Knowledge Engineering for Planning and Scheduling in the Context of Ontological Engineering: An Application in Railway Rolling Stock Maintenance

Thomas L. McCluskey, Hassna Louadah, Emmanuel Papadakis, Gareth Tucker, Adam Bevan and Peter Hughes

School of Computing and Engineering, University of Huddersfield, UK.
{ t.l.mccluskey, h.louahah, e.papadakis, g.tucker, a.bevan, p.hughes} @hud.ac.uk

Abstract

In this paper we consider the opportunities for KEPS within wide-spectrum projects which are aimed at creating precise ontological models of areas of industry. The goals of such projects are wide ranging, as is the related areas of enterprise modelling. Particularly in industrial applications, the benefits of ontological modelling are not only seem as improving human-human communication, but also in process analysis and animation, and specifically in the use of automated reasoning to do goal-directed planning and scheduling. Applications of AI planning within such integrated applications range from optimisation of processes and process scheduling, to the use of planning in automated manufacturing and robot manipulation. We survey the past work which relates to such endeavours, and illustrate our approach and motivation using an ongoing major Case Study - that of capturing knowledge structures within a railway depot maintenance operation for various purposes including automation.

Introduction

Knowledge engineering for planning and scheduling (P&S) applications, where that knowledge is part of a much wider scale knowledge engineering effort, is an important sub-area of KEPS. The community recognised the area in terms of utilising shared ontological knowledge in the 2005 workshop "The Role of Ontologies in AI Planning and Scheduling" [1] and the beneficial fusion between planning and description logic has been long recognised (Gil 2005). In the intervening period several planning applications have been made in this context, for example, in business applications (Bouillet et al. 2007) and space operations (Bonasso et al. 2013).

The aspirations of projects aimed at creating precise, integrated, ontological models of areas of industry or enterprise are wide ranging (as in the related areas of enterprise modelling (Fox and Grüninger 1997)). Particularly in industrial applications, the benefits of this modelling are not only seen as improving human-human communication, but also in process analysis and animation. This kind of formal modelling also can lead to applications of P&S within such integrated applications, with uses ranging from optimisation of processes and process scheduling, to the use of planning in robot manipulation. We survey the past work which relates to such endeavours, and introduce a major Case Study in this area - that of capturing the knowledge structures within a *railway depot maintenance* operation.

In many applications of P&S, the essential knowledge components making up the planning domain model are taken cleanly out of the their context so that they fit a Planning Domain Definition Language (PDDL) like language. The knowledge and data that are synthesised into a domain model are very often not already formalised, hence basic relational and factual knowledge needs to be crafted. Benchmarks such as those used in planning competitions tend to include the bare minimum knowledge to enable solutions to be found to expressions in the goal language. In contrast to this, there are growing efforts in developing conceptual models of enterprises, drawing on the benefits of ontlogical modelling and knowledge sharing (such as in Industry 4.0 (Gocev, Grimm, and Runkler 2018)). In these cases, knowledge needed for a planning domain model may be available in a formal, most likely ontology, language. Given the potential uses of P&S, particularly in the industrial sector, the question arises: how should we develop and *knowledge engineer* domain models and problem files in the context of such rich ontological modelling? Can we create a systematic method to support knowledge engineers in the acquisition, validation and maintenance of planning domain models from such ontological models?

This paper makes the following contributions:

- it surveys and summarises work (since the 2005 ICAPS workhop on Ontologies and Planning) which regards planning as an activity embedded in an environment that already contains a high level of formalisation in the form of ontological models within a wider area of industry or commerce;
- it describes a novel application area for P&S - that of monitoring, managing and enacting the maintenance of railway rolling stock such as train carriages and bogies - and a project which seeks to model this area as a set of integrated ontologies;
- it illustrates an approach to creating planning domain knowledge from developing ontologies, in particular the translation of the Semantic Web Rule Language (SWRL)

[1] https://ccia.ugr.es/ faro/workshop/Workshop.htm

rules to operator pre and postconditions. We describe this within the railway rolling stock application (the Case Study) with a detailed look at the ontological models of maintenance requiring brake pad inspection for railway vehicles.

Ontology-supported KEPS

In Table 1 we summarise a representative sample of practical applications and tool development carried out since the 2005 ICAPS workshop on the role of ontologies in P&S applications, in chronological order. These works all involve ontological modelling of an application area within which there is some requirement for a P&S function. The "Domain" column indicates the domain or application area that inspired the development; the "Translation" column indicates whether a translation is used between the ontology language and a planning language in order for an external planner to be used. *Integrated* indicates that the plan generation procedures or planning-related procedures are defined within the ontology language itself, and therefore no external planning engine is used. The "Comment" column relates some characteristic features of the work.

