Coupling Simulation platforms : challenges and solutions

Khadim Ndiaye

September 2017

MINES Saint-Étienne

Outline

Outline

Illustration: Goals and Scenario

Goals

- Test decentralized coordination strategies for vehicles seeking parking spots in an urban area
- Validate the strategies by adding physically controlled traffic lights in a hardware/software simulation

Scenario

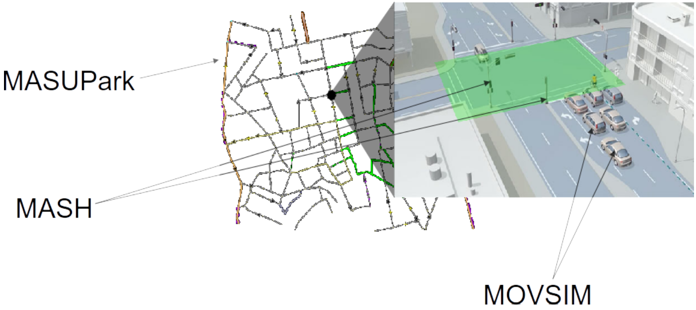
- Vehicles move around a spatial environment (road network) where free spots appear dynamically
- The vehicles cooperate in a decentralized way, to optimize their research time
- There are physically controlled signal lights along with sensors that evaluate traffic flows

Simulators

Movsim (Treiber et al., 2010): microscopic traffic simulator, is used to process physical movement of vehicles

MASH (Jamont et al., 2009): software/hardware simulator. MASH integrates real signal lights into the software simulation.

MasUPark (Zargayouna et al., 2016): multi-agent based simulator. It helps implement decentralized coordination models.



Outline

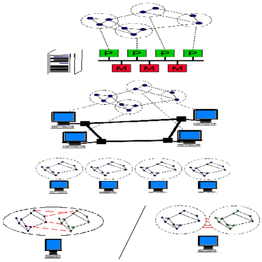
Definition

Coupling simulations

is the joint execution of independently developed simulations, exchanging data in order to achieve a set of defined goals (Yilmaz, 2004; Tolk et al., 2003)

related notions

Type	Model cardinality	Connectivity
Parallelism	1	Sole computer with concurrent processors
Distributed	1	Several computers exchanging over a network : peer to peer, middleware,...
Repliquated	1	Several independent computers
Strong / weak coupling	several	Code integration / interoperation



Interest

Interest of coupling simulations

- reuse of already built simulations
- bring together diverse expertises
- set up parallel and multi-level simulations
- facilitate the simulation of complex systems

Outline

Coupling typologies

Integrability

is the merging of simulators' code so that every relevant functionality is reproduced under a single simulator

Interoperability

is the ability of two or more simulators to exchange and use information:

- process oriented: orchestrated with simulators cooperating by the mean of defined protocols
- data oriented: achieved with synchronization solely on exchanged data

Composability

is aiming at conceptual models' interoperation and alignment independently from their technical implementation

Outline

Interoperability challenges

Data distribution

- How to wire data from one simulation to another in a technical point of view (communication protocols) (Riley et al., 2004)
- How to interface simulators that use diverse data formats (syntax)

Data alignment

- How to achieve knowledge alignment (Tolk et al., 2003) on the shared data (semantic)
- How to adapt shared informations to make them consumable by the simulators with different data models, manage differences in scales (spatial and temporal)

Illustration of interoperability challenges

Illustration of interoperability challenges:
$MASH = \{light, sensor\}$, $MasUPark = \{assistant\}$,
$Movsim = \{vehicle\}$

sharing \ receiving	assistant	vehicle	light	sensor
assistant	x	spot	x	x
vehicle	position	x	x	speed
light	x	status	x	x
sensor	x	x	flow	x

- The simulations need to agree beforehand on how and what they are exchanging with precise syntaxing
- Shared data as *parking spots* from MasUPark , have no representation in their receiving simulation (Movsim)
- Other data like *position* have different representations whether in MasUPark or Movsim for instance

