

The Role of Knowledge Engineering within a Fielded Planning Application

Thomas Leo McCluskey¹, Alan Lindsay¹, Mauro Vallati², Keith McCabe¹

¹ SimplifAI Systems Ltd, United Kingdom.

² School of Computing and Engineering, University of Huddersfield, United Kingdom.

alan.lindsay@simplifaisystems.com, keith.mccabe@simplifaisystems.com, lee@hud.ac.uk, m.vallati@hud.ac.uk

Abstract

This paper highlights knowledge engineering techniques that are essential to the success of a fielded planning application. The application, in the area of road transport management via the creation of traffic signal plans for urban traffic control, has been effected by the collaboration of academics, transport operators, transport engineers and software engineers. In this paper we summarise the developments leading up to this, give an overview of the system used to create the traffic plans, and describe the current benefits delivered by the system. We detail the knowledge engineering aspects in its deployment which are essential to its success, and end with a summary of the lessons learned during the process.

Introduction

Automated planning is now a well-established technology in a wide range of real-world applications, including game testing (Balyo et al. 2025), oil drilling, and cyber security. In fielded systems, domain models play a crucial role, as they need to encode the knowledge needed to support operational decisions, and to take into account the potential for uncertainty and evolving requirements. Achieving the right balance is fundamentally a knowledge engineering challenge.

This paper reports on over a decade of experience in applying automated planning to the Urban Traffic Control (UTC) application domain (Vallati 2023; Vallati et al. 2025b), culminating in a commercial system currently in operational use. The system generates coordinated traffic signal plans for urban regions, with the objective of improving journey times along strategic routes over specified time periods. While the underlying planner and simulation infrastructure are essential, our experience has shown that the decisive factor in achieving deployment at scale has been the systematic engineering of domain knowledge: how it is acquired, structured, validated, maintained, and evolved over time.

UTC is a representative example of a complex, safety-critical planning domain. It requires the integration of heterogeneous knowledge sources, ranging from relatively static structural information (such as road topology, signal stages, and physical constraints) to highly contextual and time-dependent data (such as traffic demands, turn-rate probabilities, and pedestrian behaviour). Moreover, much of this knowledge does not naturally align with the abstractions typically assumed in symbolic planning models. Bridging

this gap—between what planners can represent and what operators and engineers require—has been a recurring theme throughout the system’s development.

In this paper, we examine the role of knowledge engineering in the development and deployment of a planning-based UTC application. More specifically, We provide a detailed case study of how knowledge engineering techniques enabled the deployment of a planning-based system in a challenging real-world domain. We show how long-standing challenges in knowledge engineering, accuracy, validity, scalability, and maintainability (McCluskey, Vaquero, and Vallati 2017), were addressed through careful decisions about what knowledge to encode, what to compute externally, and what to leave implicit. Many of these decisions only became clear through sustained interaction with domain experts and operators, and through the constraints imposed by real operational environments.

It should be noted that the research has led to a breakthrough in the use of these techniques that has led to the granting of a patent GB2598661 (McCabe et al. 2020). The techniques described in the patent can be used for research purposes with the permission of the patent holder. Should the techniques be used for commercial purposes a license will be required prior to the start of commercial use.

Application Area

A significant body of academic research has been carried out in the area of road traffic management, by both transport engineers and computer scientists. Road junction control and simulation have also frequently been used as test domains for AI techniques, including optimisation and reinforcement learning. In practice, however, urban traffic signal control within groups of related junctions has long been dominated by a small number of mature adaptive systems, such as SCOOT (Taale, Fransen, and Dibbits 1998), SCATS, and related approaches. These systems adjust signal timings in response to detected traffic conditions and have proved effective in many operational settings.

Recent advances in data availability, sensing technologies, and communications infrastructure have created new opportunities for the application of artificial intelligence techniques within UTC, where such techniques are increasingly viewed as a means to improve performance in complex and congested networks (Wei et al. 2018; Smith

2020; Okrepilov et al. 2022). Among AI approaches, automated planning is particularly attractive (Vallati et al. 2016; Pozanco, Fernández, and Borrajo 2021). By explicitly modelling the scenario (the links, intersections, vehicle flows, and temporal constraints), planning approaches can guarantee that plans will satisfy certain safety criteria. They can also be used to reason about scenarios that have not yet been observed. As a general-purpose framework, planning representations are sufficiently flexible to represent a wide variety of network topologies, traffic demands, and optimisation criteria. Furthermore, there exists a mature and active research community, which provides a rich ecosystem of modelling tools and languages, and efficient solvers that can be readily applied to traffic signal optimisation problems.

The planning task considered in this paper is the generation of signal timing plans for multiple junctions in an urban region, in response to an explicit operational objective - typically to improve traffic flow along one or more strategic routes over a specified period. In contrast to fully traffic-responsive systems such as SCOOT, the emphasis here is on generating plans ahead of execution, so that predicted demand and longer-term objectives can be taken into account. At the same time, the resulting plans must be deployable on standard traffic signal controllers and operate safely under demand-dependent behaviour at run time.

