Towards a Real-world Simulator for Collaborative Distributed Learning in the Scenario of Urban Mobility

Andreas Poxrucker, Gernot Bahle, Paul Lukowicz
Embedded Intelligence
German Research Center for Artificial Intelligence
Kaiserslautern, Germany
Email:firstname.lastname@dfki.de

***Abstract*—Collaborative learning in collective adaptive systems is an active, open research area. In the Allow Ensembles project, we investigate this problem by a component called Evolutionary Knowledge. One problem arising in this context is that concepts of collaborative learning can hardly be studied without an actual real-world system. In this paper, we present our concept of a simulation tool of a real-world urban traffic system used as a framework to investigate collaborative learning. In contrast to existing ready-to-use traffic simulators, its purpose is not the accurate simulation of microscopic or macroscopic traffic flow models. Instead, it is used to generate data to train a knowledge model learning context parameters and their interrelations, which cannot be deduced from an analytical description of the system, but arise as emergent properties from the complexity of the system. Using the simulation we want to investigate the effects of different collaborative learning strategies on emergence in a complex urban mobility system applying different knowledge exchange patterns among entities. We describe the need for and the area of application of our simulator, show the differences to existing traffic simulation tools, and present an outline of its conceptual architecture.**

***Keywords*-distributed collaborative learning; socio technical systems; smart mobility**

I. Distributed Collaborative Learning in Large-scale Collective Adaptive System

Collective Adaptive Systems (CAS) consist of entities acting autonomously in a dynamic environment. The Allow Ensembles project [1] is concerned with the development of new ways of modelling and analysing CAS based on the concept of *cells* and *ensembles*. Each cell belongs to an entity and is associated with a certain *work flow*. A work flow is basically a sequence of individual activities specifying the behaviour of entities. One of the core principles of Allow Ensembles is the idea of collaboration between entities and is modelled using ensembles consisting of cells working together to reach the goals of their entities and overall the goal of the ensemble.

CAS are not be to considered static, but a key characteristic of them is the ability to adapt their behaviour to changes of their environment. It is thus reasonable to introduce a concept of *evolution*, which is realized in Allow Ensembles by a component called *Evolutionary Knowledge* (EvoKnowledge). In essence, it is a means of learning from experience including parameters of *context* and their interrelations, which cannot be deduced from an analytical representation of the system, but arise as a consequence of its dynamics and inherent complexity.

Introducing a learning component to CAS requires suitable machine learning models and algorithms. A central knowledge model which is shared by all entities of the system is thereby not a reasonable assumption thinking of real-world CAS, in particular. Instead, it is more realistic to reflect the structure of CAS using local knowledge models with the overall knowledge of the system being distributed among its entities. Due to the highly distributed nature of CAS as well as the usually large number of participating entities, applying classical machine learning techniques to this sort of learning problem is not an appropriate solution and new models and algorithms satisfying the previously-mentioned characteristics need to be developed. Adding the possibility to share knowledge between entities finally leads to the concept of *distributed collaborative learning*.

The research problems we are addressing in context of collaborative learning in CAS are thus twofold. First, it is necessary to derive an appropriate knowledge model which is able to represent distributions of parameters and possibly complex interrelations between them. We decided to experiment with a graphical modelling approach based on *conditional random fields*, whose description is, however, not the purpose of this paper. Second, the investigation of collaborative learning which can be realized in different ways. The possibility to use a global and centralized instance of a knowledge model is thereby not our main focus. Instead, it lies on *distributed collaborative learning* with every entity having a local instance of knowledge, which introduces the need for knowledge exchange triggers and mechanisms.

The problem of distributed collaborative learning is an active research area and far from being completely understood. Typical unsolved research problems from this context which we want to study in the Allow Ensembles project are given in the following list:

- How can collaborative learning be realized?
- What are the characteristics of centralized and dis-

tributed collaborative learning?
- How does knowledge spread within the system?
- How do different knowledge exchange patterns influence the spread of knowledge?
- Do certain exchange patterns or policies lead to the emergence of knowledge structures like knowledge hierarchies?
- How does collaborative learning influence the occurrence of emergent system properties?

II. The Urban Mobility Scenario

A perfect example of a real-world CAS can be found in the context of integrated urban mobility which is one of the leading scenarios in Allow Ensembles. The setting of the *urban mobility scenario* is an urban area in which various transport agencies provide different transportation services like taxis, buses, or trains.

In this scenario, we consider a simple ”lifecycle” executed by people which will also be the core procedure of the simulation and in which entities learn statistics about journeys they make. We assume, that people have access to a special smartphone application they use to get suggestions of possible ways to travel in the area. Specifying starting location, destination, starting time, preferred means of transportation, and a number of other parameters, people send their requests to a multimodel journey planner component creating a set of possible ways to travel. These solutions are described by a set of parameters like the time and distance to travel, or the means of transportation to use. The querying user chooses one of the journeys, executes it, and learns the experienced parameters using the underlying knowledge model.