Outline

Synchronization challenges

Causality principal

Events occurring within simulations must be processed with respect to their timestamps order (Fujimoto, 2001)

Synchronization in time

How to handle a consistent evolution of the simulations in time with respect to the casualty principal ?
There are two time synchronization approaches (Fujimoto, 1998):

- conservative: wait until events are safe to process
- optimistic: allow local causality violations, but detect them and recover using rollback mechanism

Synchronization challenges (bis)

Shared entities

are concepts in the system that are represented at least in two different simulations, and on which we may have concurrent access.
Example: the environment in multi agent based simulations

Synchronization on shared entities

- Shared entities constraint that their state is common among the simulations that represent them
- How to handle constraints induced by the existence of shared entities across the simulations ?

Illustration of synchonization challenges

Illustration of synchronization challenges:

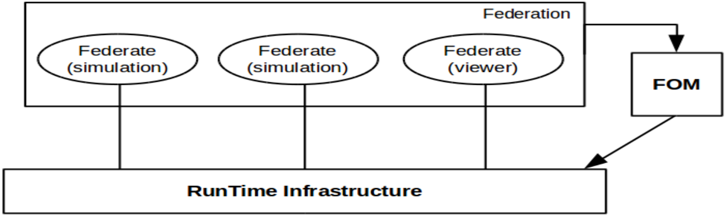
- MASH, MasUPark and Movsim each have their own time scale and clock. How do we ensure the causality principle ?
- MasUPark agents decide on chosen spots depending on the position of other agents in the environment. The positions being processed by Movsim vehicles, there must be a consistent view of the environment state in both simulations

Outline

High Level Architecture

Principles (US, 1998; Fujimoto, 1998):

- Type : process oriented interoperability
- Architecture: Federate, Federation, RTI
- Runtime infrastructure: data sharing model (OMT), temporal synchronization
- Conception rules



Advantages:

- High level abstraction : simulator independent and language free
- IEEE supported standard

Illustration

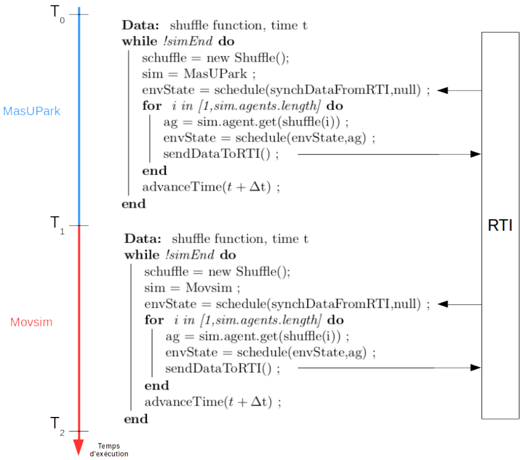
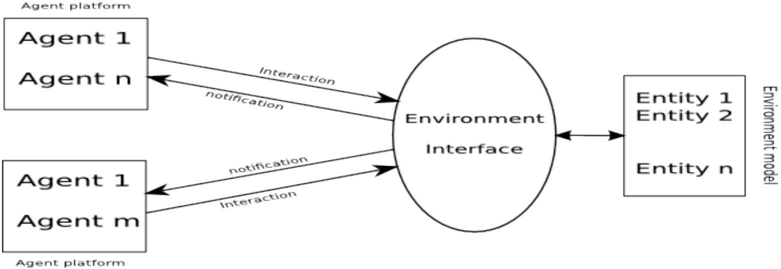


Figure: sequential scheduling

Environnement Interface Standard

Principles (Behrens et al., 2011):

- Type : data oriented interoperability
- Agent / Environment separation
- Shared environment model for agent platforms
- Standardization for platform/environment exchanges



Advantages:

