RinSim: A Simulator for Collective Adaptive Systems in Transportation and Logistics

Rinde R.S. van Lon, Tom Holvoet
DistriNet Research Group, dept. of Computer Science
KU Leuven, Leuven, Belgium
Rinde.vanLon@cs.kuleuven.be, Tom.Holvoet@cs.kuleuven.be

***Abstract*—Engineering collective adaptive systems (CAS) is a challenging task. Concurrent systems, esp. when being large-scale, are known to be hard to design as the overall system behavior non-linearly results from local behavior and interactions. They are also hard to engineer and debug, as time dependent errors are often hard to reproduce. Simulation tools and environments are often used to assist in this task.**

From our experience in developing and using simulators for decentralized systems (in traffic, logistics and smart power grid management), we learned that a simulation environment should comply to the following quality criteria.

First, from a software engineering point of view, a simulation environment itself must be designed up to the highest software quality standards - modularity, separation of concerns, test-driven development, guaranteed state consistency, etc. are particularly important quality criteria to ensure correctness, extensibility and manageability of the software.

Second, the simulation environment must provide convenient support for using and extending the simulation environment, ease the visualization of solutions, and - since its use in scientific process - offer direct support for evaluating CAS through the set-up of experiments.

In this paper, we present RinSim, an open source simulator that explicitly addresses these quality criteria, and targets the large family of transportation and logistics applications. RinSim separates the definition of the problem domain from the solution, has a modular design, is being developed in a test-driven way, etc. RinSim has been used and extended in a variety of cases within our research group, and served as the core platform in our educational program on multi-agent software development.

I. Introduction

Engineering collective adaptive systems (CAS) is a difficult task, simulators are often used during development. From our experience in research on CAS in various domains (logistics, traffic, smart grids) we learned that special purpose simulators lack flexibility and general purpose simulators lack decent, tailorable support for building your own simulation for a non-trivial field of study. In fact, simulation environments are non-trivial pieces of software which have complex requirements: time management, event management, extensibility and configurability for experiments.

In this paper we present RinSim[1] our open source simulator that simplifies the engineering of CAS and comes with support for transportation and logistics problems. Although logistics is a complex field, RinSim focusses on *simplicity* and *consistency*. The focus on simplicity results in a simulator that is easy to use, easy to extend and easy to comprehend. The focus on consistency increases the reliability of the simulator itself and of user code by adhering to the *fail-fast* principle. An important feature of RinSim is its explicit separation of problem and solution. This design decision simplifies development of CAS since it forces designers to separate these two aspects. When defining the problem one can think in terms of constraints which the solution should adhere to. When defining the solution one can freely explore the space of solutions that fall within the set of constraints as defined by the problem, without having to worry about violating consistency.

A design overview, the core features, our experiences and the conclusion are described in sections II-V respectively.

II. Design overview

In RinSim terminology, the parts of the simulation that define the problem and environment are called *models*, the parts of the simulation that solve the problem (i.e. the solution, the CAS) are called *agents*. The design of RinSim allows a minimally complex configuration by only using the models that are required. When the problem is configured, the programmer can focus on the solution without having to worry about accidentally violating simulation consistency. Figure 1 presents a high level component diagram.

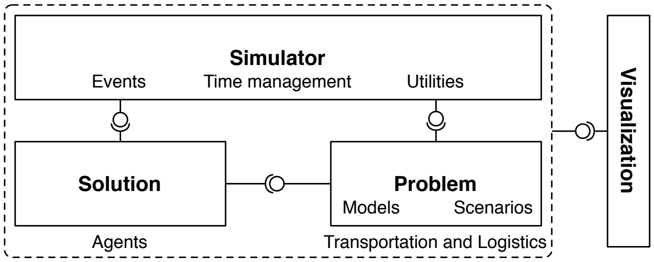

Figure 1. High level RinSim component diagram. The user defined solution part uses parts of the simulator and the problem. The core system is independent of the visualization which can optionally be used.

Actions that agents can take are considered to be part of

[1] https://github.com/rinde/RinSim/

the problem not the solution (e.g. a vehicle that can pickup things). Actions define the problem space in which a good solution has to be found.

III. Features

RinSim has a flexible and modular model design. Models focus on a single task and are independent of each other, allowing any combination of models. Generally, each model defines several constraints, it guarantees that either all constraints are met or in case of a violation it throws an exception. We describe the models and other features:

Road travel model: The road travel model is a model which supports moving objects in a 2D space. Currently it supports two types of 2D space: directed graphs and planes. The model takes care of several global properties such as enforcing position consistency and maximum speed.