One of the first works cited in the table is also one of the most impressive: researchers at IBM utilised an ontology-oriented view of the knowledge engineering and planning functions for workflow generation (Bouillet et al. 2007). Rather than developing a domain ontology separately, and creating a bridge into the use of standard planners (e.g. via the assembling of PDDL models), they keep the planning function within the OWL sphere, modelling actions as RDF transformations (the paper includes a formal definition of an action in RDF notation) and goals as RDF graph patterns. The authors declare a range of advantages for this approach in particular domain model construction and validation, ontology reuse and sharing, and the ability for a team of engineers to build up domains in a modular fashion. For knowledge engineering in particular, the work emphasises the opportunity for experts of various types (not just planning experts) to work on the domain model knowledge, as well as diagnostic tools and the opportunities that knowledge sharing brings. They highlight one of the challenges with planning within the ontology in terms of efficiency - for example the need to check very efficiently whether a predicate is achieved at a point in a plan is less efficient than in a PDDL-driven approach.

Other works which use an integrated planner within an ontology-based language include Cioffi and Thompson's work on creating a graphplan-like planner aimed at service composition problems (Cioffi and Thompson 2006); and Celino et al's use of the Open Provenance Model Profile to capture planning domain model concepts, and utilise the Ontological Framework in particular to do correctness and consistency checks of the planning knowledge (Celino and Dell'Aglio 2012).

The approach in most of the applications cited is to develop formulations of ontological knowledge independently of the planning function, and then create a translator to package up the relevant knowledge from the ontology into an external planner-friendly formulation. In this vein, Liu et al explain the advantages and importance of ontological modelling in emergency action planning applications, and show how to integrate this knowledge with Hierarchical Task Network knowledge for use with the SHOP2 hierarchical planner (Liu et al. 2013). The "translation" route is also taken in an Intelligent Transport System application (Feljan et al. 2017), in an approach to creating foundational knowledge for Industry 4.0 applications(Gocev, Grimm, and Runkler 2018), and within applications in Cognitive Systems (Behnke et al. 2015). In the last two works, the emphasis is on the support of the ontological framework to generate explanations of plans, reflecting the current emphasis on explainable AI. In particular, Gocev et al argue it is natural to integrate PDDL with ontological reasoning for explanation generation especially in an industrial "co-worker" setting. An established transformation between for example OWL and PDDL can work both ways: it can help build up the planning domain model, and help explain the plans subsequently generated. While the authors give a good account of the languages and illustrate the transformation with examples, the mapping only involves mapping expressions in the PDDL language to (annotated) expressions in OWL, where classes have been "tagged" as actions.

Bonasso et al's (Bonasso et al. 2013) work in Space applications shows an integration of an ontological model in OWL with existing planning operations and editors. Its aim is to support subject experts encode and validate application knowledge which is targeted to be used within space operations planning. They use an editor (PRONTOE) to create the ontological information, or import it from other structured web sources. Though the addition of ontological modelling appears to be an add-on, the overall support environment is impressive and appears comprehensive.

From a knowledge engineering perspective, the use of ontologies is shown above to help with plan explanation generation, or verification of action encodings, but there is little work on utilising ontological modelling to actively promote the knowledge acquisition phase of a plan generation application. One notable exception is KEWI (Wickler, Chrpa, and McCluskey 2015), which does support knowledge acquisition, but it falls short in not using the *shared* aspect of ontologies, where knowledge acquisition can take full advantage of objects and classes from previous work.

Most systems in past work seem to have built up a planning application, then augmented this with an ontology interface, or built up an ontological model and subsequently added in a planning function. Our Case Study described below leads us to focus on the development of an intelligent system covering a whole enterprise, typically an industrial process area, where the ontological modelling and planning functions are developed together. Such a development is:

- integrative: the idea is to integrate a large area of objects and processes of an enterprise in such a way that the effect of different parts of a larger system can be modelled to explore their inter-operating behaviour. In our Case Study, the integration of remote monitoring facilities, maintenance management (scheduled, routine servicing and condition-based servicing) and maintenance

Table 1: Summary of practical work since 2005 on use of ontologies in P&S

Work Reference	Domain	Translation	Comment
(Cioffi and Thompson 2006)	Web Service Composition	Integrated	Aimed specifically for KEPS
(Bouillet et al. 2007)	Workflow Compostion	Integrated	Comprehensive Tool Support
(Celino and Dell'Aglio 2012)	Simulation Learning	Integrated	Verification of Domain Models
(Asunción et al. 2005)	Crisis Management	OWL-S -HTN	Integrated System
(Liu et al. 2013)	Emergency Planning	OWL-HTN	Used with SHOP2 planner
(Bonasso et al. 2013)	Space Operations	Integrated	Supports Action Authoring and Interactive Planning
(Behnke et al. 2015)	Cognitive Systems	HTN-DL	Explanation Generation
(Wickler, Chrpa, and McCluskey 2015)	Drilling Planform Processes	KEWI-PDDL	Supports Knowledge Engineering
(Feljan et al. 2017)	Transport Systems	OWL-PDDL	Planning small part of a wider system
(Gocev, Grimm, and Runkler 2018)	Industry 4.0	OWL-PDDL	Explanation Generation
(Getuli 2020)	Building Construction	Integrated	tool-supported human planner

activity all carried out within a fixed facility called the maintenance depot needs to be considered together. Such endeavours can be seen as important within the context of initiatives such as Industry 4.0. (Gocev, Grimm, and Runkler 2018).

