AgentPolis: Towards a Platform for Fully Agent-based Modeling of Multi-Modal Transportation (Demonstration)

Michal Jakob, Zbyněk Moler, Antonín Komenda†, Zhengyu Yin, Albert Xin Jiang, Matthew P. Johnson*, Michal Pěchouček† and Milind Tambe*
† Agent Technology Center, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic
{jakob, moler, komenda, pechoucek}@agents.fel.cvut.cz
* Computer Science Department, University of Southern California, Los Angeles, CA. USA.
{zhengyuy, jiangx, matthepj, tambe}@usc.edu

ABSTRACT

AgentPolis is a fully agent-based platform for modeling multi-modal transportation systems. It comprises a high-performance discrete-event simulation core, a cohesive set of high-level abstractions for building extensible agent-based models and a library of predefined components frequently used in transportation and mobility models. Together with a suite of supporting tools, AgentPolis enables rapid prototyping and execution of data-driven simulations of a wide range of mobility and transportation phenomena. We illustrate the capabilities of the platform on a model of fare inspection in public transportation networks.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multi-agent Systems; I.6.3 [**Simulation and Modeling**]: Applications

General Terms

Algorithms, Design, Experimentation

Keywords

agent-based modeling, multi-modal mobility, simulation, platform, transportation, fare inspection

1. INTRODUCTION

Over the last two decades, high-level, equation-based transportation modeling has been being replaced by micro-simulation approaches, which achieve higher accuracy by representing transportation systems at the level of individual people and vehicles [2]. Micro-simulation is particularly popular for vehicle traffic modeling, where it is now part of several commercial packages. Adoption of micro-simulation is slower in *mobility models*, which aim to capture how *people* (and freight), rather than just vehicles, move around in space and time using different means of transportation. The state-of-the art *activity-based approaches* model mobility by encoding people's choices regarding the type, location, and time of their activities and transportation between them.

MATSim [1] is the best known platform for activity-based mobility micro-simulation. Although termed agent-based and supporting individual-level modeling, MATSim treats individuals as passive data structures which can only be updated synchronously by central modules at infrequent, predefined points in time. This reduces the ability to add new types of agents to the model and to represent dynamic and multi-agent behavior.

To overcome these limitations, AgentPolis adopts a *fully agent-based* modeling approach. Individual entities of a transportation system are represented as autonomous agents with continuous, asynchronous control modules and the ability to interact freely with the environment and other agents. The agent-based approach reduces coupling and allows modeling scenarios in which agents adjust their plans at any time during the day based on their observations of the environment and/or communication with other agents. Agent-centric design also makes AgentPolis models usable as testbeds for evaluating innovative multi-agent mechanisms for transportation control and management.

2. PLATFORM OVERVIEW

The Java-based AgentPolis platform (see Figure 1 for an architecture overview) was designed to provide maximum reusability of elements shared by most models of transportation and mobility phenomena while allowing maximum flexibility in implementing model-specific parts.

2.1 Simulation Core

In the core of AgentPolis is a discrete-event simulation platform based on the Alite multi-agent toolkit[1]. The platform consists of a high-performance discrete-event processing engine and a set of domain-independent abstractions for building discrete-event agent-based models of large systems. The discrete-event model used is more resource-efficient compared to fixed-time-step execution models used by most agent-based simulation platforms. The refined set of abstract classes and interfaces compliant with the agent-based design paradigm provides a coherent foundation for building diverse agent-based models. This contrasts with highly purpose-specific designs of existing transportation mo-

[1] `http://agents.fel.cvut.cz/projects/#alite`

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

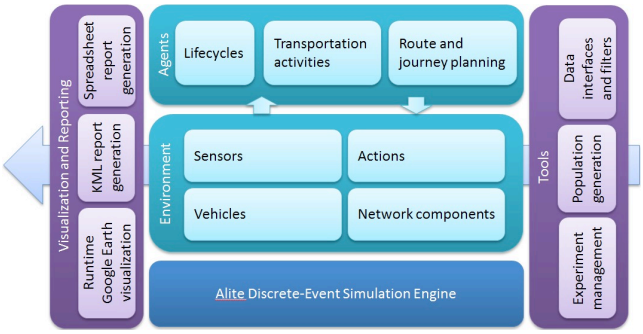

Figure 1: AgentPolis architecture.

dels, which make horizontal extension of models difficult.

