Situation Prediction Nets

Playing the Token Game for Ontology-Driven Situation Awareness*

Norbert Baumgartner[1], Wolfgang Gottesheim[2], Stefan Mitsch[2], Werner Retschitzegger[2], and Wieland Schwinger[2]

[1] team Communication Tech. Mgt. GmbH, Goethegasse 3, 1010 Vienna, Austria
[2] Johannes Kepler University Linz, Altenbergerstr. 69, 4040 Linz, Austria

Abstract. Situation awareness in large-scale control systems such as road traffic management aims to predict critical situations on the basis of spatio-temporal relations between real-world objects. Such relations are described by domain-independent calculi, each of them focusing on a certain aspect, for example topology. The fact that these calculi are described independently of the involved objects, isolated from each other, and irrespective of the distances between relations leads to inaccurate and crude predictions. To improve the overall quality of prediction while keeping the modeling effort feasible, we propose a domain-independent approach based on Colored Petri Nets that complements our ontology-driven situation awareness framework *BeAware!*. These Situation Prediction Nets can be generated automatically and allow increasing (i) prediction precision by exploiting ontological knowledge in terms of object characteristics and interdependencies between relations and (ii) increasing expressiveness by associating multiple distance descriptions with transitions. The applicability of Situation Prediction Nets is demonstrated using real-world traffic data.

Key words: Situation Awareness, Ontology, Colored Petri Nets

1 Introduction

Situation awareness in large-scale control systems. Situation awareness is gaining increasing importance in large-scale control systems such as road traffic management. The main goal is to support human operators in assessing current situations and, particularly, in predicting possible future situations in order to take appropriate actions pro-actively to prevent critical events. The underlying data describing real-world objects (e. g., wrong-way driver) and their relations (e. g., heads towards), which together define relevant situations (e. g., wrong-way driver rushes into traffic jam), are often highly dynamic and vague. As a consequence reliable numerical values are hard to obtain, which makes *qualitative situation prediction* approaches better suited than quantitative ones [14].

* This work has been funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT), grant FIT-IT 819577.

Ontology-driven situation prediction based on spatio-temporal calculi. Recently, ontology-driven situation awareness techniques [9], [2] have emerged as a basis for predicting critical situations from spatio-temporal relations between objects. Such relations are expressed by employing *relation calculi*, each of them focusing on a certain spatio-temporal aspect, such as mereotopology [23], orientation [11], or direction [22]. These calculi are often formalized by means of *Conceptual Neighborhood Graphs* (CNGs, [13]), imposing constraints on the existence of transitions between relations. CNGs are an important construct for modeling continuously varying processes [20], and are adopted in, for example, qualitative simulation [10], prediction [6], tracking moving objects [25], and agent control [12]. The domain-independent nature of calculi (i) leaves interpretations (e. g., `Close` means within 10km) to applications, (ii) does not consider object characteristics (e. g., whether they are moveable), (iii) is irrespective of interdependencies (e. g., topological transitions depend on spatial distance), and (iv) does not express any kind of distance for transitions such as probability, which altogether lead to inaccurate and crude situation predictions. Existing approaches try to increase quality by constructing domain- and even situation-specific calculi manually, which, however, requires considerable modeling effort.
Colored Petri Nets to the rescue. In order to achieve a proper balance between prediction quality and modeling effort, we propose a domain-independent approach on the basis of *Colored Petri Nets* (CPNs, [16]) that complements our ontology-driven situation awareness framework BeAware! [5]. Representing CNGs as CPNs allows, on the one hand, increasing prediction precision by exploiting ontological knowledge included in the framework in terms of object characteristics and interdependencies between spatio-temporal relations and, on the other hand, increasing prediction expressiveness by associating transitions with dynamically derived distances for multiple view-points. These so called *Situation Prediction Nets* (SPN) are derived automatically from the situation awareness ontologies of BeAware!. Petri net properties are preserved, which enables features such as predicting multiple situation evolutions in parallel, which are not as easily realizable with alternative formalisms such as state transition diagrams. The applicability of SPNs is demonstrated using real-world traffic data.
Structure of the paper. In Sect. 2, a brief overview of our work on situation awareness is given, detailing further the challenges tackled in this paper by means of a road traffic example. Section 3 discusses related work, Sect. 4 introduces SPNs, and their applicability is discussed in Sect. 5. Finally, Sect. 6 concludes the paper with lessons learned and an outlook on further research directions.