- multi-functional: the embedding of reasoning with activities into such an extensive enterprise comprises a range of roles. In our Case Study, dynamic scheduling of activities, as well as planning the operation of maintenance actions, and autonomous enactment within robotic devices, all required P&S. One needs an extensive knowledge engineering exercise to provide the 'glue' for the whole system.

Capturing an industrial enterprise this way is analogous to the work in model-based, enterprise information technology architecture, where the components of the architecture are precisely and formally specified. Though the idea of formalising enterprises through ontologies has been around for decades (e.g. (Fox and Grüninger 1997)), it seems to be the case that enterprise models are designed more for human to human communication than for automated analysis or autonomous operation (Antunes et al. 2014). The state of the art in this area seems to stop at the use of formalised models to pursue optimization, and the impact of change (Florez H 2016). In fact, the kind of exercise we are embarking on does not stop at model analysis, it ranges to goal directed synthesis of strategies to achieve business goals (which could of course include optimisation but is not limited to that).

In this context, our ambition is to create a systematic approach to support the construction of P&S applications which emerge from and take advantage of the benefits of an emerging ontological framework. These advantages include the opportunity for validation through the ontology, for the inclusion of additional knowledge through the shared aspect of the ontology creation, and the possibility for automated domain model construction. The Case Study below provided a motivation for this endeavour as an example of a project which includes both *the need for an integrated ontology and the use of automated planning processes* from its initial conception - hence the emphasis on an integrated approach to knowledge acquisition within this context. On the choice between the use of a dedicated planner within the ontological framework, and the use of an external planner, we favour the latter in general; the advantage of the availability of a wide range of planning engines, and the flexibility that this brings, helps in the multi-functional nature of the applications.

Case Study Overview

Importance of Rolling Stock Maintenance

In the UK, passenger numbers on the rail network have doubled since 1994 and the railway industry forecasts further growth in fleet sizes for all type of vehicles over the next 30 years. This places additional demands on the rolling stock maintenance facilities; also as more and more trains operate on the network train reliability becomes increasingly important to avoid disrupting the service. A large proportion of rolling stock life cycle costs are related to the preventive and corrective maintenance processes undertaken in the depot. The vision for the railway of the future is a system which includes the use of intelligent maintenance linked with enhanced condition monitoring (RTS 2020). Previous research has estimated that a potential reduction in delays of 15 per cent, valued at £90m/year, could be achieved if effective Remote Condition Monitoring (RCM) and maintenance planning was deployed. Additionally, the provision for automated condition data would improve asset management and maintenance scheduling providing better predictability and increased flexibility to respond to unpredicted events more efficiently. The general area of rolling stock maintenance can be divided into several categories; inspection (checking the condition of components), servicing (such as cleaning, fueling or topping up fluids) and maintenance (on-demand or routine repair or replacement).

The Case Study follows from the first year's work of a £1.8 million project (referred to as the SRS Project below) to establish a 'Smart Rolling Stock Maintenance Research Facility' in the Institute of Railway Research, University of Huddersfield. This work is funded by the European Regional Development Fund (ERDF). The aim is to carry out research into the improvement of the efficiency and effectiveness of rolling stock maintenance to meet the challenges of the near future of rail. The three main strands that the project focuses on are i) remote condition monitoring of rolling stock ii) depot maintenance management and organisation iii) robotic assistance for maintenance actions. As a 'glue' for the three strands an ontology is being built within a formal language to capture the structures, relationships and attributes of the main objects (rolling stock, sensors, depot assets, resources, etc). In particular, this ontology is being used to provide the building blocks of the specification of the maintenance operations. Tackling the whole enterprise in such an integrated fashion avoids fundamental problems with ontology-related approaches such as the integration problem (Osman, Ben Yahia, and Diallo 2021).

Existing Ontologies

Not surprisingly, we have found no previous work which aims to cover the scope of the SRS Project, but within the railway area there have been related works. Verstichel et al propose a hierarchical architecture for capturing the domain ontology of the train system (Verstichel et al. 2007). Their proposal in interesting in advancing a hierarchical or layered approach, where ontologies are created for each level. In another related work, Umiliacchi et al discussed the benefits of using ontologies for predictive rolling stock maintenance (Umiliacchi, Lane, and Romano 2011). They used a fragment of possible railway ontology to illustrate the idea. Our aim is to use the ontology for predictive maintenance, as well as diagnostics, and also automated planning of the maintenance activities.