Here we define the basic terminology to help us do that. We start with some (simplified) definitions. A *link* is a fixed, unidirectional piece of road ending with the stop-line of some signalised junction controlled by one signal. A *route* is a sequence of one or more links that can be travelled along contiguously by a vehicle. A *signalised junction* consists of a set of incoming links and a set of outgoing links, coordinated by traffic signals. A *stage* denotes a coordinated configuration of signal aspects at a junction, while a *phase* refers to the duration for which a particular movement, or set of movements, receives a green signal. A *cycle* is the time between successive occurrences of a designated stage. A *region* is a connected set of junctions, where each junction is connected to at least one other in the region; junctions may be internal or act as boundary sources or sinks of traffic. A *fixed signal plan* for a junction specifies a cycle time together with a sequence of stages and nominal stage durations. Such plans typically combine forced stage changes, which must occur in every cycle, with demand-dependent behaviour. In particular, a stage may be entered only if demand is detected (e.g., by vehicle or pedestrian sensors), or may terminate early ("gap-out") when demand is absent. As a result, although a plan defines a structural pattern, its execution depends in part on real-time conditions. A *deployable plan* is one that satisfies all operational and safety constraints imposed by traffic signal controllers and can therefore be implemented on-street.

Finally, where vehicles approaching a junction may select between alternative outgoing links, *turn-rate probabilities* can be estimated from historical data corresponding to similar dates and times. These probabilities are used to predict how traffic will distribute across available movements and form an important part of the contextual knowledge required by the planner.

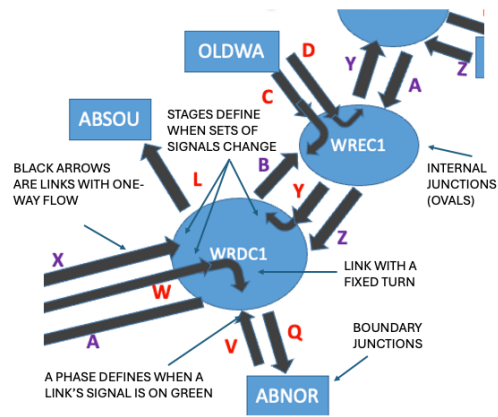


Figure 1: Basic definitions in road management

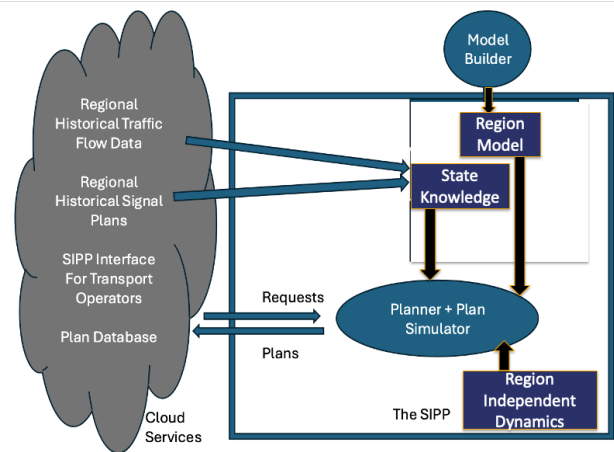


Figure 2: A simplified overview of the SIPP

Plan stages start with an *intergreen phase* which covers the time between when the lights specified in the last stage turn to red, until the specified signals in the next stage turn to green.

The conventional method in UTC to deal with the problem of traffic signal control is for engineers to create fixed signal plans as specified above. These are often crafted using a tool such as TRANSYT (Robertson 1969). On the other hand, popular approaches applied to critical areas of urban regions are demand responsive systems, such as SCOOT, which change stage times in response to real-time flow readings, in order to service higher demands. These can be in addition to the demand-driven and gap-out behaviour described earlier regarding a fixed plan.

SIPP: The Simulation and Planning Platform

Before discussing the knowledge engineering issues, in this section we outline the planning and scheduling sub-system, called the SIPP (Simulation and Planning Platform) which sits at the heart of SimplifAI's deployed system. The approach is covered by SimplifAI's international patent (Mc-