- Portability, genericity, heterogeneity
- Spatial synchronization by environment sharing

Illustration

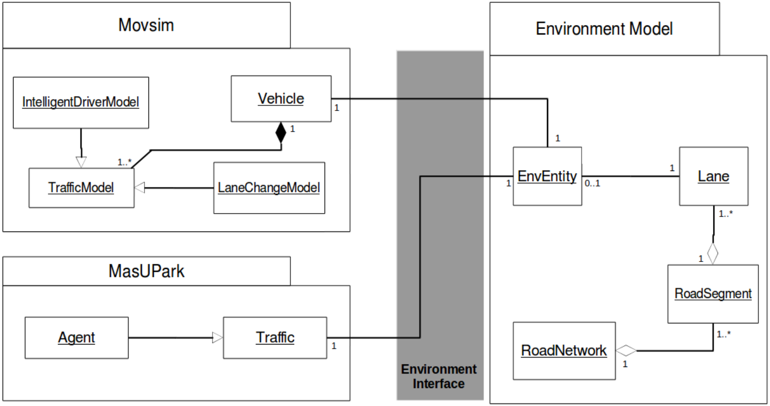
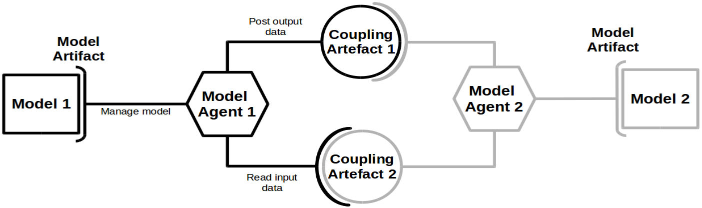


Figure: Environment model with EIS

MECSYCO

Principles (Camus et al., 2016):

- Type : Composability of models
- Assumptions on simulated models
- "Agents & Artifact" paradigm
- Coordination by conservative synchronization
- Proposed coupling methodology



Advantages:

- agent paradigm
- decentralized synchronization

Outline

Adressed challenges

Table: solutions and adressed challenges

	HLA	EIS	MECSYCO
Data distribution	Runtime Infrastructure	Controllable entities	Peer to peer
Data alignment	Federate Object Model	Interface Immediate Language	ad-hoc functions (artifacts)
Time synch	Chandy/Misra	none	Distributed Chandy/ Misra
Shared entities sync	none	Environment Interface	none

Limits

- Sustained integration efforts
- Uncertainty in the validation of the coupling
- No spatial synchronization (but EIS)
- No active scheduling control
- Strong assumptions on simulations

Further issues

Further issues

- Existing coupling solutions don't allow to independently model the coupling problem from it's implementation. Thus, bias can be induced and validation becomes tricky
- To guarantee a coherent coupling approach, themathicians should clearly express the coupling requirements that undermine their problem, separately from how it's executed.

Coupling behaviors (Movsim-MasUPark)

