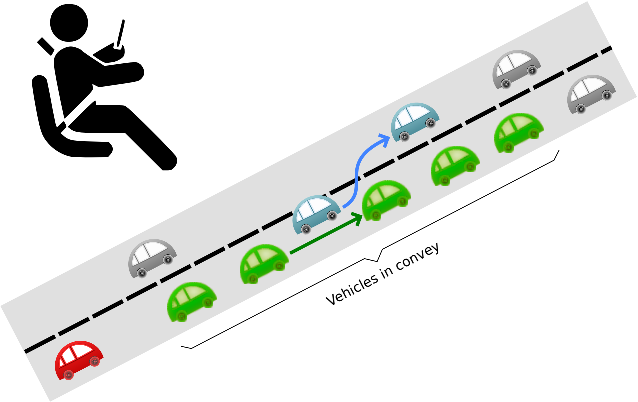
As they approach their destination drivers take over control of their own vehicles, leave the road train by pulling out to the side and then continue on their own path to their destination.

Example: Join a road train

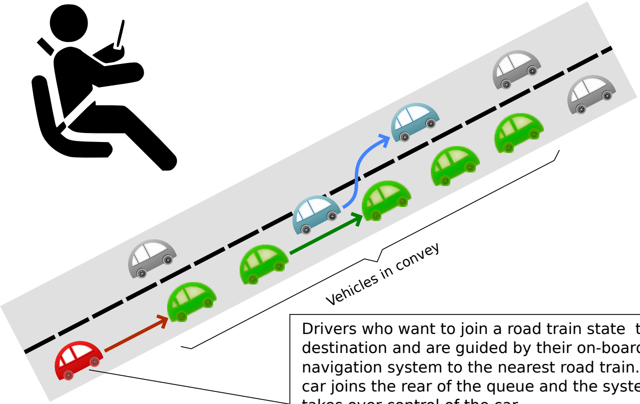


The other vehicles in the queue close the gap and continue together on their journey to the location where the road train separates once again into its individual vehicles.

Example: Join a road train



Example: Join a road train

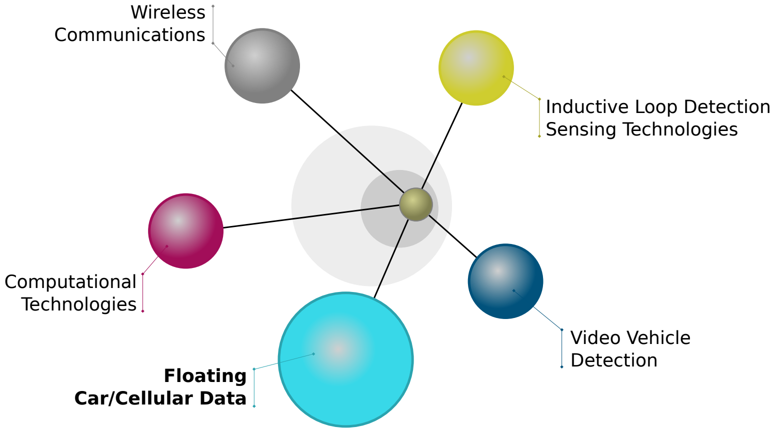


Drivers who want to join a road train state their destination and are guided by their on-board navigation system to the nearest road train. The car joins the rear of the queue and the system takes over control of the car.

Video

Project ANR SafePlatoon — 2012

Floating Car/Cellular Data



Floating Car/Cellular Data

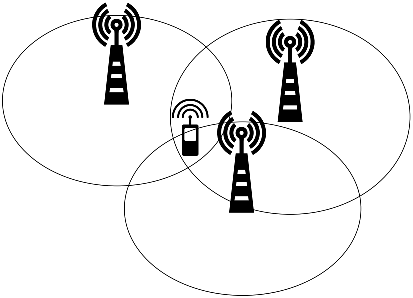
- "Floating car" or "probe" data collection contains travel speed and time data from vehicles traveling along streets, highways, freeways, and other transportation routes.

- Floating car data technology provides advantages over other methods of traffic measurements:
  - Less expensive than sensors or cameras,
  - More coverage (potentially including all locations and streets),
  - Faster to set up and less maintenance,
  - Works in all weather conditions, including heavy rain.

- Broadly speaking, three methods have been used to obtain the raw data:
  - Triangulation method,
  - Vehicle re-identification,
  - GPS based methods.

Triangulation Method

- Use mobile phones as anonymous traffic probes:
  - Mobile phones are moving with the vehicles.
  - Measuring and analyzing the network data using triangulation, pattern matching or cell-sector statistics $\Rightarrow$ converted into traffic flow information.
- Advantage: no infrastructure needs; only the mobile phone network is leveraged.
- By the early 2010s, the popularity of the triangulation method was declining.

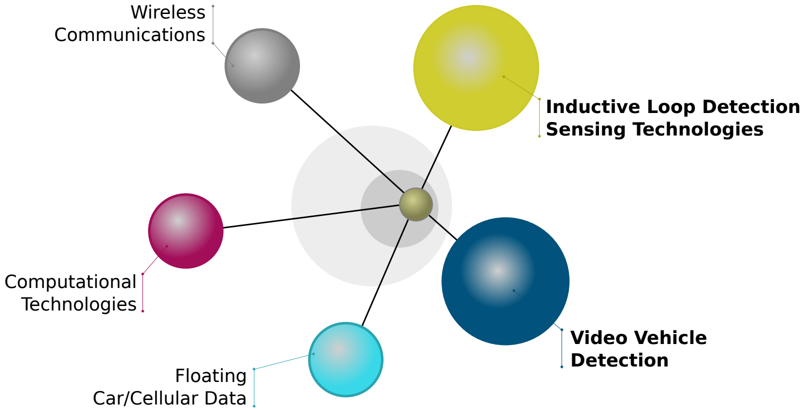


- Requires sets of detectors mounted along the road.
- An unique serial number of a device in the vehicle is detected at one location, and then detected again (re-identified) further down the road.
- Travel times and speeds are calculated by comparing the time at which a specific device is detected by pairs of sensors.
- Identifier may be the MAC (Machine Access Control) addresses from computer network devices, or the RFID serial numbers from Electronic Toll Collection (ETC) transponders.

