POLARIS: Planning- and Operations-modeling Language for Agent-based Regional Integrated Simulations
Joshua Auld, Hubert Ley, Vadim Soklov, Michael Hope, Bo Xu, Kuilin Zhang

Travel demand, traffic flow, network operations and land-use models are typically modeled in a decoupled way, i.e. each of the components is modeled separately assuming the others are fixed. Moreover, the models are often developed by different groups for different contexts, requirement, etc. In this paper we present a prototype of a software framework which allows the user to develop an integrated simulation of a transportation system in a standardized, extensible manner, as well as an implementation of an agent-based planning and network operations microsimulation model using this approach. The project uses an agent-based modeling approach to developing an integrated transportation system model. This allows the model to overcome some of the limitations of traditional aggregated transportation models, particularly with respect to sensitivity to behavioral aspects of the travelers. The Planning- and Operations-modeling Language for Agent-based Regional Integrated Simulations (POLARIS) project, then, is intended to develop such a modeling framework and demonstrate the benefits of this approach through the implementation of an integrated travel-demand and network operations agent-based microsimulation model. The model is intended to be used in evaluating network operations improvements and ITS implementations from a planning perspective. This paper provides background on the POLARIS modeling framework and details the development of several modules using the framework which form the basis of the planning and operations simulation model.

The POLARIS modeling language “Core” serves as the foundation for the rest of the work in the POLARIS project. The core libraries are designed to provide utilities which improve the functionality across multiple applications. The first is a discrete event engine which offers automatic threading of the simulation workload as well as organize the work process by iteration or sub-iteration. Corollary libraries include a tiered memory pool allocator and several custom high-performance data structures. For HPC and inter-process operation, there is a socket-based message passing library designed to interact closely with the discrete event engine. Finally, the core libraries offer a novel suite of generic programming tools which support development of model fragments which can be easily used across multiple applications and have greatly enhanced modularity within a given application.

The POLARIS core is used to implement the various modeling components which comprise the planning and operations microsimulation model. A key aspect of these various model components is that they do not generally exist as stand-alone models, but rather are implemented as agent capabilities in the general agent-based framework provided by the POLARIS core event scheduler. The first primary component is the activity-based demand model which is implemented as a series of actions and behaviors that the traveler agents engage in during the simulation process, following the agent-based paradigm. The demand behaviors modeled include time-dependent activity generation, within simulation activity attribute
planning and replanning, and a detailed activity scheduling model which resolve schedule conflicts and maintains a consistent daily schedule for the agent. The demand components, including activity generation and activity planning are also responsive to network and traffic management events, which can result in agent replanning. The demand components implemented in the POLARIS demonstration model derive from previous work in modeling activity-planning and scheduling behaviors found in activity-based Computational Process Models.

The second primary component is the network model, which includes three parts: (1) an individual traveler’s route choice model in response to traffic information, (2) a route generation model using prevailing simulated travel costs, and (3) a mesoscopic traffic simulation model based on the Kinematic Wave theory of traffic flow. In the route choice model, travelers are modeled as individual agents to dynamically select their routes with respect to their own user characteristics in response to pre-trip and/or en-route traffic information and events from Internet, VMS, GPS, and Radio disseminated from a Traffic Management Center (TMC). The route generation model is to calculate the least time paths for individual travelers using prevailing simulated travel costs. The traffic simulation model is to simulate each individual vehicle based on the Kinematic Wave theory of traffic flow. In addition, intersection operations are also simulated for signal controls such as pre-timed and actuated signals, as well as stop and yield signs. The traffic simulation model also captures dynamic capacity reductions due to special events such as weather and accidents. The network model is seamlessly integrated with the demand model by providing pre-trip and/or en-route path information for travelers taking demand side actions, and TMC operations by publishing sensor information to the TMC for real-time traffic information provision. In turn, the demand side actions and TMC operations also impact traveler’s route choice and traffic flow pattern. The network component is implemented in the agent-based framework provided by the POLARIS core event scheduler and the high-performance computing facilities.

The initial major application of the Polaris simulation model is in the evaluation of intelligent transportation system (ITS) benefits. The performance of the transportation system and the benefits of ITS and traffic management need to be evaluated under many different assumptions. Treating weather, special events, road construction and time of day as random variables lead to a large space of possible scenario that need to be evaluated. It is unreasonable and error prone to hand craft ITS response strategies to all possible scenarios. Thus, as a part of the framework an automated Traffic Management center agent is being developed. The goal of the automated TMC agent is to monitor the status of the transportation network (speed, travel times, etc) as well as network-related events (weather, incidents, etc.) and decide on a response that would allow to mitigate unusual congestion level on the network. This aspect of the model is intended to allow planning agencies to analyze the benefits of different network operational improvements.

Finally, the project also offers several analysis tools, including a scenario manager and a visualization tool. The scenario manager is implemented as a set of software tools that allow a user to define, execute, and analyze various traffic simulation scenarios in terms of conditions
such as weather, demand, incidents, and so on. This allows the modeller to account for non-typical conditions (inclement weather, special events, incident) in evaluating the model response. Furthermore, it allows for the computation of measures of effectiveness based on the expected distribution (frequency) of simulated scenarios. These features are particularly useful for benefit/cost analysis of ITS or other transportation improvement projects, where a significant amount of the expected benefits are incurred on non-typical conditions. Along with scenario management, POLARIS also offers the Antares visualization library to allow users a simplified API for displaying their simulation results. Antares uses a layer-based drawing algorithm which can draw multiple primitive types in 3D via OpenGL or in 2D via plplot as well as navigation is 3D over those drawn objects. User geometry memory is automatically managed in concert with the discrete event engine enabling simple drawing of dynamic 3D objects. On screen identification is provided for groups or single primitives; users may interact with identification to push new geometry or attributes on selection or submit changes to on-screen objects via user-defined input windows to enable full interactivity with the underlying model.

Overall, this paper will show the utility of developing complex, integrated, agent-based microsimulation models using an extensible agent-based framework. The resulting simulation model built from the POLARIS core modeling language is unique in that it is a single, shared-memory process (or multiple distributed processes if run on a cluster) for handling all aspects of the integrated urban simulation. The resulting gains in computational efficiency and performance will allow planning modes to be extended to include such previously separate aspects of the urban system, such as network operational characteristics, enhancing the utility of such models from the planning perspective.