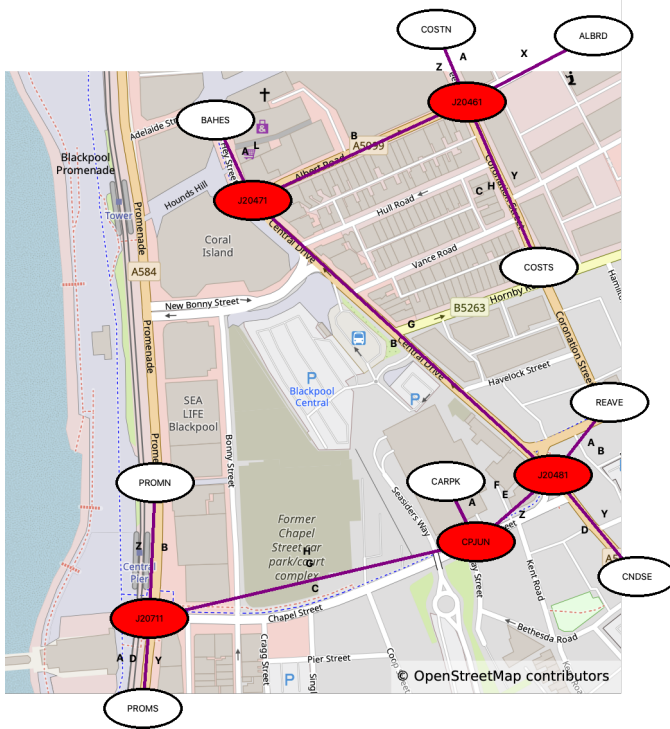


Figure 3: A schematic of a UTC region. Map data © OpenStreetMap contributors.

Cabe et al. 2020) which includes the central idea of using a goal directed planning method within a model-based application description for the UTC application area. The system is deployed via cloud services which contain the user interface to the SIPP, and the interface to provide data for the SIPP. An overview of SIPP is shown in Figure 2.

Traffic regions that the SIPP has been applied to are of similar size to a conventional SCOOT region, approximately 4 - 10 internal junctions clustered together. A diagram of an example region is shown in Figure 3. Here the red nodes are internal junctions, and white nodes denote boundary junctions. The traffic region could form any shaped graph (e.g. cyclic), but must have boundary nodes to delimit the region. To operate the system in a region, a corresponding model is built using the Model Builder, which outputs the “Region Model”. Historic flow data and signal plans are made available to become part of its “State Knowledge”. These knowledge bases are then integrated with the region-independent Dynamics Model, which is written in PDDL+ and does the job of an abstract, macroscopic traffic flow simulator.

A transport expert or operator forms a query by specifying in the SIPP Interface a date, time and duration, and a goal for the system to achieve. A goal, in its most general terms, has two parts:

- Part 1 - the objective: the desired maximum average journey time, or the minimum average speed, along a route(s) within the region
- Part 2: - the constraints: a limit on the journey time of

each of a set of routes through the region.

Part 1 captures the need to improve the flow through some of the key routes in the region. This can be used to capture the need to speed up the flow during rush hour, or to speed up the flow before or after a major event in the city centre (a football match, a large gig, a set of road works). If we can predict that a large number of vehicles will need to leave a city centre at a particular time, then this demand can be part of an initial state to construct a plan alleviating the extreme delays that will otherwise happen. Of course, criteria for two or more goals can be conflicting, and the planner aggregates the multiple objective, attempting to balance the requirements of possibly opposing goals.

Part 2 captures the need to keep the flows *across* (rather than along) the key routes from increasing too much, so as not to inconvenience other road users.

The SIPP produces a set of deployable plans which constitute a traffic signal plan for every junction in the region, such that when the junctions are given these plans to run, the goals will be met in simulation. Hence, the SIPP supports “goal directed” planning – in the sense that the goal is an implicit property (journey time) to be optimised over one or more routes.

As an example, let us consider the region shown in Figure 3. In this area, operators have noticed recurrent congestion meaning that the average journey times through several key routes between 4-5pm are too high, typically about 2,000 seconds, or an average of about 7 minutes per route. It has been decided that a plan set is required with an average journey time of less than 5 minutes per route. The cross flows are also specified and monitored, so that increases to their journey times are kept under certain limits. The SIPP is invoked in a mode where a fixed cycle time is given (e.g., 120 seconds). This operation takes approximately 90 seconds (on a standard laptop) to output a set of deployable plans that operators can use to address the congestion.

Use of the SIPP

We have modelled and used the SIPP on 14 regions in 5 different local authorities around the UK. The most complex region contains 26 junctions (16 boundary junctions and 10 internal), 9 pedestrian crossings and 45 links.

To engineer knowledge for a new region from scratch to operational use can take 1 - 5 working days for a knowledge or transport engineer, depending on the region’s current traffic management system and the availability of traffic and traffic signal data. For SCOOT regions, the amount of resource required is at the lower end, given the availability of parameters, and our experience with this method.

As referred to above, for the SIPP to create new plans within deployed regions takes around 1 - 5 minutes processing time, either in the Cloud or on a Laptop. The processing time taken depends on the size of the region (number of junctions, number of links, duration required). On account of the efficient, greedy heuristic search mechanism of the planner, we have found the increase in time, during operation, to increase in approximately linear time to these size metrics. An advantage of using a structured and automated approach

to knowledge engineering of regional models is that if two or more regions are adjacent, they can be composed into one connected region with minimal effort.