The experienced parameters may, however, be different to the estimations of the planner. To see this, take the travel time as an example. It is estimated by the planner component based on deterministic calculations on the street network graph and the schedules of public transportation. The ultimately experienced travel time, however, may be significantly higher for example as there was a lot of traffic in the streets at this time of the day. Or maybe it was impossible to use the suggested bus because it was already full as it was rush hour and the weather was bad at that time. Factors like the time of the day and the weather, which we call context parameters, are not considered by the planner and their influence cannot be deduced analytically, but has to be learnt from experience.

III. Contribution

It is easy to see, that investigating *collaborative learning* in CAS requires a significant amount of data for training and evaluation purposes. It should, however, also be clear, that this is generally infeasible in real-world CAS. With respect to the urban mobility scenario, one would have to develop the smartphone application, distribute it to a relevant amount of people in a suitable urban area, and collect data over a long enough period of time. As if this was not already enough, performing experiments with different settings would increase the effort even more making evaluations in real-world CAS complex, expensive, and time-consuming.

We thus started to develop a simulation tool serving as a ”replacement” for the real-world which we present in this paper. It is settled within the scenario of an integrated urban mobility system presented in section II and is based on real-world data to keep to results as close to reality as possible. We present an outline of our ongoing work towards such a simulator explaining the concept of the simulation and highlighting the integration of collaborative learning, in particular.

The rest of this document is structured as follows. In section IV we delimit our simulator from existing traffic simulation tools explaining the different intents. Section V starts with the presentation of the features of our simulator and explains our efforts towards making it a ”real-world” simulation. Furthermore, we explain its conceptual architecture and the integration of knowledge and the triggering of knowledge exchange mechanisms for distributed collaborative learning. Finally, we summarize the content of this document in section VI and give an outlook on our future work.

IV. Towards a Real-world Simulation

Computer aided traffic simulation is an important means in traffic engineering and transportation planning. It is a well-known research area with many ready-to-use solutions already existing working with traffic flow models on different levels of granularity. Their purpose is the simulation of traffic flow given a transportation system (e.g. street network of a city) or just a part of it (e.g. a junction or a roundabout) to support planning and optimizing infrastructure. Two main approaches can be identified in literature. The more popular microscopic models on the one hand consider single vehicles describing their dynamic properties like the position or velocity individually. Famous simulators based on microscopic modelling include SUMO [2], VISSIM [3], or CORSIM [4]. Macroscopic traffic flow models like MASTER [5] or FREFLO [6] on the other hand consider traffic systems and networks from a more global perspective not focussing on individual vehicles.

Although the setting of these simulations is very similar to our solution, their purpose is totally different. Our solution uses urban mobility as an example of a CAS to study strategies and the effects of collaborative learning. Its purpose is not accurate simulation of traffic flows. It does neither take details like exact street geometries, multi-lane streets, or traffic lights into account, nor does it work on sophisticated movement models of the entities. The movement of entities along in the street network is just a means to the end to generate data to learn, trigger knowledge exchange, and finally study collaborative learning based on the simulation results.

Nevertheless, we want to create a simulation that is at least close to the real-world. We thus build an urban traffic system based on existing real- world data modelling the area of Trento, Italy. The street network is deduced from a real street map of Trento, which is obtained as a GIS shapefile from OpenStreetMap [7] and includes the geometry of the street network as well as movement relevant properties like the speed limit on the street segments. Information about the underlying public transportation network mainly come from the *Trento Mobility Service* [8], which is a web service exposing a REST-API to query the required information and which is integrated to our simulation in the *DataService* component (see next section). Data we can get from the service include:

- Real routes and schedules of public transportation in Trento. This includes buses and trains from different transport agencies (e.g. *Trento City Bus*, *Trentino Intercity Bus*, etc.).
- Real-time delay of public transportation.
- Real-time information about available parking slots in parking garages.
- Real-time information about road works and closed roads.

Querying travel suggestions requires a journey planner which must be compatible to the underlying transportation network. The previously-mentioned *Trento Mobility Service* additionally offers such a planner which can be queried using its route-planning REST-API. As an additional step towards a real-world simulation, we also want to include realistic traffic flow data from Trento describing at which time of the day people travel to which destination areas.