Pickup and delivery model: This model is designed such that it can support the general pickup-and-delivery problem (PDP) as defined in [1]. This means it supports dial-a-ride problems, vehicle routing problems and pickup-and delivery problems each which may or may not be dynamic and/or has time windows. In the model vehicles can pickup and deliver parcels. The model enforces capacity constraints, (un)loading time constraints and position consistency.

Agent communication model: The agent communication model provides an easy to use system for sending and receiving messages between two parts of a system. It defines an inbox which participants can read to check if they have new messages. The communication model can also be used to broadcast messages to a certain area.

Time management: A system for time is used that enforces time consistency throughout the system.

Problem scenarios: Problem scenarios can easily be created and played in the simulator.

Graphs and OpenStreetMap data: The road model in RinSim is represented as a graph, a tuple of vertices and edges (V,E). Existing OpenStreetMap[2] graphs can be loaded and graphs can be serialized in the DOT format.

Visualization: Visualization is entirely separate from the core simulation, it can be easily defined and customized. The standard implementation is displayed in Figure 2.

Code quality: To ensure code quality of the system several industry practices have been adopted, among them are extensive use of unit tests, using a continuous integration server and static code and documentation analysis tools.

IV. Experiences

RinSim is used in several research projects and in our education program on multi-agent system (MAS) software development. We highlight three experiences:

Delegate MAS paper: In [2] RinSim was used for the implementation of Delegate MAS for PDP, this proved to be very effective and reduced development time.

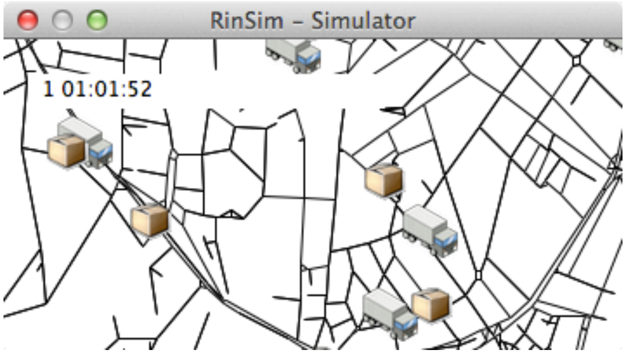

Figure 2. A screenshot showing the standard visualization. The black lines are a visualization of the roads, the boxes are the packages and the trucks are the vehicles.

Genetic programming paper: In [3], genetic programming was used for designing a MAS for the dial-a-ride problem. All kind of irrational agents emerged during the evolutionary design process, some which could potentially violate the problem constraints. The focus on consistency in RinSim allowed us to find these problems very quickly, greatly speeding up the development process.

RinSim as educational platform: For a course on MAS at the KU Leuven we have provided the students with the RinSim simulator. Compared to previous years, using RinSim the students were able to create higher quality solutions and were more enthusiastic about the project.

V. Conclusion

We have presented RinSim, an open source discrete time MAS simulator for solving transportation and logistics problems. The explicit separation of problem and solution and focus on simplicity in the design of the simulator has made it a very easy to use system for the design of CAS. Besides, the focus on consistency makes RinSim a very reliable choice when developing software for research.

Acknowledgments

This research is partially funded by the Interuniversity Attraction Poles Programme Belgian State, Belgian Science Policy, and by the Research Fund KU Leuven.

References

[1] S. N. Parragh, K. F. Doerner, and R. F. Hartl, “A survey on pickup and delivery problems. Part I: Transportation between customers and depot.” *Journal für Betriebswirtschaft*, vol. 58, no. 1, pp. 21–51, Mar. 2008.

[2] S. Hanif, R. R. S. van Lon, N. Gui, and T. Holvoet, “Delegate MAS for large scale and dynamic PDP: A case study,” in *Intelligent Distributed Computing V*, vol. 5, Delft, 2011

[3] R. R. S. van Lon, T. Holvoet, G. Vanden Berghe, T. Wenseleers, and J. Branke, “Evolutionary Synthesis of Multi-Agent Systems for Dynamic Dial-a-Ride Problems,” in *GECCO Companion ’12* , Philadelphia, USA, 2012.

[2] http://www.openstreetmap.org/