GPS-based Methods

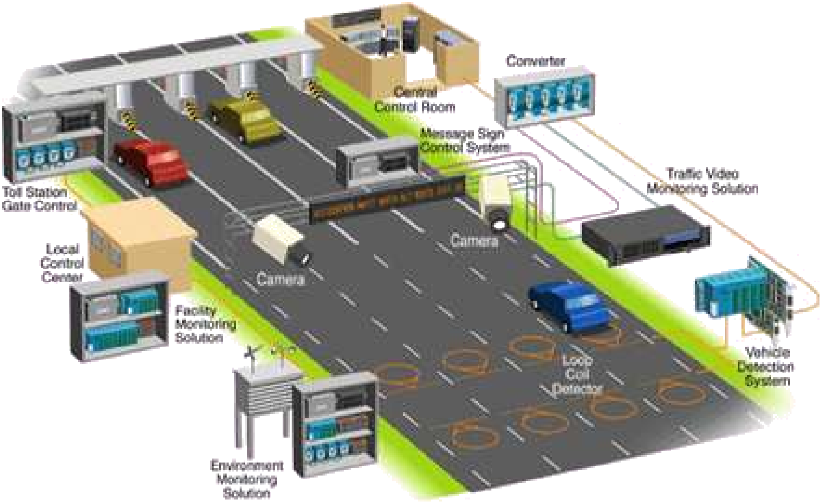
- An increasing number of vehicles are equipped with in-vehicle GPS (satellite navigation) systems that have two-way communication with a traffic data provider.
- Position readings from these vehicles are used to compute vehicle speeds.
- Ground positioning systems may be used rather than GPS. They are more accurate.

Sensor Technologies



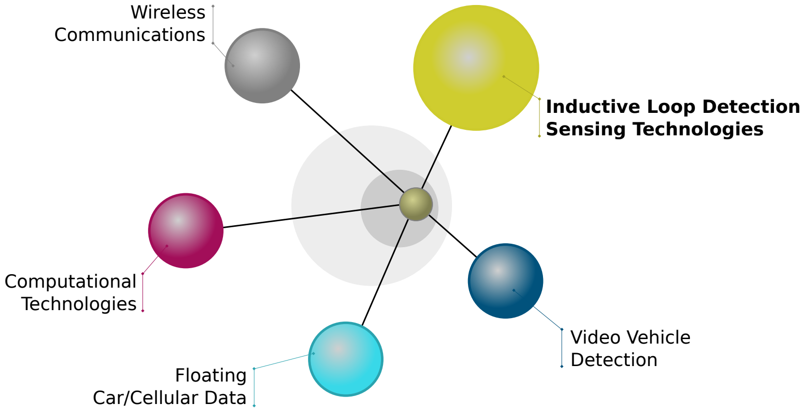
Sensor Technologies

- Sensing systems for ITS re vehicle-and infrastructure-based networked systems.
- Infrastructure sensors are indestructible devices that are installed or embedded in the road or surrounding the road.
- They may be manually disseminated during preventive road construction maintenance or by sensor injection machinery for rapid deployment.



Converter
Central Control Room
Message Sign Control System
Traffic Video Monitoring Solution
Toll Station Gate Control
Local Control Center
Camera
Camera
Facility Monitoring Solution
Loop Coil Detector
Vehicle Detection System
Environment Monitoring Solution

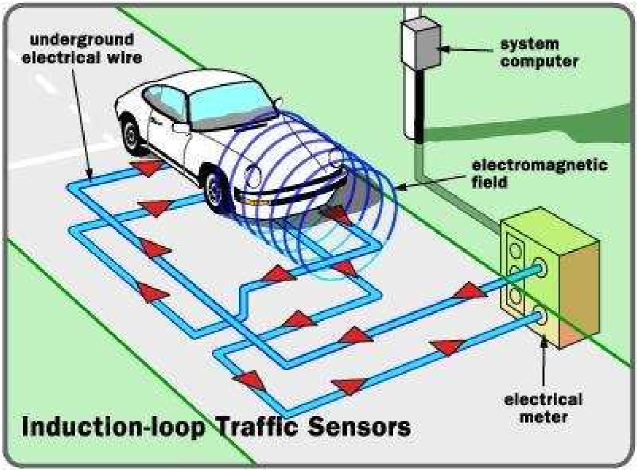
Inductive Loop Detection



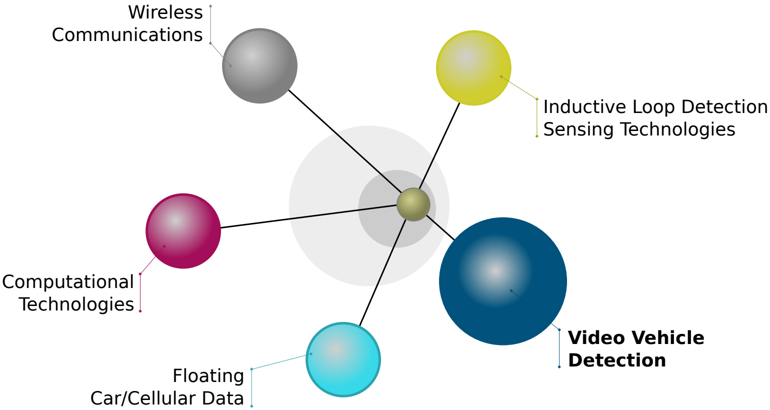
Inductive Loop Detection

- Inductive loops can be placed in a roadbed to detect vehicles as they pass through the loop's magnetic field.
- The simplest detectors count the number of vehicles during a unit of time that pass over the loop.
- While more sophisticated sensors estimate the speed, length, and weight of vehicles and the distance between them.
- Loops can be placed in a single lane or across multiple lanes, and they work with very slow or stopped vehicles as well as vehicles moving at high-speed.

Inductive Loop Detection



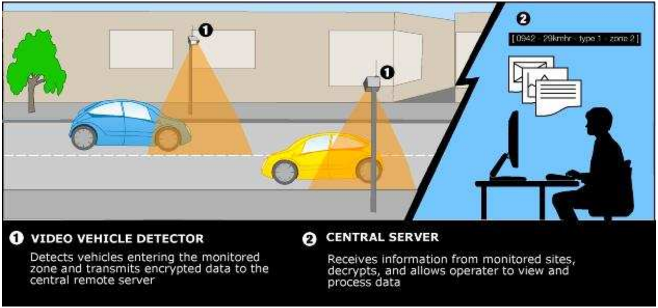
Video Vehicle Detection



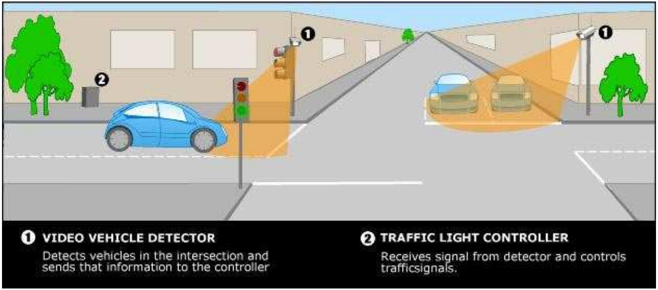
Video Vehicle Detection

- Traffic flow measurement and automatic incident detection using video cameras(gray-scale, color, 2D, 3D) is another form of vehicle's detection.
- "non-intrusive" traffic detection method: no component directly into the road surface or roadbed.
- Processors analyze the changing characteristics of the video image as vechicles pass.
- Most video detection systems require some initial configuration to "teach" the processor the baseline background image: usually inputting the distances between lane lines or the height of the camera above the roadway.