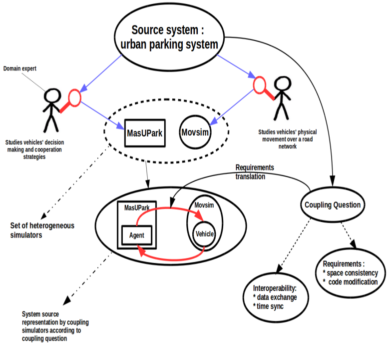


Figure: Coupling needs

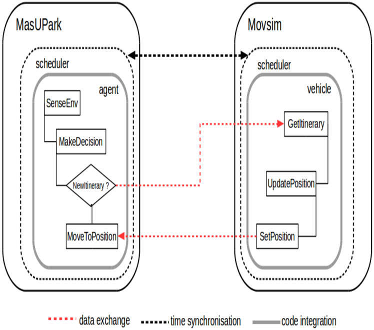


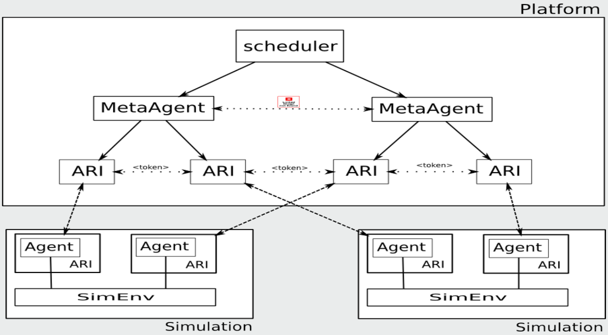
Figure: Movsim-MasUPark interactions

Further issues

Proposal

- Framework to describe coupling requirements
- Coupling behaviors with a middleware platform:
  - Agent Representation Interface for interoperability issues
  - Multi-agent organization for synchronization issues

Architecture



Behrens, Tristan M, Koen V Hindriks, and Jürgen Dix (2011). "Towards an environment interface standard for agent platforms". In: *Annals of Mathematics and Artificial Intelligence* 61.4, pp. 261–295.

Camus, Benjamin, Thomas Paris, Julien Vaubourg, Yannick Presse, Christine Bourjot, Laurent Ciarletta, and Vincent Chevrier (2016). "MECSYCO: a Multi-agent DEVS Wrapping Platform for the Co-simulation of Complex Systems". PhD thesis. LORIA, UMR 7503, Université de Lorraine, CNRS, Vandoeuvre-lès-Nancy; Inria Nancy-Grand Est (Villers-lès-Nancy, France). URL: https://hal.inria.fr/hal-01399978/ (visited on 01/27/2017).

Fujimoto, Richard M. "The High Level Architecture Outline". In:

— (1998). "Time management in the high level architecture". In: *Simulation* 71.6, pp. 388–400.

Fujimoto, Richard M. (2001). "Parallel simulation: parallel and distributed simulation systems". In: *Proceedings of the 33nd conference on Winter simulation*. IEEE Computer Society, pp. 147–157. URL: http://dl.acm.org/citation.cfm?id=564145 (visited on 01/27/2017).

Jamont, Jean-Paul and Michel Occello (2009). "A multiagent tool to simulate hybrid real/virtual embedded agent societies". In: *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*. IEEE Computer Society, pp. 501–504.

Riley, George F., Mostafa H. Ammar, Richard M. Fujimoto, Alfred Park, Kalyan Perumalla, and Donghua Xu (2004). "A federated approach to distributed network simulation". In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 14.2, pp. 116–148. URL: http://dl.acm.org/citation.cfm?id=985795 (visited on 03/13/2017).

Tolk, Andreas and James A Muguira (2003). "The levels of conceptual interoperability model". In: *Proceedings of the 2003 Fall Simulation Interoperability Workshop*. Vol. 7. Citeseer, pp. 1–11.

Treiber, M. and A. Kesting (2010). "An Open-Source Microscopic Traffic Simulator". In: *IEEE Intelligent Transportation Systems Magazine* 2.3, pp. 6–13. ISSN: 1939-1390. DOI: 10.1109/MITS.2010.939208.

US, Department Of Defense (1998). "High Level Architecture Interface Specification,Version 1.3 of IEEE P1516.1". In: April.

Yilmaz, Levent (2004). "On the need for contextualized introspective models to improve reuse and composability of defense simulations". In: *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology* 1.3, pp. 141–151.

Zargayouna, Mahdi, Flavien Balbo, and Khadim Ndiaye (2016). "Generic model for resource allocation in transportation. Application to urban parking management". In: *Transportation Research Part C: Emerging Technologies* 71, pp. 538–554. ISSN: 0968-090X. DOI: 10.1016/j.trc.2016.09.002. URL: //www.sciencedirect.com/science/article/pii/S0968090X16301589 (visited on 01/27/2017).