While we found no work directly contributing to the SRS Project, we are using published ontological knowledge to help build up our conceptual model, such as:

- The RailML[2] standard provides a set of XML schemas to enable interoperable communication between heterogeneous railway applications. The rolling stock schema includes a high-level description of the rolling stock-related concepts. As the schema is defined collaboratively by experts, we are reusing the rolling stock fragment to support our models.
- The Smart Rail[3] ontology includes a high-level depot and rolling stock related concepts description. In our case, we need a domain ontology covering both rolling stock and depot-related concepts. This ontology includes a location definition that we can reuse in the context of our project. Indeed, we need the location of the train and also depot locations. The measurement units related to the remote condition monitoring/inspection data is also included. In the context of our SRS Project, we are utilising the existing ontologies capturing measurements and measurement units.
- The Vehicle ontology[4] is not related to railway vehicles; however, it makes use of two other ontologies, time and Space location that are needed in our context. As stated earlier, our ontology makes use of an existing time ontology already. The space ontology will be added to the list of space ontologies that can be reused in our project. This ontology includes an Ontology to capture concepts related to measurements that we will analyze.
- The RSSB T1010[5] report (Architecture Requirements) defines an interesting rolling stock XML schema that includes some structural aspects such as parts of the vehicle, for example bogies (a structural element of a train which contains two wheelsets and associated track and braking equipment and suspension components). Despite the relevance of the proposed schema, it is still limited and needs to be extended and formally captured. The benefits of using ontologies in the context of condition-based monitoring are also discussed. The schema includes very few concepts, but it gives an idea of what we need to capture. This schema, along with the RailML schema combined with experts from our team, helped to define our high-level rolling stock ontology.

Overview of the Current Model

In the context of the SRS Project, we need to capture four aspects as follows:

- Domain concepts (rolling stock and depot, vehicles and components);
- Time;
- Space (maintenance workspace - depot);
- maintenance activities and resources (including tools, humans and robot workers, and consumables).

We have defined a depot ontology using expert knowledge and maintenance manuals. Figure 1 gives a pictorial abstraction of our OWL ontology: for example, this asserts that a `Depot` has one or more `Sheds` having one or more `roads`. A `Depot` has a set of available `Resources` that can be `Workers`, `Materials`, `Consumables` or `Tools`.

A `Depot` is also equipped with `Machinery` covering the set of the machines required for servicing, inspection, and maintenance. A fleet is usually linked to a `Depot` for routing and periodic preventive maintenance.

[2]https://www.railml.org/

[3]https://ontology.tno.nl/smart-rail/

[4]http://ontology.eil.utoronto.ca/icity/Vehicle/1.2/

[5]https://www.rssb.co.uk/research-catalogue/CatalogueItem/T1010 - requires a login to access