Video Vehicle Detection



Video Vehicle Detection



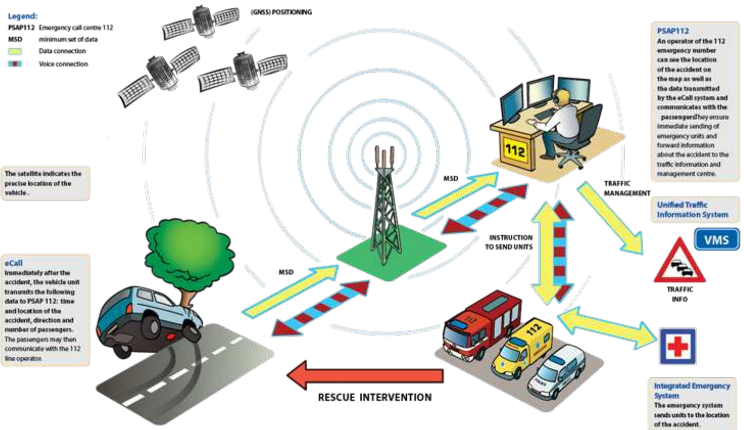
Other Sensing Technologies

1. Bluetooth detection.
2. Audio detection.
3. etc.

Intelligent Transport Applications

1. Emergency vehicle notification systems.
2. Automatic road enforcement.
3. Variable speed limits.
4. Collision avoidance systems.
5. Dynamic traffic light sequence.

eCall Alert System

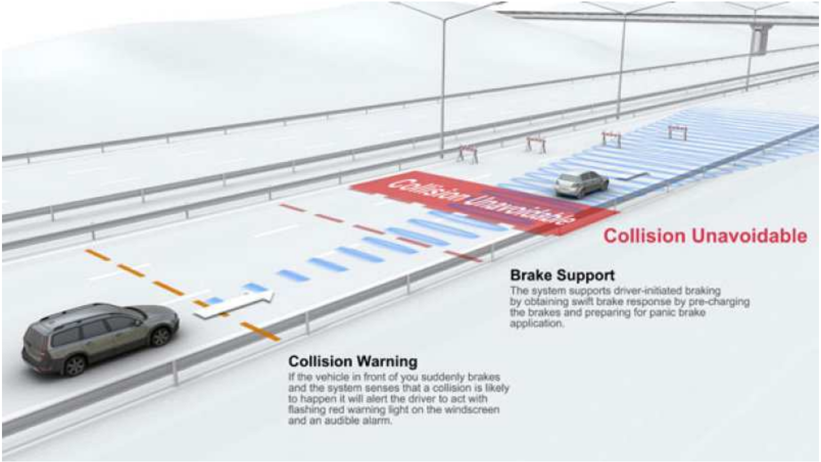


Automatic Road Enforcement

- Speed cameras: identify vehicles traveling over a legal speed limit (radar to detect vehicle's speed, electromagnetic loops buried in each lane).
- Red light cameras: detect vehicles that cross while a red traffic light is showing.
- Bus lane cameras: identify vehicles traveling in lanes reserved for buses.
- Level crossing cameras: identify vehicles crossing railways at grade illegally.
- White continuous line cameras: identify vehicles crossing these lines.
- High-occupancy vehicle lane cameras: identify vehicles violating HOV requirements.

Automatic Road Enforcement & Variable Speed Limit

Collision Avoidance Systems



How to create a model for ITS?

How to validate a model for ITS?

How to implement a model for ITS?

How to deploy a software for ITS?

A complex system is a system that exhibits some (and possibly all) of the following characteristics:

1. A big/huge number of interactions among entities.
2. Different levels of abstraction.
3. Hierarchical organization.
4. Emergent organization.
5. Local interactions and nonlinear relationships.
6. System's components may be considered as complex systems.
7. Feedback loops.

[Simon, 1996, Le Moigne, 1999, Holland, 1995]

An Agent-Oriented Approach for Modeling CS

[Henderson-Sellers, 2005]

Multiagent Systems (MAS) are considered as adapted for modeling complex systems.

Multiagent systems are well suited for:

- managing the heterogeneous nature of the system components.
- modeling the interactions between these components.
- trying to understand the emergent phenomena that result from these interactions.

Outline

Agent: a first definition

Agent [Wooldridge, 2001]

An agent is an entity with (at least) the following attributes / characteristics:

- Autonomy
- Reactivity
- Pro-activity
- Social Skills - Sociability

Autonomy of an Agent

Autonomy

Agents encapsulate their internal state (that is not accessible to other agents), and make decisions about what to do based on this state, without the direct intervention of humans or others;

- Able to act without any direct intervention of human users or other agents.
- Has control over his own internal state.
- Has control over his own actions (no master/slave relationship)
- Can, if necessary/required, modify his behavior according to his personal or social experience (adaptation-learning).

Reactivity

Agents are situated in an environment, (physical world, a user via a GUI, a collection of other agents, Internet, or perhaps many of these combined), are able to perceive this environment (through the use of potentially imperfect sensors), and are able to respond in a timely fashion to changes that occur in it;

- Environment static $\Rightarrow$ the program can execute itself blindly.
- Real world as a lot of systems are highly dynamic: constantly changing, partial/incomplete information
- Design software in dynamic environment is difficult: failures, changes, etc.
- A reactive system perceives its environment and responds in a timely appropriate fashion to the changes that occur in this environment (Event-directed).

Pro-activity of an Agent

Pro-activity

Agent do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative; They pursue their own personal or collective goals.

- Reactivity is limited (e.g. Stimulus $\Rightarrow$ Response).
- A proactive system generates and attempts to capture objectives, it is not directed only by events, take the initiative.
- Recognize/Identify opportunities to act/trigger something.

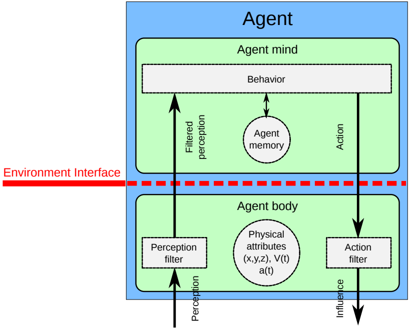
Sociability - Social Ability

Agents interact with other agents (and possibly humans), and typically have the ability to engage in social activities (such as cooperative problem solving or negotiation) in order to achieve their goals. Unity is strength.

- Many tasks can only be done by cooperating with others
- An agent must be able to interact with virtual or/and real entities
- Require a mechanism to exchange information either directly (Agent-to-Agent) or indirectly (through the environment).
- May require a specific (agent-communication) language.

An agent:

- is located in an environment (situatedness)
- perceives the environment through its sensors.
- acts upon that environment through its effectors.
- tends to maximize progress towards its goals by acting in the environment.