2 Motivating Example

Road traffic management systems responsible for example, for improving traffic flow and ensuring safe driving conditions are a typical application domain of situation awareness. Based on our experience in this area, examples from road traffic management further detail the challenges of enhancing the quality of *neighborhood-based predictions* [10] in situation awareness. In such neighborhood-based predictions, the relations of a current situation are the starting point for tracing transitions in CNGs to predict possible future situations.

In our previous work [5], we introduced a generic framework for building situation-aware systems that provides common knowledge about (i) situations, which consist of objects and relations between them, and (ii) relation neighborhood in a domain-independent ontology. This ontology is used in generic components to derive new knowledge from domain information provided at runtime. A prototypical implementation supports assessing situations in real-world road traffic data, which in turn forms the basis for simple predictions [6] following a neighborhood-based approach. To illustrate the shortcomings of neighborhood-based prediction approaches—which are rooted in relation calculi and CNG characteristics in general—let us consider the following example: Suppose that an initial situation is assessed in which a wrong-way driver is heading towards an area of road works, as depicted in Fig. 1. This initial situation is characterized

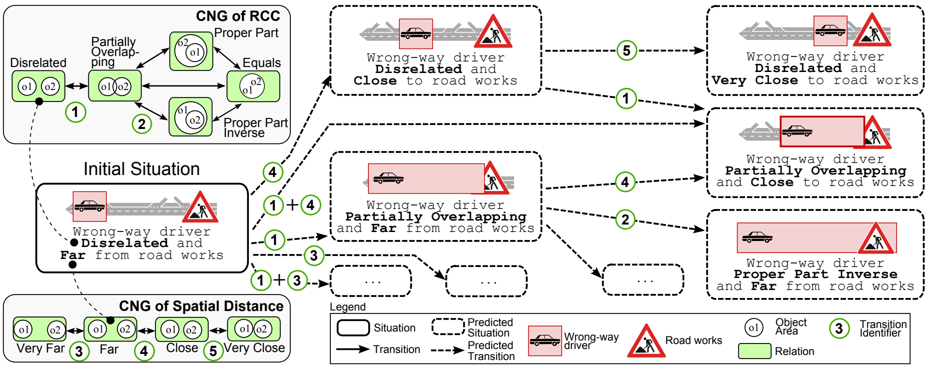

Fig. 1: Prediction of possible future situations from an assessed, current one.

by two relations (`Disrelated` from Region Connection Calculus (RCC) [23] and `Far` from Spatial Distance calculus) between the objects `Wrong-way driver` and `Road works`. Human operators would like to know potential future situations—for instance, a wrong-way driver in the area of road works—in order to take appropriate actions. With neighborhood-based prediction, we can provide this information by following the edges of `Disrelated` (1) and `Far` (3, 4) in the respective CNGs, thereby predicting five possible subsequent situations, which form the basis for further predictions. This leads, however, to combinatorial explosion, making such crude predictions—even when using small CNGs as in the example above—inaccurate and incomprehensible to human operators. Current approaches [1] try to tackle these problems with manually defined constraints resulting in high modeling effort. To further illustrate the challenges of enhancing prediction quality, we provide examples below and a summary in Fig. 2.

Challenge A: Increasing precision with object characteristics. CNGs model relations independently of objects and their characteristics, which results