The first SIPP-generated set of plans was used in the Kirklees area, to manage the traffic flow out of the Green Day Gig in June 2022, from the local John Smith Stadium. As well as large event management, the SIPP has been used to generate plans for rush hour to alleviate traffic congestion. Some of these SIPP generated plans used for rush hour peak periods have been continuously operational for over a year.

An independent evaluation of the benefits SimplifAI's system has made to traffic management, and the potential benefits of its future use, was carried out by Nota Bene Consulting. The evaluation report concludes that in "a comparison of Before and After journey times on [Region A] and [Region B] corridor, the results presented in this section provide the clearest evidence of SimplifAI's effectiveness in optimising signal plans", and again, "SimplifAI is a well-conceived and credible traffic signal optimisation platform that delivers real, measurable benefits when applied in appropriate contexts". Their indicative results suggests that the improvement to journey times through regions can range up to 50% in applications to recurrent congestion, and a corridor where the SIPP is implemented can save thousands of pounds each year in savings on journey times (NotaBene 2025).

Overview of Knowledge Engineering Approach

Designing and developing the components of a system which form the inputs and outputs of a planning algorithm and together form a commercially successful system, will in general, require a great deal of time and effort.

We start by attempting to break down the components of what is needed. Through the treatment we will use the Traffic Plan-generation application as an example.

Dynamics Model

Capture of the "physics" of the application famously seems to be front and centre when developing domain models for research purposes or for evaluating a planner. A key decision to make is what is an adequate representation for the parts of an application that change as a result of deliberate acting or of an exogenous event. In our case the main dynamic parts are the flows of traffic along road links, and the change of traffic signals at junctions. This demanded the use of more expressive forms than actions, hence we chose PDDL+ as the basic language. The domain dynamics of the SIPP is captured in a PDDL+ model derived from one published about 10 years ago (Vallati et al. 2016). The original model can be summed up as "bucket" dynamics in that traffic from one link flows into another continuously, but once in a link can be assumed to be ready to transfer to the next link. It is easy to notice that is an approximation in that the vehicles take no time to travel along a link. We have developed the dynamics to incorporate more expressive features, such as the ability to take into account the time traffic takes to flow along a link. Thus the resulting traffic *simulation* via PDDL+ is

very efficient, but also adequate enough to be able to predict journey times of flows with new signal plans accurately (Doria et al. 2026b). So while Google, for example, is good at predicting journey times with existing conditions, the SIPP predicts them when arbitrary signal plans are implemented in the region. It is however important to notice that within SIPP, journey times are not explicitly represented within the PDDL+ model. They are instead calculated by using some dedicated external components, that leverage on PDDL* for simulating traffic behaviours. The problem of incorporating journey times within PDDL+ is still an open challenge.

In contrast, other researchers in the UTC Planning field have made different decisions on representing model dynamics for planning applied to UTC. For example, Pozanco, Fernández, and Borrajo (2021) use a discrete rather than continuous PDDL model: this is much more efficient though may not be as accurate as the PDDL+ representation. Answer Set Programming has also been used to represent simplified model dynamics, and address the UTC problem in a similar planning-based fashion (Tarzariol, Maratea, and Vallati 2025). Capturing model dynamics highlights the adequacy vs efficiency tradeoff often seen in knowledge engineering. Since our flow model does not explicitly simulate traffic flowing along a road link, it is implicitly inferior to many traffic simulators with explicit simulation of flow; however, the model is efficient enough to be used in plan generation. Furthermore extra knowledge can be added to enhance the modelling accuracy of this restriction. For example, assuming traffic keep to speed limits, a lower bound can be put on the journey time of traffic flowing along a link.

Modelling Uncertainty and Non-determinism

These two factors must be considered in planning applications as they are aspects of reality that need to be taken into account in predicted state changes and action effects. There are planning languages and planners which are capable of representing these aspects to some degree, hence the question arises whether to model them explicitly and allow a planner to reason with them. Our experience indicates that for complex domains it is best to deal with these "outside" of plan reasoning, for (a) plan generation efficiency reasons - incorporating these aspects into the plan generation process makes for a more complex search space (b) expressiveness - the limits of expression for these aspects are determined by the planner, which is likely to limit the adequacy of the representation.

For the UTC application, non-determinism is found in the actions that cause a signal change from state to state: whereas the main state changes can be determined a priori, *demand-driven* stage changes can only be determined at the time of execution. Some states may not be demanded and hence not taken up, and some may finish prematurely if there is no demand ("gap-out"). Our solution here is to use the planner to create abstract contingent plans angled to solve the current goal. In the search for the best plan these candidate plans are evaluated with respect to their performance using historical data which is taken from a similar past time period. The historical demand pattern and flow rates create a concrete plan which then can be evaluated deterministically.

Uncertainty in applications is clearly a major challenge, since action or process effects can only ever be approximated. Again, representing this uncertainty explicitly for reasoning within a planner may diminish unacceptably the planner's efficiency and effectiveness. In the UTC application, one uncertainty is the choice of out-link traffic take when entering a junction via an in-link (the turn-rate probability). We used a combination of historical data, in-link knowledge and constraints reasoning to estimate these probabilities for each in-link of each junction. This is done just prior to planning as part of the initial state compilation. Another uncertainty is the flow rates into the traffic region over a duration (the "demands"). Again, we can use historical and situational data to estimate these over a particular duration.

