CHAPTER 43

Discontinued Modules

Kai Nagel and Andreas Horni

This chapter lists modules that were important for several projects in the past, but which are no longer being developed.

43.1 DEQSim

DEQSim was used for project *Westumfahrung* (Balmer et al., 2009a). It was a queue-based, event-based parallel simulation written in C++ (Charypar et al., 2007b; Charypar, 2008). This simulation included handling of reduced capacities due to traffic lights in an aggregate manner (Charypar, 2008, p.139 ff). It also supported modeling of gap back propagation at junctions (Charypar, 2008, p.98 ff).

Events were written do file by DEQSim and subsequently read by MATSim. This represented a major framework performance bottleneck. DEQSim was therefore replaced by a Java version, the JDEQSim (see Section 4.3.2).

43.2 Planomat

Chapter 45 explains how MATSim can be extended. One long-standing extension point is the `PlanStrategy` extension point (Section 45.2.9). It allows the addition of "innovative" strategy modules (see Section 4.5), above and beyond those available by default.

One such replanning model was Planomat (Meister et al., 2006; Meister, 2011). It replaced the randomizing modules for (departure) time innovation (Section 4.5.1.1) and for mode innovation (Section 4.5.1.3), with a module that computed a joint best reply for both choice dimensions internally, using a Genetic Algorithm. Thus, it evaluated not just one random alternative per iteration, as standard MATSim would do, but multiple alternatives within one single iteration, to obtain an (at least locally) optimal solution. Planomat was successfully applied in the project

How to cite this book chapter:

Nagel, K and Horni, A. 2016. Discontinued Modules. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) *The Multi-Agent Transport Simulation MATSim*, Pp. 285–286. London: Ubiquity Press. DOI: http://dx.doi.org/10.5334/baw.43.

"KTI Frequencies" for time and mode innovation for sub-tours (Balmer et al., 2010, p.10). Unfortunately, there were three interacting problem complexes with Planomat:

- Any strategy module generating best reply plans must be able to compare plans and select a better one, at least along the considered choice dimensions. This is typically achieved by giving such a module an objective function which needs to be optimized. For example, a fastest path router minimizes the travel time; a generalized cost router minimizes the generalized travel cost.

  All best reply modules here face the challenge that they cannot run the full mobsim (= network loading = synthetic reality) every time they need such information. As a result, all best reply modules are forced to build some internal synthetic reality model.

  Planomat did this by running its plans through a simplified mobsim of its own. This mobsim was a re-implementation of the most important aspects for the core mobsim. Unfortunately, however, this meant that Planomat would not automatically pick up any change or addition to the core mobsim. Consequently, Planomat's idea of a good plan often diverged from MATSim's, especially when MATSim extensions were used. In other words, any addition to the MATSim system: e.g., tolls, or opening/closing time restrictions, or differentiating link travel times by turning directions, would have to be mirrored inside Planomat.
- Planomat always tended to return the same solution: understandable from a best-reply module, but it becomes a problem when what the module thinks is a best reply starts to differ from what the MATSim core thinks.

  While an innovative strategy that deliberately generates diversity can be useful even when not fully consistent with the MATSim core (Nagel et al., 2014), this cannot function with a non-diverse innovative strategy, since it then insists on returning only suboptimal plans.
- In addition, Planomat used the MATSim core router in a way that hindered further software development of the core router. Essentially, Planomat used MATSim classes and methods that were not designed for re-use, but just happened to be public.

  It was thus an obstacle for a major MATSim core router re-design, undertaken by T. Dubernet (see Section 45.2.7).

The combination of these three issues meant that Planomat was eventually discarded: Moving it to the new router infrastructure would have entailed a major piece of one-time work. After that, maintaining Planomat's best-reply capability would have been a permanent work-intensive obligation. It was thus decided instead to invest our scarce resources in the design of a better core, allowing extensions to survive without much manual intervention. Although this will always be work in progress, Chapter 45 explains our substantial progress toward pluggable extensibility.

However, it must be noted that the improved software architecture does not resolve the general conceptual problem; best reply modules somehow need to follow core system development. Chapter 39 discusses a newer alternative that re-uses mobsim output for plan evaluation without having to run the full mobsim every time. An alternative approach, based on plan diversity, is investigated by Nagel et al. (2014). Additionally, Chapter 49 discusses aspects of diversity in plan set generation.

43.3 PlanomatX

PlanomatX was based on Planomat. It extended it by performing activity choice and adopting a Tabu Search approach (Feil, 2010). To cope with the curse of dimensionality (due to the added choice dimension), PlanomatX introduced schedule recycling, basically a warmstart concept. Because of problems when using the standard MATSim logarithmic utility function for activity choice, PlanomatX also derived an alternative utility function from Joh (2004). Rough estimates for its parameters based on an MNL exist, but turned out to be problematic, as shown in Section 97.4.3.

PlanomatX, derived from Planomat, suffered from the same maintenance problems and was eventually abandoned for the same reasons.