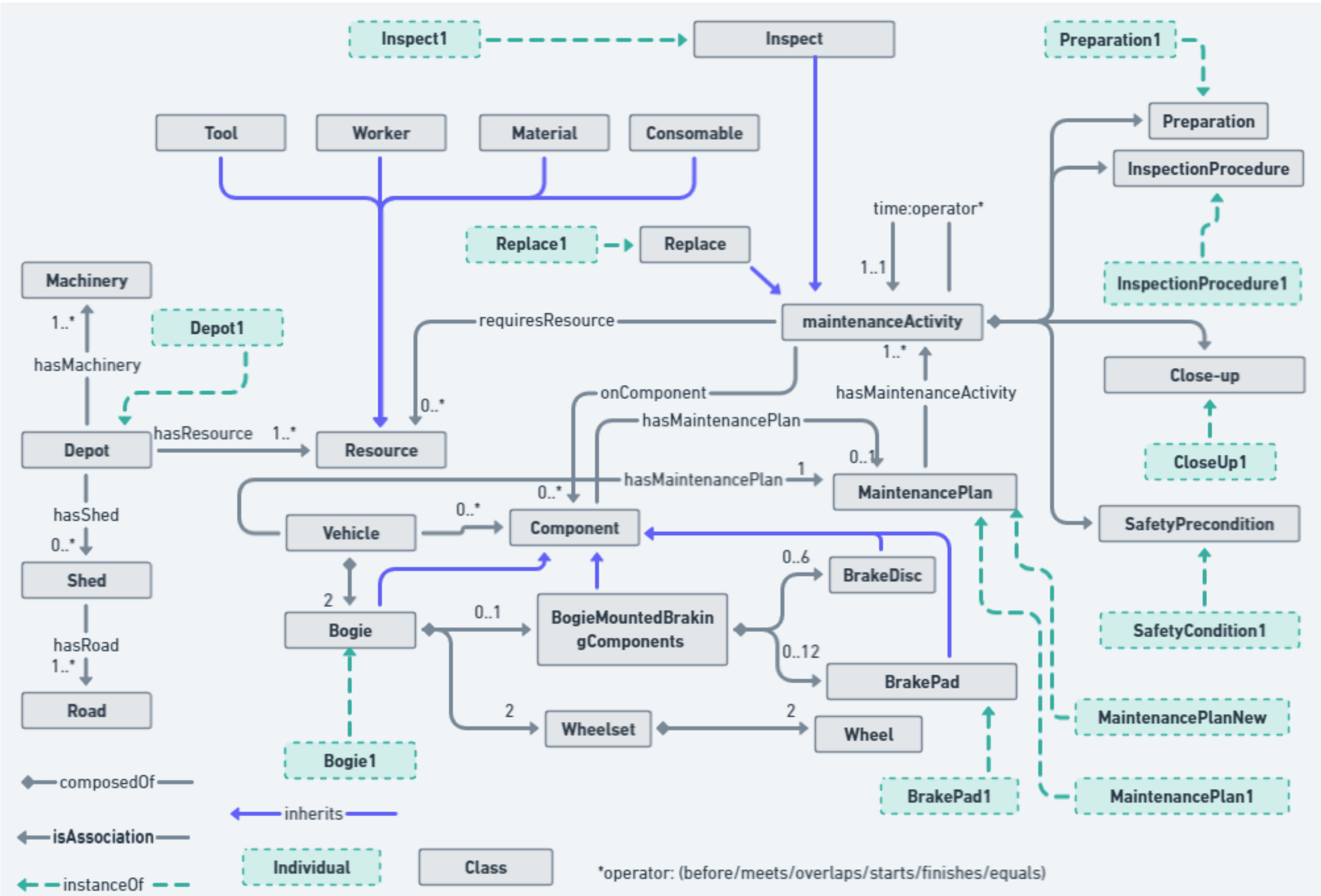

Figure 1: Ontology for capturing the depot and rolling stock concepts

The ontology is then linked to the rolling stock ontology using the `MantenanceActivity` class, where a Depot Road is used to perform a set of maintenance activities.

As stated above, we have reused existing community ontologies to define a high-level rolling stock ontology. Experts within IRR helped then to refine the `Bogie` fragment of the ontology. We also used standards and external private resources to define the Bogie sub-components attributes and maintenance activities-related concepts.

The component's attributes are required for maintenance purposes. The `BrakePad thickness` measurement for example is used as an input to the Brake Pad inspection. Indeed, based on the actual value of this attribute, the next maintenance step will be decided.

The assertions of the ontology (usually referred as Assertion Box or ABox) represent the state of a maintenance depot at a particular time. However, this instance of the world is not static. Various actions or the passing of time might introduce changes in the ABox, which have to be accommodated in real time or even be planned in advance to predict and avoid any potential critical event. As part of the ontology development process, we encode knowledge in Semantic Web Rule Language (SWRL) [6] to express all those actions that would revise the state of the maintenance depot.

SWRL is a combination of OWL DL with the markup language RuleML, which allows the automatic creation of OWL assertions when a predefined condition evaluates true. Inference rules written in SWRL have the form of an implication: the body resembles a condition, expressed as a conjunction of OWL assertions and the head expresses the knowledge to be inferred. For instance consider the rule:

$$hasParent(?Jo, ?Ann) \wedge hasBrother(?Ann, ?Bob) \rightarrow hasUncle(?Jo, ?Bob)$$

If the ontology states that Ann is Jo's parent and Ann has a bother named Bob, then a new assertion is created stating that Bob is Jo's uncle.

In the context of the maintenance depot, inference rules are used to describe how different actions may change the current state of the depot. The body of a rule contains the inputs and preconditions of the action. These are the entities that participate in this change and the constraints that enable it, respectively. The head of the rule, on the other hand,

[6]https://www.w3.org/Submission/SWRL/

describes the effect that this action has on the ontology: a) output, new entities that have been created; b) postconditions, assertions that express all the changes that the action brought to the current state of the depot.

In the current phase of our conceptualization, the rules are populated manually - hard coded - by ontology engineers. The rules encapsulate knowledge that is derived from formal textual sources, indicatively maintenance depot operation standards, guides and inspection manuals (e.g. RIS-2766 Rail Industry Standard for Wheelsets [7]). Some instances of inference rules and their effect on the ontology are described on Section .

Examples Illustrating Translation to Planning Knowledge

For the purposes of this paper we illustrate how a combination of ontological items and a reasoner infused with inference rules can contribute to the automated construction of planning models based on OWL assertions. In the below illustration this draws in a large part from ontological knowledge expressed in SWRL. Previously, the only use of SWRL knowledge within the related works appears to have been in Bonasso et al's Ontology Editor PRONTOE (Bonasso et al. 2013). The use of SWRL there (we assume) was for validation checking, rather than as a direct aid to acquiring the knowledge to insert into a planning domain model.