V. Simulating Collaborative Learning in an Urban Mobility System

A. Features

Before going into more detail about the architecture of the simulator the following list presents its main features and functionalities:

- **Simulation of a real-world urban traffic system.** To let entities learn travel parameters, it is necessary to simulate a more-or-less complete urban mobility system offering various means of transportation. Although this could be done using an artificially-created setting, we want create a simulation of a CAS that is as close to a real-world CAS as possible. We thus use a model of the area of Trento, Italy, as data about the street network, public transportation etc. are easily available.
- **Simulation of entity journeys.** In order to learn travel parameters, entities need to be able to query journeys from a planner component and execute them learning their parameters together with the context.
- **Training of knowledge models based on data obtained from simulated journeys.** The simulator allows entities to collect information during their journeys and use them as training data for their knowledge models.
- **Triggering of knowledge exchange.** To realize collaborative learning, it is necessary for the simulator to trigger the exchange of knowledge based on some patterns or rules.

B. Conceptual Architecture For Simulating Collaborative Learning

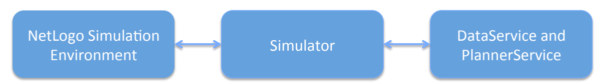

Figure 1. Conceptual architecture of the simulator

The conceptual architecture of our simulator is depicted in figure 1 . It is composed of three components. The *NetLogo Simulation Environment* [9] is a framework to develop multi-agent based, time-discrete simulations we use as underlying simulation engine to control the execution of our simulator. It provides a graphical user interface to start and stop the simulation, adjust the execution speed, specify an arbitrary set of input parameters, and observe possible output parameters in real time during execution. Additionally, NetLogo offers the possibility to graphically visualize the running simulation, i.e. the movement of entities on the map. This is a useful feature for demonstration purposes and can be switched off on demand to increase simulation speed. An example of the NetLogo environment with a loaded map and some entities travelling on it can be seen in figure 2.

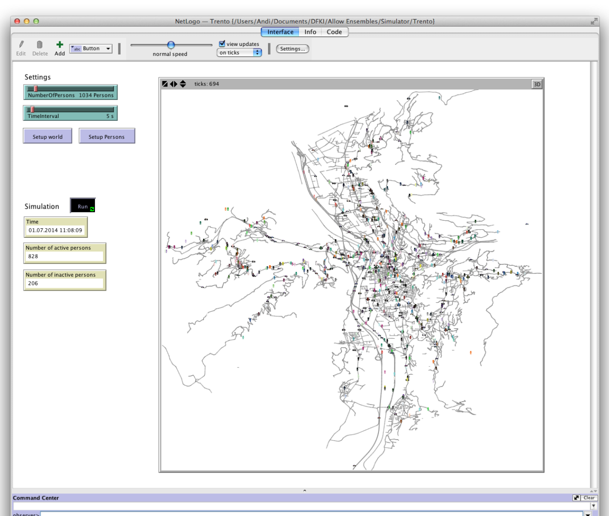

Figure 2. NetLogo environment with loaded map

The *DataService* offers information about the underlying urban mobility system. Among others, it provides geographical data, e.g. the street map or the position of stops of public transportation, and routes and schedules of available means of transportation. To query travel suggestions, simulated entities send their requests to the *PlannerService*. It creates a set of possible travel solutions based on the underlying transportation infrastructure described by the *DataService*. Working on real-world data, both services can be considered as "links" connecting the simulation to the real-world.

The *Simulator* is the core component of our simulation and is realized as an extension to NetLogo using its Java-based extension API. Nevertheless, there is only a loose coupling with the NetLogo simulation environment making it possible to use a different engine without any greater effort if necessary. As depicted in figure 3, the *Simulator* requires a number of input models and allows the observation of the state of the simulated world and of the individual entities. It is possible for example to monitor the knowledge models of the entities to analyse the spread of knowledge within the system, or to observe the global system state investigating the occurrence of emergent properties like traffic jams in the street. As an input, it requires models of the environment to simulate, of the entities and their behaviour, as well as a knowledge model for learning and triggers and mechanisms for exchange. The following paragraphs briefly summarize the mentioned models.

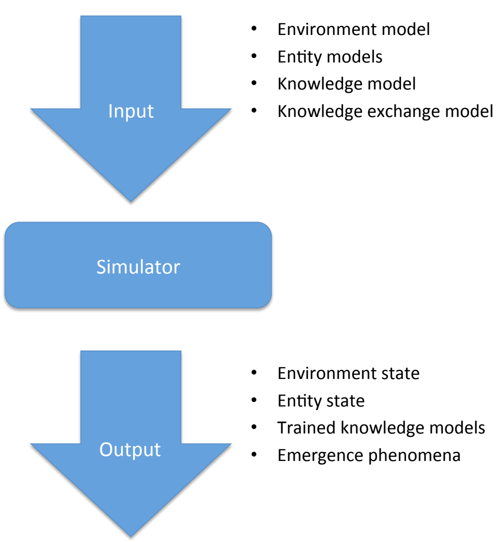

Figure 3. Simulator input and output

Environment: In essence, the environment of our simulation consists of an urban mobility system which is defined by its geographical structure, i.e. a network of streets, and an overlayed public transportation network given by a set of transportation agencies offering a number of routes. Every route in turn is defined by a set of stops and a schedule which must be executed by appropriate transportation entities. Although it is possible to use an artificially-created urban mobility system, e.g. create a random street graph and invent a suitable public transportation network, we want to develop a real-world simulation. We thus build our environment from real-world data as described in IV.