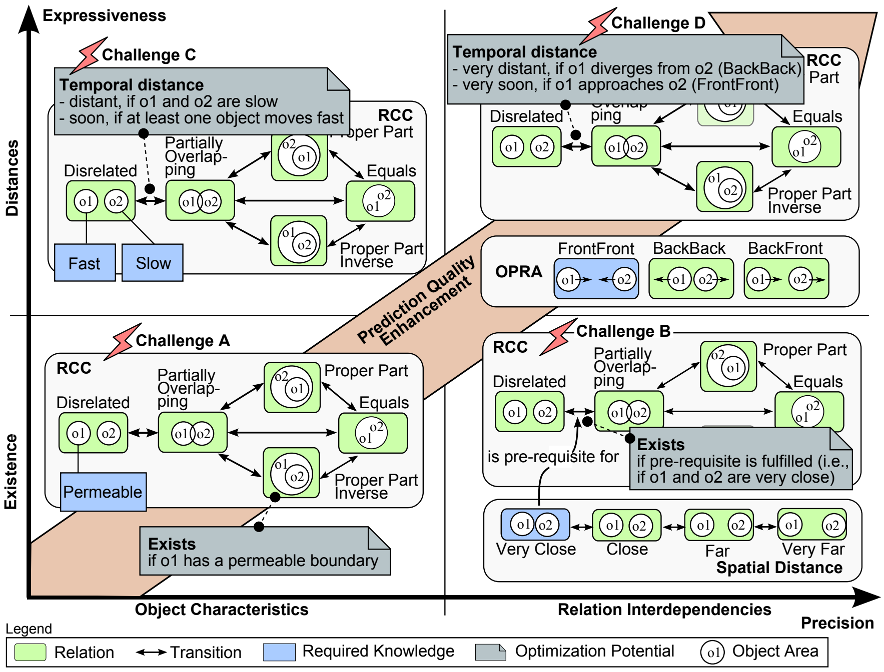

Fig. 2: Challenges in prediction quality enhancement.

in a vast number of predicted situations, of which many are actually impossible. Let us revisit our example: Since a wrong-way driver has non-permeable boundaries, all predictions that require areas of other traffic objects to become a proper part of the wrong-way driver are wrong (i. e., resulting in impossible situations, e. g., "Wrong-way driver Proper Part Inverse road works"). If only we had knowledge about object characteristics, such as boundary permeability, available in CNGs, we could increase prediction precision as we exclude impossible situations. In [4], we laid the basis for tackling this challenge by proposing an ontology for representing object characteristics and corresponding optimization rules. In this paper, we describe how to translate this knowledge to SPNs.

Challenge B: Increasing precision with relation interdependencies. A CNG describes relations of a single calculus without taking interdependencies between different calculi into account, which again leads to a large number of false predictions. For example, let us reconsider the initial situation "Wrong-way driver disrelated and far from traffic jam" involving two different calculi. In a world in which motion is continuous, transitions in these calculi are not independent. In our example, this means that two objects will first transition in Spatial Distance calculus from `Far` to `Close`, then advance to `Very Close` before a transition in RCC from `Disrelated` to `Partially Overlapping` can happen. If only we were able to describe such relation interdependencies between different calculi, we could further increase prediction precision.

Challenges C and D: Increasing expressiveness with distance descriptions. CNGs only describe the mere existence of transitions between relations, but do not associate any distance descriptions with them. Human operators, however, are keen to know details beyond existence, such as temporal distance, probability, impact, and confidence. For example, the duration of a transition from `Far` to `Close` depends on the speed of both objects, while the transition probability is influenced by the direction of motion (i. e., whether or not the wrong-way driver heads towards road works, which is expressed by an orientation relation `FrontFront` of OPRA [11]). If only we had such knowledge about object characteristics, for example their speed, and about relation interdependencies, for example mereotopology on spatial distance, we could derive distance descriptions to increase prediction expressiveness.

3 Related Work

In this section, we discuss related research in qualitative neighborhood-based prediction and simulation because, in fact, such predictions are based on simulating evolutions of situations. We distinguish between methods trying to increase *precision* and those focusing on *expressiveness*. Since our approach generates CPNs from ontologies, we also cover related work in this area.

Increasing precision. Increasing precision is a major concern in fields such as qualitative simulation [1], [7] and robot agent control [12]. In [1], a qualitative simulation method was presented which manually defines (i) simulated situations more precisely with unmodifiable relations and object characteristics (e. g., relative positions of static objects, termed *intra-state constraints*), and (ii) customized CNGs (termed *inter-state constraints*, or dynamic constraints in a similar approach [7]) to describe valid transitions for determining the next simulated situation. Similarly, in [12], the effect of object characteristics (e. g., whether objects can move or rotate) on the conceptual neighborhood of a particular calculus was emphasized. To represent this knowledge in terms of CNGs, six different manually defined conceptual neighborhoods of the orientation calculus $\mathcal{OPRA}_m$ were introduced. The drawback common to all these approaches is that intra- and inter-state constraints (CNGs) must be defined manually for each domain or even worse for each prediction. Situation Prediction Nets, in contrast, can be derived automatically from such ontological knowledge, allowing us, at the same time, to increase prediction precision.