Our approach is based on the analogy between an action as it is defined in the conceptualization shown above and the action as an element within PDDL. The body of a rule, containing the inputs and the conditions of an action contribute to the formation of the preconditions in a PDDL action. Similarly, the head of a rule that is responsible for the effect of the action in the ontology indicates elements that reveal the postconditions within a PDDL action.

We consider the scenario of a vehicle with a defective brake pad that undergoes an ordinary maintenance. The goal is to execute a series of actions that will fix the damage and ensure the safe operation of the train. For the sake of simplicity, we assume a simplified version of the initial ontology, which depicts the state of the world shown in Figure 1. For the sake of readability, we assume a simplified SWRL syntax that shows only those entities that are necessary to understand each example. In addition, we assume the following fictional SWRL functions:

- *_create(?name, ?class)* creates a new instance ($?name$), which is of type $?class$;
- *_lessThan(?a, ?b)* that stands for $?a < ?b$;
- *_greaterThan(?a, ?b)* that stands for $?a > ?b$;
- *_sum(?N, ?a, ?b)* that stands for $?N = ?a + ?b$;
- *_subtract(?S, ?a, ?b)* that stands for $?S = ?a - ?b$.

Using this setup, we list a series of examples that illustrate how SWRL rules on maintenance activities can provide elements of planning knowledge, which can then be used to synthesize the state-based pre and post condition actions.

[7] https://www.rssb.co.uk/standards-catalogue/CatalogueItem/RIS-2766-RST-Iss-1

In the first example, we emphasize on the action of vehicle inspection. According to the Bogie Inspection Guidelines, an inspection includes several sub-activities:

- **preparation** of the equipment to be inspected
- **procedure** of the inspection
- **close up** - reverse the effects of preparation to restore the equipment to its initial state
- adhere to the **safety conditions**

These sub-activities must happen in a particular temporal order: starting with preparation, followed by the inspection and finished with the closing up, whereas the safety conditions must be met for the whole process (Figure 2).

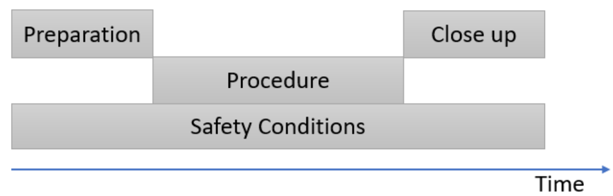

Figure 2: Temporal order of sub-activities during an inspection

Rule (1) expresses this knowledge using the Allen operators, through the OWL-time ontology. In detail, if an inspection activity is scheduled, then all the aforementioned sub-activities must be created and adjusted to adhere to the temporal ordering.

(Rule 1:) *Inspection(?i)* $\wedge$ *_create(?sc, SafetyCondition)* $\wedge$ *_create(?pr, Procedure)* $\wedge$ *_create(?cu, CloseUp)* $\wedge$ *_create(?p, Preparation)* $\rightarrow$ *starts(?p, ?i)* $\wedge$ *finishes(?cu, ?i)* $\wedge$ *equals(?sc, ?i)* $\wedge$ *after(?pr, ?p)* $\wedge$ *after(?cu, ?pr)*

The terms in the body of the rule describe the required actions for a successful inspection. This is done with the series of $\_create$ terms, which express all the new activities that must be created/planned within the depot. The head of the rule outlines the temporal ordering of these activities, which in turn defines the effects of the inspection process. More specifically, the temporal constraints of the durative action *inspection* can be implied by mapping the Allen operators, starts, finishes and equals with the PDDL constraints *at start*, *at end* and *over all* respectively. Furthermore, the ordering that the procedure of inspection is after the preparation and before the closeup can infer: a) the preparation is part of the condition of the action; b) the closeup is part of the effect and the procedure of inspection occurs at the end of the condition and continues as an effect of the action. A PDDL representation of the action is shown in Figure 3.

The second example focuses on the overall duration of composite actions. Along the same lines as the previous example, the goal is to determine the minimum possible duration of the inspection process. We assume that the sub-activities: preparation, inspection procedure and close up are executed in a sequential order. Rule (2) expresses

```
(:durative-action inspection
    :parameters (?v - vehicle)
                (?bp - brakepad)
    :duration(= ?duration 3)
    :condition (and (over all (safetyCondition ?v))
                    (at start (preparation ?v))
                    (at end (inspection_procedure ?bp))
    :effect (and    (at start(inspection_procedure ?bp))
                    (over all (safetyCondition, ?v))
                    (at end(closeup ?v))
```

Figure 3: Example of derived PDDL temporal conditions

that the estimated duration of the inspection (denoted as ?iDuration) is greater or equal to the sum of the duration of all the individual sub-activities. The term *hasDuration* in the head of the rule or the *sum* function in the body may be used to express the overall duration of the inspection, in PDDL terms, as a static or dynamic value, respectively.