Entities: With respect to the scenario, there are currently three types of entities in our simulation:

1) **Person** This type of entity represents the persons who are travelling in the urban environment executing multimodal journeys using different means of transportation. They can walk, go by bike, use their own car, or take a means of public transportation. To obtain suggestions how to travel from their starting location to a destination, they query the multimodal journey planner offered by the *PlannerService*.
2) **Transport agency** Transport agencies are entities who monitor and control a certain set of routes of public transportation. Examples therefore are bus or train companies. They do not have a physical presence in the simulated world, but manage the observance of the schedules of their routes by assigning transport entities when required. The available transport agencies in Trento and their routes are obtained from the *DataService*.
3) **Means of public transportation** Means of public transportation are agents representing buses or trains for example. They are executing the schedule of routes they are assigned by their respective transport agency.

The behaviour of entities is modelled using a lightweight implementation of the Allow Ensembles flow concept. A flow is basically just a sequence of activities to be executed with every activity corresponding to a certain functionality. The concrete activities an entity can perform depend on its type. *Persons* for example can query the journey planner to get travel suggestions and execute them by going on foot, cycling, driving with their own car, or using means of public transportation. For buses, there is an activity to execute the current trip it is assigned by its transport agency following the geographical route and picking up persons waiting at stops.

Knowledge And Knowledge Exchange: As we will probably experiment with different knowledge models, we do not focus on a particular one or on a certain representation in this document. Instead, it is enough to say, that the simulation will provide travel statistics in form of a number of parameters which can be given to the model for learning. As we are interested in investigating distributed collaborative learning in particular, it is necessary for the model to support some mechanism of knowledge exchange and for the simulator to trigger it. From an abstract point of view, this is done based on one or more measures of closeness which can be given in various ways:

- the physical distance between two entities,

- entities being on the same bus,
- entities being connected to the same WiFi network,
- social relations, e.g. friendships

A lot of other relation types are possible. The simulator keeps track of an entity's relations updating them if necessary and triggering knowledge exchange if two entities come close to each other for example.

VI. Conclusion And Future Work

We presented a simulation of a real-world urban mobility system as an example of a CAS to investigate collaborative learning and possible emergent effects that arise from it considering a global point of view of the system. It is built on real data modelling urban traffic in the area of Trento. Based on the presented conceptual architecture, we have developed a first prototype version that is capable of simulating a realistic model of the urban mobility system of Trento including its public means of transportation. Person entities can query the journey planner and execute suggested trips collecting journey parameters. Our next steps will be the inclusion of an appropriate knowledge model to learn these statistics and to experiment with different triggers for knowledge exchange.

Acknowledgment

Partial funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement #600854 Smart Society - hybrid and diversity-aware collective adaptive systems: where people meet machines to build smarter societies (http://www.smart-society-project.eu/)

This work has been partially funded by the EU Project ALLOW Ensembles (600792).

References

[1] "Allow ensembles project homepage," http://www.allow-ensembles.eu, accessed: 2014-07-01.

[2] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012. [Online]. Available: http://elib.dlr.de/80483/

[3] M. Fellendorf, "Vissim: A microscopic simulation tool to evaluate actuated signal control including bus priority," in *64th Institute of Transportation Engineers Annual Meeting*, 1994, pp. 1–9.

[4] A. Halati, H. Lieu, and S. Walker, "Corsim-corridor traffic simulation model," in *Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and Opportunities*, 1997.

[5] D. Helbing, A. Hennecke, V. Shvetsov, and M. Treiber, "Master: macroscopic traffic simulation based on a gas-kinetic, non-local traffic model," *Transportation Research Part B: Methodological*, vol. 35, no. 2, pp. 183–211, 2001.

[6] H. J. Payne, "Freflo: A macroscopic simulation model of freeway traffic," *Transportation Research Record*, no. 722, 1979.

[7] "Openstreetmap," http://www.openstreetmap.org, accessed: 2014-07-01.

[8] "Trento mobility service," http://dev.smartcampuslab.eu/doku.php?id=core:mobilityservice, accessed: 2014-07-01.

[9] U. Wilensky, "Netlogo," http://ccl.northwestern.edu/netlogo/, 1999, center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.