Increasing expressiveness. Increasing expressiveness is of concern, for instance, when assigning preferences to CNGs in order to customize multimedia documents [18] or to describe costs for assessing spatial similarity [19]. Laborie [18] uses spatio-temporal calculi to describe relations between parts of a multimedia document, and CNGs to find a similar configuration, in case a multimedia player cannot deal with the original specification of the document. In order to increase expressiveness with preferences for selecting the most suitable configuration, distances between relations are described with statically defined, quantitative weights on CNG edges. Similarly, Li and Fonseca [19] increase expressiveness with weights to describe distances in terms of static costs for making transitions in a CNG. These costs are used to assess spatial similarity: less costly transitions connect relations that are more similar. We take these approaches further by increasing expressiveness with qualitative distances for multiple view-points which incorporate relation interdependencies in addition to object characteristics.

Translating ontologies to Petri nets. Petri nets are commonly known models appropriate for describing the static and dynamic aspects of a system, thereby enabling prediction of future situations by simulating evolutions [24]. Of particular interest for defining our Situation Predictions Nets are extensions to the original place-transition net formalism in the form of hierarchical Colored Petri Nets [16], because they allow representing ontological knowledge with complex data types. Translations from ontologies to Petri nets are described in the literature as a pre-requisite, for instance, for defining a hybrid ontological and rule-based reasoner [26] or for achieving formal analyses of Web services [8]. In [26], patterns were presented for translating OWL axioms of a particular ontology with their accompanying SWRL rules into Petri nets in order to create a combined ontological and rule-based reasoner. In [8], translations from concepts in OWL-S Web service specifications (such as choice, sequence, and repeat-until) into a custom-defined Petri net variant were introduced to check static properties of the Web service, such as its liveness. Both approaches focus on translations from ontologies to Petri nets, but, in contrast to our approach, they do not exploit dynamic information to influence the behavior of their nets.

4 From Ontologies to Situation Prediction Nets

In this section, we propose Situation Prediction Nets (SPN) based on CPNs for tackling the challenges described in Sect. 2 with the goal of enhancing prediction quality in domain-independent situation awareness. Note that the modeling examples given in this section—in order to keep them concise and comprehensible—only show simplified subsets of our nets, describing aspects relevant to situation awareness, such as mereotopology, distance, speed, and orientation.

As illustrated in Fig. 3, the overall architecture of our approach formalizes a conceptual view of objects, spatio-temporal relations between them (i. e., a prediction's *static definition*), and situations (i. e., a prediction's *dynamic starting point*) as knowledge for situation-aware systems in an ontology[1], which in turn forms the basis for generating Situation Prediction Nets. This situation awareness ontology is structured into: (i) a domain-independent part including spatio-temporal calculi and their CNGs, accompanied by algorithms for situation assessment, duplicate detection, and prediction, forming our situation awareness framework BeAware! [5], and (ii) a domain-dependent part extending the domain-independent part during implementation of situation-aware systems.

[1] Our situation awareness ontology [5] builds upon the notion of Barwise and Perry [3], which makes the proposed approach applicable to a wide range of efforts in situation awareness, for instance, to Kokar's approach [17].