- Mobility: agent's ability to move through different nodes of a network/grid.
- Adaptability: ability to modify his actions/behavior according to external conditions and perceptions.
- Versatility: ability to perform different tasks or to meet different objectives.
- Trustiness: level of confidence that inspires the agent to delegate tasks, perform action, collaborate with other agents.
- Robustness: ability to continue to operate in fault situations, even with lower performances
- Persistence: Ability to keep continuously running by retrieving or saving their internal state even after a crash or unexpected situations.
- Altruism: disposition of an agent to assist other agents in their tasks.

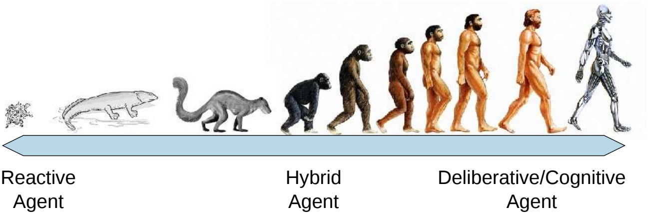
Agent: another Definition

Agent [Ferber, 1999]

Agent is a virtual (software) or physical entity which:

- is capable of acting in an environment;
- can communicate directly with other agents;
- is driven by a set of tendencies (in the form of individual objectives or of a satisfaction/survival function which it tries to optimize);
- possesses resources of its own;
- is capable to perceive its environment (but up to a limited extent);
- has only a partial representation of this environment (and perhaps none at all);
- possesses skills and can offer services; Add a comment to this line
- may be able to reproduce itself;
- whose behavior tends towards satisfying its objectives, taking account of the resources and skills available to it and depending on its perception, its representation and the communication it receives.

- Reactive: Each agent has a mechanism of **reaction to events**, without having an explanation/understanding of the objectives nor planning mechanisms. Typical Example: The ant colony.
- Cognitive/Deliberative: Each agent has a **knowledge** base that contains all information and skills necessary for the accomplishment of their tasks/goals and the management of his interactions with other agents and his environment: reasoning, planning, normative. Typical Example: Multi-Expert Systems.



Outline

Two perspectives/approaches on Multiagent Systems

Mono-agent approach

- The system is composed of a single agent.
- Example: Personal Assistant

Multi-agent approach

- The system is composed of multiple agents.
- The realization of global/collective task relies on a set of agents, on the composition of their actions.
- The solution emerges from the interactions of agents in an environment.

Multiagent systems: a first Definition

Multiagent systems

An MultiAgent Systems (MAS) is a system composed of agents that interact together and through their environment.

Interactions:
$\rightarrow$ Direct, agent to agent
$\rightarrow$ Indirect, Stigmergy, through the Environment

Multiagent systems: From local to global

Micro perspective (local): Agent

Individual level

- Reactivity - Pro-activity
- Autonomy
- Delegation

Macro perspective (global): Multiagent systems

Society/Community level

- Distribution
- Decentralization (control and/or authority)
- Hierarchy
- Agreement technologies (coordination)
- Emergence, social order/pattern, norms

Multiagent Systems [Ferber, 1999]

System comprising the following elements:

- An environment E, usually a space (with volume, 3D).
- An array of objects, O. These objects are situated.
- A set of agents, A, which are specific objects.
- A set of relations, R, which links the objects (and thus agents).
- A set of operations, Op, making it possible for agents to receive, produce, process and manipulate the objects in O.
- Operators with the task of representing the application of these operations and the reaction of the world to this attempt of modification.

Outline

Outline

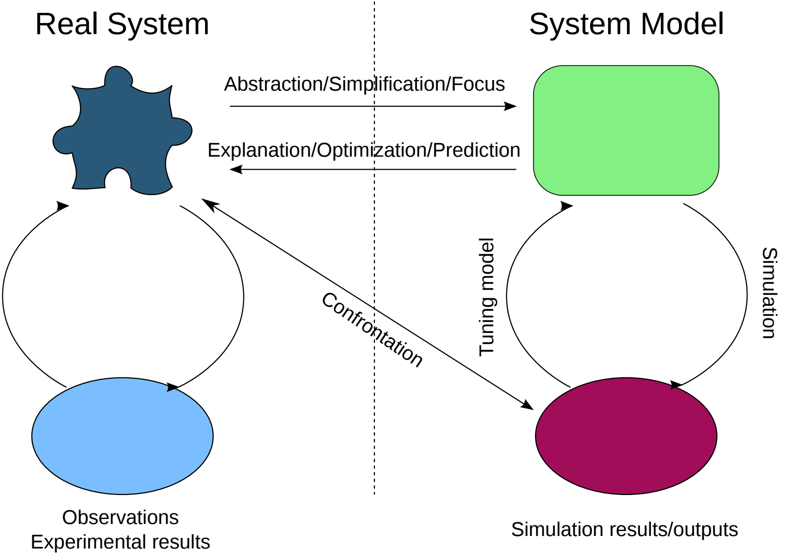
A Definition of Simulation

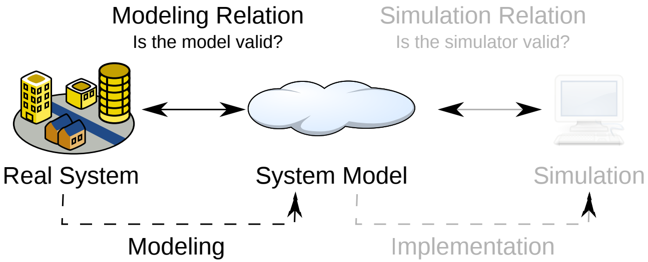
[Shannon, 1977]

The process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies (within the limits imposed by a criterion or a set of criteria) for the operation of the system.

Why simulate?

- Understand / optimize a system.
- Scenarii/strategies evaluation, testing hypotheses to explain a phenomenom (decision-helping tool).
- Predicting the evolution of a system, e.g. metrology.

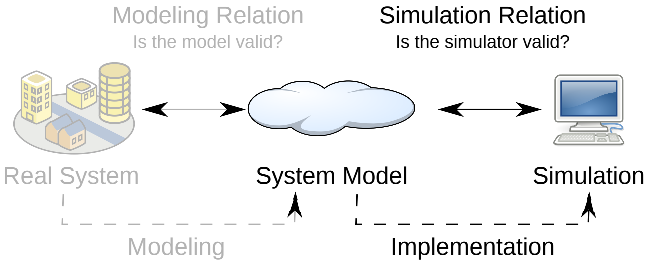

Real System
System Model
Abstraction/Simplification/Focus
Explanation/Optimization/Prediction
Confrontation
Tuning model
Simulation
Observations
Experimental results
Simulation results/outputs



- To determine if the system model is an acceptable simplificiation in terms of quality criteria and experimentation objectives.
- Directly related to the consistency of the model simulation.

[Zeigler, 2000]

Simulation Relation: Model $\leftrightarrow$ Simulator

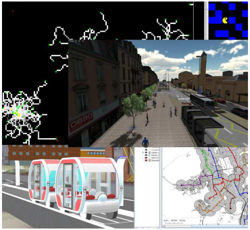


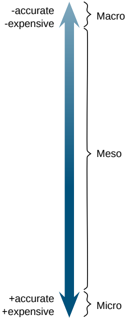
- To guarantee that the simulator, used to implement the model, correctly generates the behavior of the model.
- To be sure that the simulator reproduces clearly the mechanisms of change of state are formalized in the model.