(Rule 2:) *Inspection(?i)* $\wedge$ *Preparation(?p)* $\wedge$ *Procedure(?pr)* $\wedge$ *CloseUp(?cu)* $\wedge$ *hasDuration(?p, ?pDuration)* $\wedge$ *hasDuration(?pr, prDuration)* $\wedge$ *hasDuration(?cu, ?cuDuration)* $\wedge$ *_sum(?iDuration, ?pDuration, ?prDuration, ?cuDuration)* $\rightarrow$ *hasDuration(?i, iDuration)*

Next we focus on the actual procedure of inspection, where the brake pads of a vehicle are checked for defects or any sign of deterioration. The input of this action is the component that undergoes the inspection along with the attributes that describe its state. The output is the necessary maintenance activity that will ensure the right functioning of the equipment.

Rule (3) is built according to the inspection guidelines for brake pads and expresses that if a brake pad is worn out for more than 14 mm, then this damage is considered as critical and an immediate replacement is necessary. This structure provides the planning knowledge to synthesize the action of initiating a new replacement activity for the defective braking pad. The preconditions are defined in the body of the rule (thickness is less than 14mm), whereas the head outlines the effect of this action, which is the initiation of a replacement activity of critical priority, in which the defective brake pad must be replaced with a new one.

(Rule 3:) *Bogie(?bg)* $\wedge$ *BrakePad(?bp)* $\wedge$ *hasPart(?bg, ?bp)* $\wedge$ *thickness(?bp, ?thickness)* $\wedge$ *_lessThan(?thickness, 14)* $\rightarrow$ *_create(?mp, Replace)* $\wedge$ *_create(?newbd, BrakePad)* $\wedge$ *hasPriority(?mp, Critical)* $\wedge$ *requiresResource(?mp, ?newbd)* $\wedge$ *onComponent(?mp, ?newbd)*

As a final example, rule (4) expresses how the aforementioned replace action will take place. This is achieved by checking the availability of spare brake pads (in a more realistic example the available staff and tools should be also taken into account). The effect of this action is the attachment of the spare part to the parent component, the bogie and the adjustment of the available spare parts after the replacement.

(Rule 4:) *Bogie(?bg)* $\wedge$ *BrakePad(?bd)* $\wedge$ *hasPart(?bg, ?bd)* $\wedge$ *Replace(?mp)* $\wedge$ *onComponent(?mp, ?bd)* $\wedge$ *requiresResource(?mp, ?newbd)* $\wedge$ *Depot(?dp)* $\wedge$ *hasBrakePadAvailability(?N)* $\wedge$ *_greaterThan(?N, 0)* $\wedge$ *_subtract(?newN, ?N, 1)* $\rightarrow$ *hasPart(?bg, ?newbd)* $\wedge$ *hasAvailability(?newN)*

The body of the rule implies the preconditions of the replacement action, that is, the availability of spare parts in the depot. The terms in the head, on the other hand, outline the postconditions: the bogie is equipped with a new part ($?newbd$) and the available number of spare brake pads is reduced by 1 ($newN = N - 1$).

To summarise this exercise, we have illustrated how knowledge already captured in the ontology can be translated to a PDDL language (PDDL2.1), in terms of action schema. Assertions in the ABox will then form an initial state, and be combined with a user-supplied goal to create a problem file. We are currently designing an automated process to perform this extraction, as part of a systematic method to assemble domain models to be used with an external plan engine. We aim to use this same translation method in any area of the SRS Project where P&S services may be required.

Conclusion

In this paper we have focused on the challenges and opportunities of KEPS within the context of a wider project encompassing ontological modelling of domain knowledge. Applying a coherent enterprise-wide approach, such as in the SRS project, can integrate heterogenous activities within a large enterprise, with the possibility of multiple automated functions such as process simulation and maintenance optimisation. In our case, we are aiming for integrative support, through a central hierarchical ontology, from remote condition monitoring at one end of the problem, through to robotic maintenance at the other.

We surveyed and summarised work in the last 15 years that involved capturing knowledge in ontologies *and* required the implementation or use of P&S functions. These works covered a wide range of areas, and together emphasised the great opportunities and benefits that this wider modelling effort would bring to KEPS. We then introduced a novel application area for P&S, that of managing and enacting maintenance activities in a rail rolling stock depot, and illustrated our progress in modelling this formally. Within this context, we illustrated (through an example involving bogie brake pads) a way to use SWRL rules to help create a planning domain model. Our aspiration is to make this part of a methodical approach to automatically assembling such planning domain models from the ontology, and using an external planner to solve problems such as maintenance optimisation (e.g. maximum availability and reliability of trains, minimum use of resources, minimum waste (by avoiding replacing components too early)).