2.2 Transportation Domain Model Library

Built on top of the base domain-independent abstractions, the transportation domain library provides a set of specific components for rapidly building transportation and mobility models. The library consists of the following software classes and modules:

- *transportation network components*—software classes for building transportation environments, in particular road, railway and metro networks, road intersections, public transportation stops and interchanges;
- *vehicles*—software classes representing vehicles (cars, buses, trains) and their properties;
- *transportation actions and sensors*—software classes mediating access between agents and their surrounding transportation environment, e.g., getting on/off a vehicle, moving along a street segment or detecting an arriving train;
- *transportation activities and lifecycles*—reactive control structures that can be composed to create a desired agent's behavior, e.g., travelling on a public transport or driving a vehicle between locations;
- *route and journey planning*—functional modules providing agent reasoning capabilities, in particular efficient route planning in road networks and multi-modal journey planning with public transportation services.

2.3 Simulation Tools

Rapid construction, execution and experimentation with AgentPolis models is supported by a range of tools:

- *Data interfaces and filters* allow to work directly with transportation-related data in standard formats, including OpenStreetMap format for map data, and Google Transit Feed Specification for public transportation networks and timetables.
- *Population generation tools* allow generating large numbers of agents with realistic distributions of demographic attributes (age, gender, income, car ownership etc.) based on real-world census data.
- *Experiment configuration, management and deployment tools* enable defining in a compact form experiment scenario batches and automatically executing them on available computing resources.
- *Visualization and reporting tools* allow viewing simulation runs as well as simulation results, including their geospatial and temporal context and aggregation over multiple runs and scenarios, in an interactive browser based on Google Earth.

3. APPLICATION TO FARE INSPECTION

Simulation of public transportation fare inspection is one of the models built using the AgentPolis platform. The model allows evaluating the effectiveness of fare inspection strategies provided by human experts or computational tools, in particular the TRUSTS (Tactical Randomization for Urban Security in Transit Systems)[3] system for scheduling randomized fare inspection patrols. TRUSTS adopts a game-theoretic approach, modeling the problem as a leader-follower Stackelberg game, and employs a novel compact representation of the mixed strategies as flows in a history-duplicate transition graph.

The agent-based model of fare inspection comprises two types of agents—the *FareAwarePassenger* has a standard daily travel pattern extended with ticket purchase logic determining whether a ticket should be purchased for a particular journey; the *TicketInspector* agent inspects passengers on specific trains and at specific stations according to a given patrolling schedule, provided, e.g., by the TRUSTS scheduler. The performance of each inspection strategy can be tested against passengers with different levels of rationality and observability of the environment. A number of performance metrics can be measured including fines collected, revenue lost and inspection coverage. So far, the fare inspection model was developed for the Los Angeles Metro system and involves the simulation of almost 400 thousand rides on five metro lines a day.

We are developing other models using the AgentPolis platform, including real-time ride sharing, auction-based taxi allocation and on-demand parcel delivery logistics. More information about new developments can be found on the platform's website[2].

Acknowledgements

Funded by Ministry of Education, Youth and Sports of Czech Republic (grant No. LD12044) and European Commission FP7 (grant agreement No. 289067). We acknowledge support from MURI grant W911NF-11-1-0332.

4. REFERENCES

[1] M. Balmer, K. Meister, M. Rieser, K. Nagel, and K. W. Axhausen. Agent-based simulation of travel demand: Structure and computational performance of MATSim-T. In *TRB Conference on Innovations in Travel Modeling*, 2008.

[2] K. Bernhardt. Agent-based modeling in transportation. *Artificial Intelligence in Transportation*, E-C113:72–80, 2007.

[3] A. Jiang, Z. Yin, C. Kietkintveld, K. Leyton-Brown, T. Sandholm, and M. Tambe. Towards optimal patrol strategies for fare inspection in transit systems. In *AAAI Spring Symposium on Game Theory for Security, Sustainability and Health*, 2012.

[2] `http://agents.fel.cvut.cz/projects/agentpolis/`