[Zeigler, 2000]

Microscopic Simulation

- Explicitly attempts to model the behaviors of each individual.
- The system structure is viewed as emergent from the interactions between the individuals.

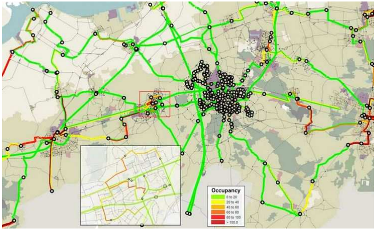


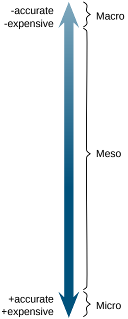


[Hoogendoorn, 2001, Davidsson, 2000]

Mesoscopic Simulation

- Based on small groups, within which elements are considered homogeneous.
- Examples: vehicle platoon dynamics and household-level travel behavior.

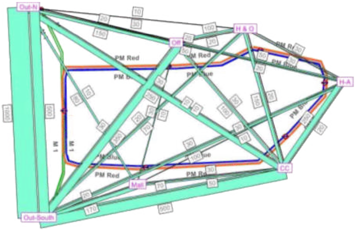


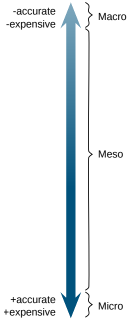


[Hoogendoorn, 2001, Davidsson, 2000]

Macroscopic Simulation

- Based on mathematical models, where the characteristics of a population are averaged together.
- Simulate changes in these averaged characteristics for the whole population.
- The set of individuals is viewed as a structure that can be characterized by a number of variables.

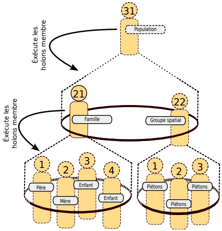


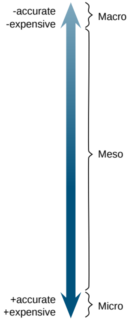


[Hoogendoorn, 2001, Davidsson, 2000]

Multilevel Simulation

- Combines various levels.
- Specify how the different levels are interacting together: one is input of the other, dynamic selection of the level.

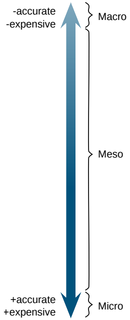


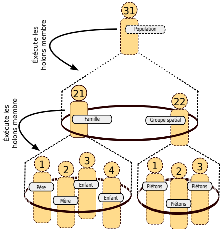


[Hoogendoorn, 2001, Davidsson, 2000, Galland, 2014]

Multilevel Simulation

- Combines various levels.
- Specify how the different levels are interacting together: one is input of the other, dynamic selection of the level.





Multiagent-based Simulation (MABS), aka. ABS, is traditionnally considered as a special form of microscopic simulation, but not restricted to.

Agent-based Simulation: General Idea

- Create an artificial world composed of interacting agents.
- The behavior of an agent results from:
  - its perceptions/observations;
  - its internal motivations/goals/beliefs/desires;
  - its eventual representations;
  - its interaction with the environment (indirect interactions, ressources) and the other agents (communications, direct interactions, stimuli).
- Agents act and modify the state of the environment through their actions.
- We observe the results of the interactions like in a Virtual Lab $\Rightarrow$ Emergence.

Main Characteristics and Advantages

- More flexible than macroscopic models to simulate spatial and evolutionary phenomena.
- Dealing with real multiagent systems directly: real Agent = simulated Agent.
- Allows modelling of adaptation and evolution.
- Heterogeneous space and population.
- Multilevel modeling: integrate different levels of observation, and of agent's behaviors.

Limitations and Drawbacks

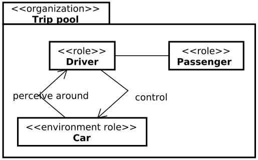
- Offer a significant level of accuracy at the expense of a larger computational cost.
- Require many and accurate data for their initialization.
- It is difficult to apply to large scale systems.
- Actual simulation models are costly in time and effort.

Outline

Driving Activity

- Each vehicle is simulated but road signs are skipped ⇒ mesoscopic simulation.
- The roads are extracted from a Geographical Information Database.

- The simulation model is composed of two parts [Galland, 2009]:
  1. the environment: the model of the road network, and the vehicles.
  2. the driver model: the behavior of the driver linked to a single vehicle.



Road Network

- Road polylines: $S = \left\{ \langle path, objects \rangle \middle| path = \langle (x_0, y_0) \cdots \rangle \right\}$
- Graph: $G = \{S, S \mapsto S, S \mapsto S\} = \{\text{segments}, \text{entering}, \text{exiting}\}$

Operations

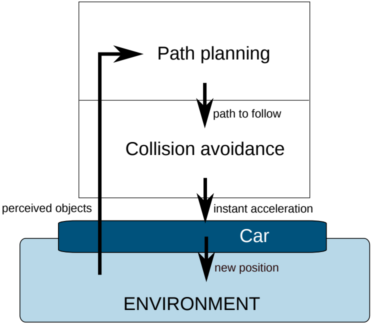
- Compute the set of objects perceived by a driver (vehicles, roads...):

$$P = \left\{ o \middle| \begin{array}{c} distance(d, o) \leq \Delta \wedge \\ o \in O \wedge \\ \forall (s_1, s_2), path = s_1.\langle p, O \rangle.s_2 \end{array} \right\}$$

  where *path* is the roads followed by a driver *d*.
- Move the vehicles, and avoid physical collisions.

[Galland, 2009]

Architecture of the Driver Agent



Jasim model [Galland, 2009]

- Based on the A* algorithm [Dechter, 1985, Delling, 2009]:
  - extension of the Dijkstra's algorithm: search shortest paths between the nodes of a graph.
  - introduce the heuristic function *h* to explore first the nodes that permits to converge to the target node.

- Inspired by the D*-Lite algorithm [Koenig, 2005]:
  - A* family.
  - supports dynamic changes in the graph topology and the values of the edges.

- Principle: compute the acceleration of the vehicle to avoid collisions with the other vehicles.
- Intelligent Driver Model [Treiber, 2000]

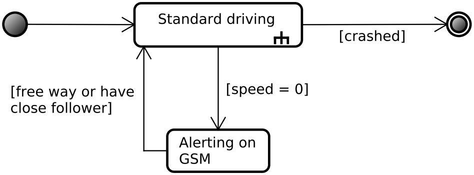
$$\texttt{followerDriving} = \begin{cases} -\dfrac{(v\Delta v)^2}{4b\Delta p^2} & \text{if the ahead object is far} \\ -a\dfrac{(s+vw)^2}{\Delta p^2} & \text{if the ahead object is near} \end{cases}$$

- Free driving:

$$\texttt{freeDriving} = a\left(1 - \left(\frac{v}{v_c}\right)^4\right)$$

What is simulated?