Long-term Static, Structural Knowledge

So-called "static" knowledge is likely to be the largest component of a real application. This will contain all the structural knowledge required for the plan generation function that remains unaffected by actions or events during any plan generation run. The amount of knowledge is likely to expand over the development phase, as more understanding of what users require is made explicit to the developers. There will be decisions to be made about what knowledge has to be explicit, what knowledge will be created for plan generation, and what is important but already implicit in existing knowledge.

In the UTC example, static knowledge includes connections, capacity and lengths of the road links; and stages, stage length constraints, allowed turns, maximum exit flow rates, and signal-stage mappings, for junctions.

An example of the implicit/explicit knowledge capture tradeoff is whether to represent the gradient of road links leading up to a junction, which have a bearing on the average flow rate across a junction. For efficiency, we left this value implicit within a pre-calculated flow rate value for each link end, though this may miss changes to flow rate in icy conditions. Future planned incorporation of weather data into the SIPP will cause an expansion of the structural knowledge to represent the effects of different weather patterns.

Configuration-driven State Knowledge

The compilation of knowledge for a particular planning run will include the initial state of knowledge that changes via plan execution, and unchanging knowledge that can only be assembled when each call is made. In the UTC case, examples of the former include traffic concentrations and states of each signals. Examples of the latter include turn-rate probabilities and flow rates along input links (the demands). Whereas we treat the demands as unchanging given a particular plan, the rates of course themselves can change over a duration that the plan operators (the expected flow rates over a duration can change in a stepwise manner in our implementation, to model the historically expected flows).

Initially, we anticipated that real time information would be available from sensors in the road network to allow plan generation in close to real-time (Bhatnagar et al. 2022b). Unfortunately, in the urban areas we were aiming to deploy, neither the flow rates were readily available, nor the turn

rate probabilities for each traffic flow. We found the solution to these problems with the use of historical data - traffic flow and turn rate probabilities for the current scenario are estimated from the similar historical scenarios. One or more periods - date/time/duration - in history are selected where conditions are close to the desired date/time/duration in the future that the plan is required for. A fixed time plan is extracted from the historical plan, together with the take-up of demand-dependent junctions, the turn rate probabilities, the link saturations etc. The latter forms the initial state in the search for a plan to solve the goal at the future date.

Goal Structure

What constitutes a goal for a particular planning run has to be linked explicitly to the users' requirements. In a benchmark or competition domain model created to test a generally applicable planning engine, typically this means a collection of facts that appear in the post condition of an action specification. Tying this up to what is useful for a particular application may well be the biggest challenge in creating the system.

In the UTC case, the vague goal of "minimise delay for traffic while preserving safety" is often taken as the task that transport operators must aspire to; it is what signal plans are aimed at, and forms the basis on which the traffic signal hardware is designed. Finding a goal that is both useful to the application problem owners (here transport operators) and one which can be solved by automated planners amounts to looking for some goal which falls between two extremes: a concrete expression in the planning language containing conditions, (its initial design influenced by the need to efficiently generate signal plans) vs a service goal such as minimising delay which cannot readily be put into planning-state terms.

Examples of goals that could be used include to maintain the average speed above a certain level for a particular key strategic corridor, or to clear the traffic from a major event within a certain time-frame. Existing UTC systems do not work towards such explicit specific goals, and it can be difficult to enable the operators of such systems to set such goals.

Initially, our research focused on goals which were combinations of explicit conditions in the PDDL+ model discussed in Vallati et al., such as the occupancy of a link (McCluskey and Vallati 2017). We trialled the initial versions of the UTC planning system in areas of the city of Manchester using goals which were "numerical expressions corresponding to saturation levels on sets of links" (Antoniou et al. 2019). In a more recent publication, a goal is defined as the number of vehicles that are required to travel through some links of interest in the region (Bhatnagar et al. 2022a). Hence goals expressed the idea of keeping the level of traffic on a link below a certain level or enabling a certain number of vehicles from a link in a defined period. This work was extended to incorporate more expressive versions of the domain representation, of the goal language, and improved heuristics (Percassi et al. 2023; El Kouaiti et al. 2024; Castellanos-Paez, Percassi, and Vallati 2025; Doria et al. 2026a; Chiari et al. 2026).

Our experience has shown, however, that these goals can

be rather too specific for transport operators to grasp and use. A less abstract goal, and more in tune with traffic management, is required, and more promising is a goal specification utilising the average journey time over a time period, or more or less equivalently, a minimum average speed, along a route or a set of routes through a region. The challenge here is that these goals are implicit properties of a plan rather than an explicit state condition, hence calculation of the distance to them to guide planning is not easy.