Planned directions for future work are to create parsers and translators to automatically extract planning knowledge (in PDDL for example) as illustrated in the brake pad example given. Work already done and tools created in translating between PDDL and Web Ontology languages in the *web service composition* area will assist us here (Durcik and Paralic 2011). Also, we intend to extend the current temporal representation of maintenance activities to spatio-temporal entities, which will facilitate the generation of pre and post conditions that include both temporal and spatial dependency. This conceptualization ties well with the spatial extent of a depot, where the spatial intersection of vehicles, equipment or tools poses an important requirement.

References

Antunes, G.; Bakhshandeh, M.; Mayer, R.; Borbinha, J.; and Caetano, A. 2014. Using Ontologies for Enterprise Architecture Integration and Analysis. *Complex Systems Informatics and Modeling Quarterly* doi:https://doi.org/10.7250/csimq.2014-1.01.

Asunción, M. D. L.; Castillo, L.; Fdez-olivares, J.; García-pérez, O.; González, A.; and Palao, F. 2005. Knowledge and plan execution management in planning fire fighting operations .

Behnke, G.; Bercher, P.; Biundo, S.; Glimm, B.; Ponomaryov, D.; and Schiller, M. 2015. Integrating Ontologies and Planning for Cognitive Systems. In *Proceedings of the 28th International Workshop on Description Logics (DL 2015)*. CEUR Workshop Proceedings.

Bonasso, P.; Boddy, M.; Kortenkamp, D.; and Bell, S. 2013. Ontological Models To Support Planning Operations. in: Proceedings of 8 th International Workshop on Planning and Scheduling for Space.

Bouillet, E.; Feblowitz, M.; Liu, Z.; Ranganathan, A.; and Riabov, A. 2007. A Knowledge Engineering and Planning Framework based on OWL Ontologies.

Celino, I.; and Dell'Aglio, D. 2012. An Ontological Formulation and an OPM Profile for Causality in Planning Applications. In *The Semantic Web*. Springer Berlin Heidelberg. ISBN 978-3-642-29923-0.

Cioffi, M.; and Thompson, S. G. 2006. Planning with the Semantic Web by fusing Ontologies and Planning Domain Definitions. In *Research and Development in Intelligent Systems XXIII, Proceedings of AI-2006, the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence.* Springer.

Durcik, Z.; and Paralic, J. 2011. Transformation of Ontological Represented Web Service Composition Problem into a Planning One. *Acta Electrotechnica et Informatica* 11. doi:10.2478/v10198-011-0014-y.

Feljan, A. V.; Karapantelakis, A.; Mokrushin, L.; Liang, H.; Inam, R.; Fersman, E.; Azevedo, C. R. B.; Raizer, K.; and Souza, R. S. 2017. A Framework for Knowledge Management and Automated Reasoning Applied on Intelligent Transport Systems. *CoRR* URL http://arxiv.org/abs/1701.03000.

Florez H, Sánchez M, V. J. 2016. A catalog of automated analysis methods for enterprise models. *Springerplus* doi:doi:10.1186/s40064-016-2032-9.

Fox, M.; and Grüninger, M. 1997. *Ontologies for Enterprise Modelling*. doi:10.1007/978-3-642-60889-6_22.

Getuli, V. 2020. *Ontologies for Knowledge modeling in construction planning: Theory and application*. Firenze University Press. doi:10.36253/978-88-5518-184-6.

Gil, Y. 2005. Description Logics and Planning. *AI Magazine* doi:10.1609/aimag.v26i2.1814.

Gocev, I.; Grimm, S.; and Runkler, T. A. 2018. Explanation of Action Plans Through Ontologies. In *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*. Springer International Publishing.

Liu, D.; wei Wang, H.; yu Liu, K.; and Wang, J. 2013. An ontology-based emergency domain knowledge model for HTN planning. *IFAC Proceedings Volumes* 13th IFAC Symposium on Large Scale Complex Systems: Theory and Applications.

Osman, I.; Ben Yahia, S.; and Diallo, G. 2021. Ontology Integration: Approaches and Challenging Issues. *Information Fusion* doi:https://doi.org/10.1016/j.inffus.2021.01.007.

RTS. 2020. Railway Technical Strategy 2020. https://railtechnicalstrategy.co.uk/wp-content/uploads/2020/11/The-Rail-Technical-Strategy.pdf.

Umiliacchi, P.; Lane, D.; and Romano, F. 2011. Predictive maintenance of railway subsystems using an Ontology based modelling approach. 9th World Conference on Railway Research.

Verstichel, S.; Van Hoecke, S.; Strobbe, M.; Van den Berghe, S.; De Turck, F.; Dhoedt, B.; Demeester, P.; and Vermeulen, F. 2007. Ontology-driven middleware for next-generation train backbones. *Science of Computer Programming* .

Wickler, G.; Chrpa, L.; and McCluskey, T. 2015. Ontological support for modelling planning knowledge. In *Knowledge Discovery, Knowledge Engineering and Knowledge Management - 6th International Joint Conference, IC3K 2014, Revised Selected Papers*, Communications in Computer and Information Science. Springer Verlag.