1. Vehicles on a French highway.
2. Danger event → "an animal is crossing the highway and causes a crash".
3. Alert events by GSM.
4. Arrival of the security and rescue services.



voxelia

http://www.voxelia.com

Video

Video done with the SIMULATE® tool — 2012 © Voxelia S.A.S

http://www.voxelia.com

Outline

Environment has several dimensions [Odell, 2003]

- Physical:
  - Principles and processes that govern and support a population of entities
  - Each agent has a body corresponding to its physical representation [Saunier, 2015].
- Communication:
  - Principles, processes and structures to transport information between agents.
- Social:
  - Principles, processes and structures to support coordinated interaction between agents in a communication environment.

Hypothesis

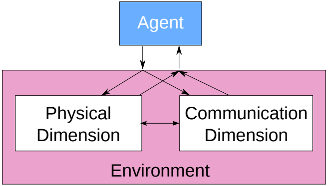
A change of state in a dimension can cause a change in another or several dimensions.

Solution 1

Agent as a propagation vector.

Solution 2

Interactions between dimensions.



How to model interactions between the dimensions?

[Galland, 2015]

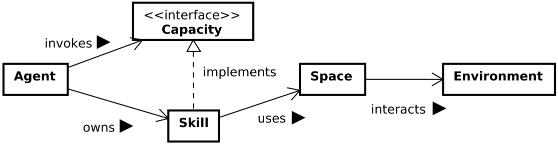
- Multi-dimension diffusion / polymorphism:
  - An event/message can be simultaneously interpreted (differently) by several dimensions.
  - Example: GPS Alert may change the social status and spatial indicator of dangerousness.

- Propagation of interactions:
  - An event/message in a given dimension generates another event/message in the other dimension.
  - Example: the detection of physical collision may trigger an emergency message in the communication dimension.

- Constrained perception:
  - A perception of an event/message in a dimension may be constrained by the properties associated with the other dimensions.
  - Example: a traffic light (physical dimension) perceives only emergency vehicles (social status) that are close (physical dimension) to the light.

[Galland, 2015]

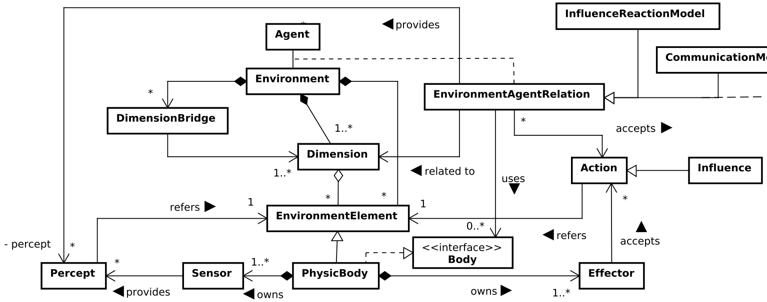
Modeling based on the concepts of the SARL's metamodel.



- Space: interaction place between agents or between agents and their environment.
- Agent: autonomous entity with a set of skills to realize the capacities it exhibits.
- Capacity: specification of a collection of actions.
- Skill: possible implementation of a capacity.
  Skills realized by behaviors mapping a collection of perceptions represented by Events to a sequence of Actions. Event is the specification of some occurrence in a Space that may potentially trigger effects by a listener

[Galland, 2015]

Proposal: Environment Model



[Galland, 2015]

"PhysicalSpace" supporting the interaction between agents and the physical dimension

- Emit influences from a specific body.
- Destroy the body related to an agentification.

```
space PhysicalSpace {
  var env : Environment
  def getBodyFactory: PhysicBodyFactory
    {}
  def putEnvironment(body: AgentBody,
                     perceptionListener:
     Agent) {
    env.add(body).for(perceptionListener
     )
  }
  def influence(body : AgentBody,
                influences : Influence*)
     {
    for (i : influences) emit(i, env.
     scope)
  }
  def destroyBody(body : AgentBody) {
    env.destroyObject(body)
  }
}
```

```
skill RoadEnvironmentSkill
      implements RoadEnvironmentCapacity
      {
  var body : AgentBody
  def install {
    body = bodyFactory.newInstance
    getSpace(PhysicSpace)
       .putEnvironment(body, owner)
  }
  def influence(inf: influence) {
    getSpace(PhysicSpace).influence(body
     , inf)
  }
  def uninstall {
    getSpace(PhysicSpace).destroyBody(
     body)
  }
  ...
}
```

[Galland, 2015]

Exchanges of messages on the Internet

```
space InternetSpace extends EventSpace {
  val env : Environment
  def emit(e : Message, scope : Scope) {
    e.destination = scope
    super.emit(e, new Scope(env))
  }
  def register(agent : Agent) : Address
     {
    super.register(agent)
  }
  def unregister(agentAddress : Address)
      {
    super.unregister(agent)
  }
}
```

```
event Message {
  var destination : Scope
}

capacity InternetCapacity {
  def emit(e : Message,
           scope : Scope = null)
}

skill InternetSkill
      implements InternetCapacity {
  def install {
    getSpace(InternetSpace)
       .register(owner)
  }
  def emit(e : Message, scope : Scope=
     null) {
    getSpace(InternetSpace).emit(e,
     scope)
  }
  def uninstall {
    getSpace(InternetSpace).unregister (
     owner)
  }
}
```

[Galland, 2015]

Content

- One instance of the models for each environment dimension.
- Rules of interaction between the dimensions, defined with:
  - a predicate p: rule activation condition,
  - a function f: actions to perform when the rule is activated.

Missions

- To compute the environment reactions from the agent influences.
- To compute the perceptions for each agent in the physical dimension.
- To propagate messages within the communication dimension.

[Galland, 2015]

- When receiving an influence, the rules are applied, and the influence is preserved if no rule is deleting it.
- When receiving a message, a similar algorithm is applied.
- Influences and saved messages are stored for later use in the lifecycle.

```
behavior Environment {
  var roads : RoadNetwork
  var physicSpace : space
  var communicationSpace: space
  ...
  on Influence {
    if (applyRules(occurrence, occurrence.object)) { saveInfluence(occurrence) }
  }
  on Message {
    for (participant : this.socialSpace.participants) {
      if (occurrence.scope.matches(participant) \&\& applyRules(occurrence,
     participant))
       { saveMessage(occurrence) }
    }
  }
}
```

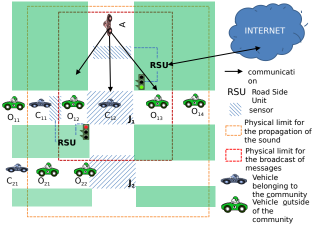
[Galland, 2015]

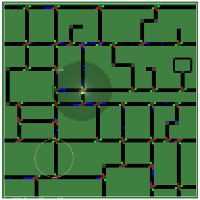
Principles

- Simulation of traffic and car crashes.
- Simulation of the displacements of the emergency cars.
- “Green wave” for emergency cars.