In our current implementation the planner uses greedy best-first search through a space of abstract solution plans, and progressing the most promising solution plan by evaluating the implicit properties of these plans (here specifically the journey time of routes).

In summary, the distance between goals/tasks that are useful in an application area (ones whose solutions might assist an operator), and the goals/tasks that a planner can deal with, is a crucial measure. In our case, this distance did not become clear until we involved the transport operators of the current system closely with the SIPP's development.

Knowledge Engineering and Safety Issues

Delivery of a service in the public sector, or the production or management of operations in the private sector, are subject to varying amounts of rules and regulations to ensure safety and security. For example in the UTC area, safety concerns include making sure traffic signals do not lead to vehicle collisions because of inconsistent signals, whereas security concerns include the issue of a malicious third party gaining control of the signal operation.

In such circumstances, it is essential to document the current levels of safety, and where the planning function fits into those levels. Often the safety rules can be encoded explicitly in the engineering process, and hence available for use in behaviour explanations.

In the case of UTC, the safety and security falls into various distinct levels:

- (i) our own software for checking that the plans produced met any safety criteria specified. This is supported by the use of explicitly stated safety rules in the static knowledge;
- (ii) legacy software that checks the integrity of any plans loaded into UTC hardware;
- (iii) in-built safety checks within traffic signal hardware. Hence, any plan the SIPP generates has an extensive series of safety checks, and opportunities for bug checks, before going into operation.

Currently, many authorities enforce manual-loading of plans into UTC hardware, meaning that the system is not fully automated. This supports security in terms of the prevention of malicious virtual agents, as it precludes access online, and an additional level of validation.

Automating parts of the KE process

It is essential that the initial acquisition, and subsequent maintenance of the state (dynamic and static) model is as automated as possible, given its likely size and complexity.

To be able to roll out the planning function to different scenarios (in UTC, this means different physical regions), there needs to be a tool-supported method of creating this knowledge from external knowledge sources. This is crucially important, as the construction of these models (as recognised in the KEPS literature) is often described as the weak point of symbolic approaches. One way to assist this is to export data from a currently used system which is being superseded. Another is to encode a set of meta-constraints or axioms that must be true of the model of every scenario/region; these can be encoded in the model/state building tool to help in the validation of the static knowledge.

In the UTC case, data from an existing UTC system can be captured and modelled for processing by the SIPP with little effort, given a set of adaptors we have created to form PDDL-like expressions using an API. Utilising the structure and environmental parameters from the existing solution is most efficient in order to create the domain model. In our initial work, when deploying the system to replace SCOOT during some periods of operation, we automated the distillation of parameters within the traffic region from the SCOOT system, and their translation into a PDDL model. Access to operational SCOOT data was either online or within machine-readable documents, hence the build up of the domain model is largely automated.

Recently we have expanded the use of the system onto non-SCOOT regions, hence a more general knowledge engineering approach is used: (i) an operator-friendly spreadsheet template is used as the front end to a build program which generates the permanent knowledge of the application. The build program will have access to the meta-constraints above to help ensure the validation of the built model (ii) an API is specified for the ingestion of all historical data. We have investigated using state of the art LLMs and RAG techniques in domain model creation, in order to make this interface more efficient, but we found that the method is limited by the very nature of how the LLMs are constructed (Vallati et al. 2025a).

Unexpected Changes During Development

The path to deployment and usefulness to a particular sector was in our case a long process stretching over 10 years. Inserting an AI Planning function into a well developed major service, with existing well-established suppliers who have a vested interest in preserving the status quo, proved not to be easy. The fundamental difference in our solution to UTC was not only the idea of goal-directed plans, but also the idea that plans for a duration (typically 1 hour) were to be generated before being executed. Our initial work into traffic signal plan generation used a range of generic PDDL+ planners such as ENHSP (Scala et al. 2020). This proved efficient and effective for finding plans tuning up to 60 minutes to satisfy PDDL+ conditions, with some domain specific heuristics added (Franco et al. 2018; Doria et al. 2026a; Percassi et al. 2023). Over time, however, the idea of goal achievement based on a final state changed to one of a goal as a property of a whole plan, giving rise to the use of a plan-space searching planner.