Dimensions of the environment

- Physical: Road network, traffic lights, road sensors.
- Communicational: Wireless Network, RSU, Internet.



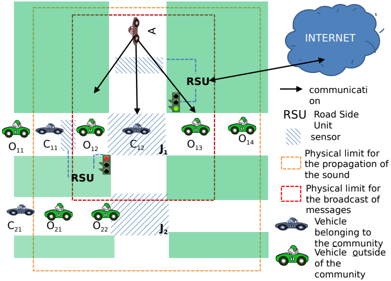


[Galland, 2015]

Agent *A* broadcasts a priority request within its community:

- Priority request sent in the communication dimension, but the broadcast is limited according to the position of vehicles to the physical environment by the V2X propagation model.

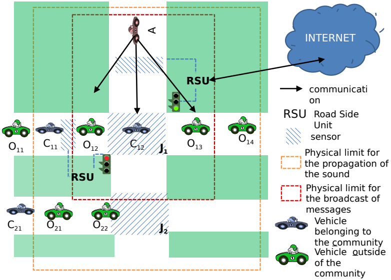
```
rules +=
   [ env, e, o |
     e instanceof PriorityRequestMessage
      ]
  =>
   [ env, e, o |
     e.scope = Scopes.addresses(
        env.roads.vehiclesAtDistance(
           e.source, env.physicSpace.
     V2X_distance)
        )
   ]
```



[Galland, 2015]

Agent A sends a priority request (resp. Siren influence) that is transformed into Siren influence (resp. priority request).

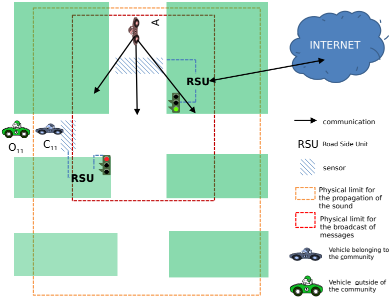
```
rules += [ env, e, o |
           e instanceof Siren ]
=> [ env, e, o |
     env.communicationSpace.emit(
       new PriorityRequestMessage(e.
      source)
     )
   ]

rules += [ env, e, o |
           e instanceof
      PriorityRequestMessage  ]
=> [ env, e, o |
     env.physicSpace.influence(
       new Siren(e.source)
     )
   ]
```



[Galland, 2015]

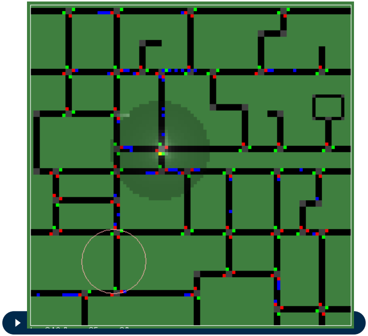
A physical collision involves sending an alert message in the communication dimension.

```
rules +=
  [ env , e, o |
    e instanceof PhysicalCollision ]

=>

  [ env, e, o |
    env.emit( new Alert(e.position))
  ]
```



[Galland, 2015]

Demo



This demo is realized with the Jaak Library

[Galland, 2015]

Benefits of ITS

- Time savings.
- Better emergency response times and services.
- Reduced crashes and fatalities.
- Cost avoidance.
- Increased customer satisfaction.
- Energy and environmental benefits.
- Decreasing of probability of congestion occurrence.

Safety

- Road crashes cause suffering and loss of life. Many collisions occur due to the stop-start nature of traffic in congested areas. $\Rightarrow$ ITS for smoothing traffic flows, reducing congestion and reducing certain types of accidents.
- Cooperative-ITS: involves communications between vehicles and road-side infrastructure
  $\Rightarrow$ improve safety by providing warnings on heavy braking or potential collisions at intersections.
- Information provided through ITS can also be used to direct traffic away from accidents and alert emergency services as soon as the accident occurs.

Three Key Benefits (2/3)

Productivity

- Congestion:
  - lowers productivity,
  - causes flow-on delays in supply-chains, and
  - increases the cost of business.

$\Rightarrow$ ITS can increase productivity by finding innovative ways to increase the capacity of our current infrastructure.

Three Key Benefits (3/3)

Environmental Performance

- ITS enables the reduction of congestion and stop-start driving.
- It can also enables the reduction of fuel consumption and greenhouse gas emissionsc compared with normal driving conditions.

Why Simulation and Agent-Based Simulation?

Why Simulating?

- Too dangerous to deploy in real World.
- Too costly to deploy in real World.
- Rapid prototyping.
- Testing standard and extrem scenarios.
- Debugging of the algorithms.

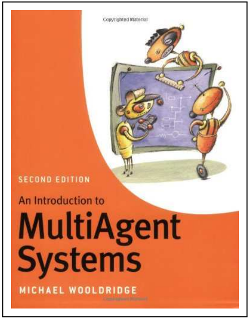
Why Agent-based Simulation?

- Natural modeling paradigm for ITS.

Thank you for your attention. . .

Appendix

Outline

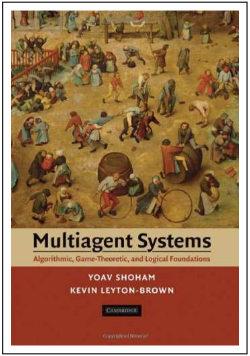


An Introduction to Multiagent Systems
2nd edition

Michael WOOLDRIDGE

Wiley, 2009

ISBN 0-47-0519460

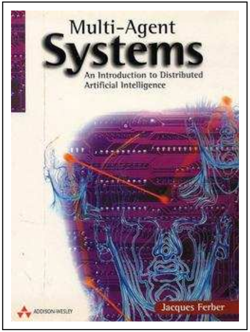


Multiagent Systems: algorithmic, game-theoretic, and logical foundations

Yoav SHOHAM and
Kevin LEYTON-BROWN

Cambridge University Press, 2008

ISBN 0-52-1899435

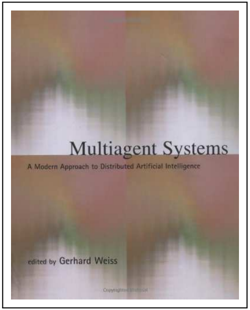


Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence

Jacques FERBER

Addison Wesley, 1999

ISBN 0-20-1360489

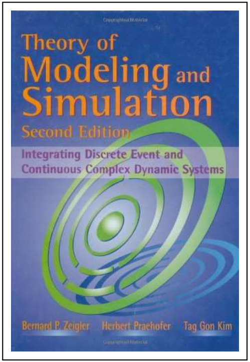


Multiagent Systems: a modern approach to distributed Artificial Intelligence

Gerhard WEISS

MIT Press, 2000

ISBN 0-26-2731312

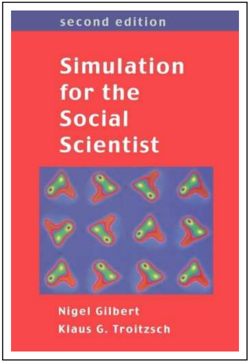


Theory of Modeling and Simulation
2nd edition

Bernard ZEIGLER, Herbert Praehofer, and Tag Gon Kim

Academic Press, 2000

ISBN 0-12-7784551



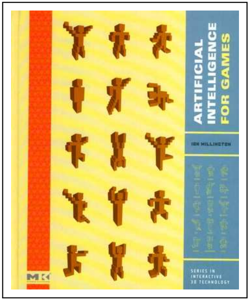
Simulation for the Social Scientist

2nd edition

Nigel GILBERT and Klaus TROITZSCH

Open University Press, 2005

ISBN 0-33-5216005

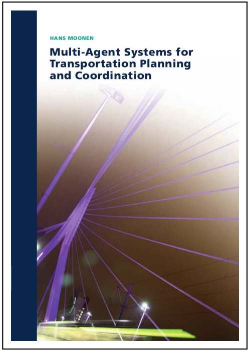


Artificial Intelligence for Games

Ian MILLINGTON

Morgan Kaufmann Publishers & Elsevier Science, 2006

ISBN 0-12-4977820

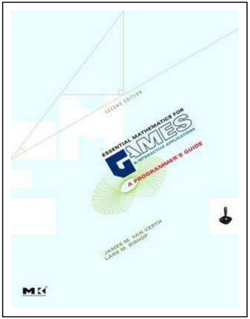


Multi-Agent Systems for Transportation Planning and Coordination

Hans MOONEN

Erasmus Research Institute of Management, 2009

ISBN 978-90-5892-216-8



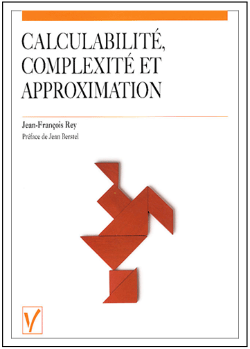
Essential Mathematics for Games & Interactive Applications: a programmer's guide

2nd edition

James VAN VERTH and Lars BISHOP

Morgan Kaufmann Publishers, 2008

ISBN 0-12-3742971



Calculabilité, Complexité et Approximation

Jean-François REY

Vuibert France, 2004

ISBN 2-71-1748081

Professor

Université de Bourgogne Franche-Comté

Université de Technologie de Belfort-Montbéliard, France

Topics: Multiagent systems, Agent-based simulation, Agent-oriented software engineering, Mobility and traffic modeling

Web page: `http://www.multiagent.fr/People:Galland_stephane`
Email: stephane.galland@utbm.fr

Open-source contributions:

- `http://www.sarl.io`
- `http://www.janusproject.io`
- `http://www.aspecs.org`
- `http://www.arakhne.org`
- `https://github.com/gallandarakhneorg/`

Outline

Davidsson, P. (2000).
Multi agent based simulation: Beyond social simulation.
Multi Agent Based Simulation, LNCS series, 1979.

Dechter, R. and Pearl, J. (1985).
Generalized best-first search strategies and the optimality of a*.
J. ACM, 32(3):505–536.

Delling, D., Sanders, P., Schultes, D., and Wagner, D. (2009).
Engineering route planning algorithms.
In Lerner, J., Wagner, D., and Zweig, K., editors, *Algorithmics of Large and Complex Networks*, volume 5515 of *Lecture Notes in Computer Science*, pages 117–139. Springer Berlin Heidelberg.

Ferber, J. (1999).
Multiagent Systems: An Introduction to Distributed Artificial Intelligence.
Addison-Wesley Professional.

Galland, S., Balbo, F., Gaud, N., Rodriguez, S., Picard, G., and Boissier, O. (2015).
Contextualize agent interactions by combining social and physical dimensions in the environment.
In Demazeau, Y., Decker, K., De la prieta, F., and Bajo perez, J., editors, *Advances in Practical Applications of Agents, Multi-Agent Systems, and Sustainability: The PAAMS Collection. Lecture Notes in Computer Science 9086.*, pages 107–119. Springer International Publishing.

Galland, S., Gaud, N., Demange, J., and Koukam, A. (2009).
Environment model for multiagent-based simulation of 3D urban systems.
In *the 7th European Workshop on Multiagent Systems (EUMAS09)*, Ayia Napa, Cyprus.
Paper 36.

Bibliography (#2)

Galland, S., Gaud, N., Demange, J., and Koukam, A. (2014).
Multilevel model of the 3D virtual environment for crowd simulation in buildings.
In *1st International Workshop on Agent-based Modeling and Simulation of Cities (AgentCities14)*, pages 822–827, Hasselt, Belgium. Elsevier.
Procedia Computer Science, vol. 32.

Henderson-Sellers, B. and Giorgini, P., editors (2005).
Agent-Oriented Methodologies.
Idea Group publishing.

Holland, J. H. (1995).
Hidden order: how adaptation builds complexity.
Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

Hoogendoorn, S. P. and Bovy, P. H. (2001).
State-of-the-art of vehicular traffic flow modelling.
Special Issue on Road Traffic Modelling and Control of the Journal of Systems and Control Engineering, 215(4):283–303.

Koenig, S. and Likhachev, M. (2005).
Fast replanning for navigation in unknown terrain.
Robotics, IEEE Transactions on, 21(3):354–363.

Le Moigne, J.-L. (1999).
La modélisation des systèmes complexes.
DUNOD, 4ième edition.

Odell, J. J., Parunak, H. V. D., Fleischer, M., and Brueckner, S. (2003).
Modeling agents and their environment.
In *Proceedings of the 3rd International Conference on Agent-oriented Software Engineering III*, Lecture Notes In Computer Science, pages 16–31, Berlin, Heidelberg. Springer-Verlag.

Saunier, J., Carrascosa, C., Galland, S., and Kanmeugne, P. s. (2015).
Agent bodies: An interface between agent and environment.
E4MAS 2014 - 10 years later, LNCS, 9068(1):1–16.

Shannon, R. E. (1977).
Simulation modeling and methodology.
SIGSIM Simul. Dig., 8(3):33–38.

Simon, H. A. (1996).
The Science of Artificial.
MIT Press, Cambridge, Massachusetts, 3rd edition.

Treiber, M., Hennecke, A., and Helbing, D. (2000).
Congested traffic states in empirical observations and microscopic simulations.
Phys. Rev. E, 62:1805–1824.

Wooldridge, M. and Ciancarini, P. (2001).
Agent-oriented software engineering: The state of the art.
In *Agent-Oriented Software Engineering: First International Workshop (AOSE 2000)*, volume 1957 of *Lecture Notes in Computer Science*, pages 1–28. Springer-Verlag.

Zeigler, B. P., Praehofer, H., and Kim, T. G. (2000).
Theory of Modeling and Simulation.
Academic Press, 2nd edition edition.

Bibliography (#4)