As is fairly typical in planning, initially we assumed that all the plans could be generated from scratch. A solution plan in UTC is more accurately stated as a set of plans, one for each junction in the region, hence we have the additional complexity of the concurrent execution of each plan. Typically with UTC traffic responsive solutions, if the method (such as SCOOT) is suspended in a region, then the alternative method needs to generate plans for *every junction*, even if some of the junctions were not involved in goal achievement. This consideration, and the complexity of generating a plan for each junction from scratch, led us to decide to use a planning method that generated a goal plan from an already complete plan, one generated by the previous UTC system - in other words *derived from history*. The SIPP has access to a pre-used set of plans (PO) for a region in the preceding weeks and months, and associated historical data including traffic demands and traffic plans, obtained via its cloud interface. The historical plans may be generated by the previous control mechanism (e.g. SCOOT), but as the system is deployed over a period of time these historical plans may have been SIPP-generated. An old plan is chosen if it fits in with the current environmental data - if it was generated under the same conditions - for example a plan needed at 8am on a schoolday morning will retrieve an old plan from similar circumstances. The planner then works by changing this old plan incrementally to solve the goal(s) given to it by the transport operator. This is superior to planning from scratch, it can give us reasonable plans for junctions not involved in goal achievement, and it also has the benefit of capturing (initially at least) desirable properties of traffic plans that would be difficult to generate. Primarily, it would be expected that the historically-based plan would be structured to capture a desired offset for adjacent junctions in the region¹ an indication of the behaviour of demand-dependent junctions and the frequency of pedestrian crossing use.

Lessons Learned

In this section we summarise some of the lessons learned in the process of engineering the knowledge making up an automated planning application for urban traffic control. Firstly, we refer to a paper published in 2017 “Embedding automated planning within urban traffic management operations” (McCluskey and Vallati 2017) in ICAPS by the authors which described the use of automated planning in UTC to create signal plans for use within a state of the art traffic simulators (AIMSUN and SUMO). The major lesson learned here is that it took over 5 years to develop the system from a traffic simulation to its use on-street. The effort was taken up by knowledge engineering of the final system, and the creation of a domain-specific planner. There was also a not insignificant amount of effort in software engineering in

¹An offset is the adjustment of cycle start times between adjacent junctions that favours the flow in a particular direction - notably along the main corridor of a traffic route. Its effective operation means that a platoon of traffic proceeding along a corridor would approach the next junction as the the signal turns green. Of course the offset idea is limited to certain directions in a region, and also to the tight connections of junctions.

order to provide online access to input data, and a route to output plans and reports to users. It is interesting to contrast the lessons learned then with the final development of the system now - most of which seem to be the result of (i) using a simulation-based system (ii) the use of general planning technology (iii) a lack of understanding at the time of the form of deployable plans. Secondly, here we list our revised lessons learned from the period of final implementation:

- We recommend using a range of different encodings and plan languages to prototype the application. As application knowledge is acquired by the development team, the merits of each different encoding will become clearer over time.
- Be ready to deal with tricky aspects of the application (e.g. non-determinism) outside of the planning process. In other words, do not try to include every feature of the real problem in the solver and focus on the core aspects in optimization.
- Utilise the existing technology and historical records in the application which automated planning is going to supersede wherever possible (e.g. to provide initial state values, to provide initial plans from which start the planning process).
- From the start of the project, it is crucial to understand the operational environment in which the planning system will operate, in terms of technological, functional, and interaction constraints and characteristics.
- From the start of the project study in detail what the concrete form and function of the required plans will be, and acquire those details from the operators who currently implement the plans.
- From the start of the project, study in detail what goals the planner needs to solve to be of greatest use to the stakeholders, and how the plans detailed in the point above this one can be evaluated in terms of these goals (as the goals may not be explicit in a state).
- Build up a range of tools, which embody the axioms/meta-constraints common to all scenarios, to automatically create both configuration-dependent state knowledge, and the static structural knowledge. This process can be repeated on maintenance and scenario re-use, making the results of the process less prone to errors.

Conclusion

We have outlined the development of a system which, over the years, has sought to use AI Planning techniques to generate traffic signal plans for use by transport operators in urban traffic control applications. We describe the SIPP that now is part of a commercial system working for local authorities in the UK. We then describe the steps in the knowledge engineering process that cover the development of the various types of knowledge used within the planning and simulation system. Finally, we detailed some lessons learned in the knowledge engineering process, which we hope will inform future deployment of planning technology.

References

- Antoniou, G.; Batsakis, S.; Davies, J.; Duke, A.; McCluskey, T. L.; Peytchev, E.; Tachmazidis, I.; and Vallati, M. 2019. Enabling the use of a planning agent for urban traffic management via enriched and integrated urban data. *Transportation Research Part C: Emerging Technologies*, 98: 284–297.
- Balyo, T.; Barták, R.; Chrupa, L.; Červenka, M.; Dvořák, F.; Gocht, S.; Lipčák, L.; Macek, V.; Roháček, D.; Ryzí, J.; et al. 2025. Using planning for automated testing of video games. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*.
- Bhatnagar, S.; Guo, R.; McCabe, K.; McCluskey, T. L.; Scala, E.; and Vallati, M. 2022a. Leveraging Artificial Intelligence for Simulating Traffic Signal Strategies. In *Proc. of ITSC*, 607–612. IEEE.
- Bhatnagar, S.; Mund, S.; Scala, E.; McCabe, K.; and Vallati, M. 2022b. On-the-Fly Knowledge Acquisition for Automated Planning Applications: Challenges and Lessons Learnt. In *Proc. of ICAART*.
- Castellanos-Paez, S.; Percassi, F.; and Vallati, M. 2025. Improved Models for Automated Planning-based Urban Traffic Control. In *International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*.
- Chiari, M.; Percassi, F.; Gerevini, A.; and Vallati, M. 2026. Near-Future Traffic Forecasting for Planning-based Traffic Signal Optimisation. In *Proceedings of the Thirty-Sixth International Conference on Automated Planning and Scheduling*.
- Doria, F.; Percassi, F.; Maratea, M.; and Vallati, M. 2026a. A Domain-specific Heuristic for PDDL+-based Traffic Signal Optimisation. In *The 40th Annual AAAI Conference on Artificial Intelligence*.
- Doria, F.; Percassi, F.; Maratea, M.; and Vallati, M. 2026b. Traffic Signal Plans Explorer: A General Framework for Visualising Traffic Evolution. In *40th Annual AAAI Conference on Artificial Intelligence (AAAI'26)*.
- El Kouaiti, A.; Percassi, F.; Saetti, A.; McCluskey, L.; and Vallati, M. 2024. PDDL+ Models for Deployable yet Effective Traffic Signal Optimisation. In Bernardini, S.; and Muise, C., eds., *Proceedings of the Thirty-Forth International Conference on Automated Planning and Scheduling*, 168–177.
- Franco, S.; Lindsay, A.; Vallati, M.; and McCluskey, T. L. 2018. An Innovative Heuristic for Planning-Based Urban Traffic Control. In *Proc. of ICCS*, 181–193.
- McCabe, K.; McCluskey, T.; Vallati, M.; and Blackwood, J. 2020. Traffic Strategy System and method of implementing the same. In *Patent GB2598661*.
- McCluskey, T.; and Vallati, M. 2017. Embedding Automated Planning within Urban Traffic Management Operations. In *Proc. of ICAPS*, 391–399.
- McCluskey, T. L.; Vaquero, T. S.; and Vallati, M. 2017. Engineering Knowledge for Automated Planning: Towards a Notion of Quality. In *Proceedings of the Knowledge Capture Conference, K-CAP*, 14:1–14:8.
- NotaBene. 2025. *independent Evaluation of Simplifai solution*. InnovateUK.
- Okrepilov, V. V.; Kovalenko, B. B.; Getmanova, G. V.; and Turovskaj, M. S. 2022. Modern trends in artificial intelligence in the transport system. *Transportation Research Procedia*, 61: 229–233.
- Percassi, F.; Bhatnagar, S.; Guo, R.; McCabe, K.; McCluskey, T. L.; and Vallati, M. 2023. An Efficient Heuristic for AI-based Urban Traffic Control. In *Proc. of MT-ITS*, 1–6. IEEE.
- Pozanco, A.; Fernández, S.; and Borrajo, D. 2021. On-line modelling and planning for urban traffic control. *Expert Syst. J. Knowl. Eng.*, 38.
- Robertson, D. 1969. Transyt: A Traffic Network Study Tool. In *RRL Report LR 253, Road Research Laboratory Crowthorne, Berkshire*.
- Scala, E.; Haslum, P.; Thiébaux, S.; and Ramirez, M. 2020. Subgoalting Techniques for Satisficing and Optimal Numeric Planning. *J. Artif. Intell. Res.*, 68: 691–752.
- Smith, S. F. 2020. Smart Infrastructure for Future Urban Mobility. *AI Magazine*, 41(1): 5–18.
- Taale, H.; Fransen, W.; and Dibbits, J. 1998. The second assessment of the SCOOT system in Nijmegen. In *IEEE Road Transport Information and Control*, 21–23.
- Tarzariol, A.; Maratea, M.; and Vallati, M. 2025. A CASP-Based Solution for Traffic Signal Optimisation. *Theory and Practice of Logic Programming*, 25(4): 794–812.
- Vallati, M. 2023. The Power of Good Old-Fashioned AI for Urban Traffic Control. In *International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications*, 3–10.
- Vallati, M.; Barták, R.; Chrupa, L.; McCluskey, T.; and Petrick, R. 2025a. Knowledge Engineering for Planning and Scheduling in the LLM Era. In *Proceedings of the Thirty-Fifth International Conference on Automated Planning and Scheduling*, 391–395.
- Vallati, M.; Dodaro, C.; Doria, F.; Fiorentino, S.; Maratea, M.; Percassi, F.; and Tarzariol, A. 2025b. ASP and PDDL+ Applications in Urban Traffic Distribution and Control. In *Joint Proceedings of the Workshops and Doctoral Consortium of the 41st International Conference on Logic Programming*.
- Vallati, M.; Magazzeni, D.; De Schutter, B.; Chrupa, L.; and McCluskey, T. 2016. Efficient Macroscopic Urban Traffic Models for Reducing Congestion: A PDDL+ Planning Approach. In *Proc. of AAAI*, 3188–3194.
- Wei, H.; Zheng, G.; Yao, H.; and Li, Z. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2496–2505.