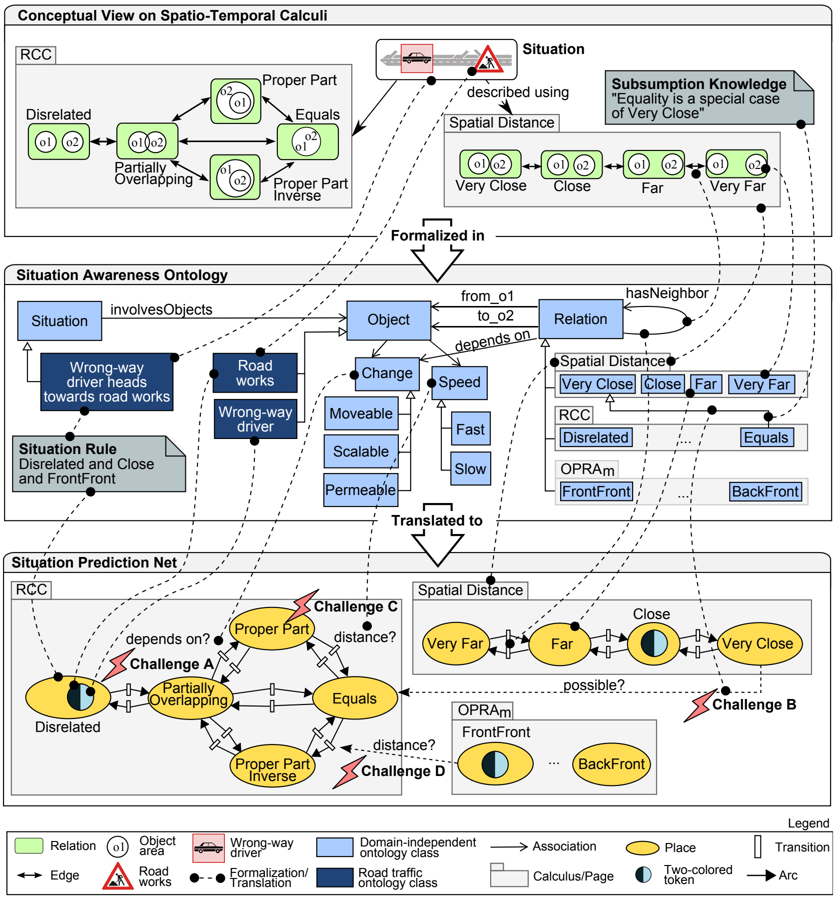

Fig. 3: From ontological knowledge to Situation Prediction Nets.

Translating from ontological knowledge to CPNs. We first describe how to translate automatically ontological knowledge to plain CPNs as a basis for Situation Prediction Nets. As illustrated in Fig. 3 and formalized in Tab. 1, translating the static structure of a CNG (an undirected, unweighted graph) to a CPN (a bipartite directed graph) is straightforward: (i) We represent every calculus on a dedicated Petri net page, (ii) define the CPN's color set, (iii) create one place per node of the corresponding CNG, (iv) represent every edge with two transitions (one for each direction), and connect transitions with arcs to the respective places. (v) Situations, being defined by objects and relations, can be modeled by objects as two-colored tokens, being composed of the two objects to be related, and placing them in the corresponding relation places. For example, the situation "wrong-way driver disrelated from, close to, and heading towards road works" is modeled by tokens consisting of the objects wrong-way driver and road works placed in `Disrelated`, `Close`, and `FrontFront` (see Fig. 3).

Table 1: Ontology to CPN translation.

	Ontology concept	CPN concept
	A Situation Awareness Ontology is a tuple $SAW = (RC, OT, RT, ST, AC, RN, rcm, n, st, occ, rcd, idf)$, satisfying the requirements below.	CPN is a tuple $CPN = (\Sigma, P, T, A, N, C, G, E, I)$, a hierarchical CPN is a tuple $HCPN = (S, \ldots)$ satisfying the requirements below [15].
(i)	RC is a finite set of relation calculi	S is a finite set of pages, each one being a CPN
	$S = RC$	
(ii)	OT is a finite set of object types, ST is a finite set of situation types, such that $ST \subseteq OT$	Σ is a finite set of non-empty types, called color sets. C is a color function $C : P \rightarrow \Sigma$ E is an arc expression function $E : A \rightarrow expression$, such that $\forall a \in A : [Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \Sigma$
	The color set, color function and arc expression consists of object tuples (we only consider binary relations): $\Sigma = (OT \times OT)$ $E(a) = \begin{cases}(o_1, o_2) & \text{in every case}\end{cases}$ $C(p) = \begin{cases}(OT \times OT) & \text{in every case}\end{cases}$	
(iii)	RT is a finite set of relation types, $RT \cap OT = \emptyset$ rcm is a relation calculus membership function, $rcm : RT \rightarrow RC$	P is a finite set of places
	All relations of a family are added to the set of places of the family's corresponding page. $\forall s \in S \exists rc \in RC : P_s = \{r \in RT \mid rcm(r) = rc\}$, every relation is represented by a place $\bigcup P_i = \bigcup RT_i$	
(iv)	n is a neighborhood function $n : RT \rightarrow RT$, defining for each relation type a finite set of relation neighbors $\forall r \in RT : RN_r = \{r' \mid n(r) = r'\}$	T is a finite set of transitions. A is a finite set of arcs, $P \cap T = P \cap A = T \cap A = \emptyset$ N is a node function, $N : A \rightarrow P \times T \cup T \times P$
	For each pair of neighboring relation types, a transition with two arcs connecting the respective places exists: $\forall r, r' \in RT : n(r) = r' \Rightarrow \exists a_1, a_2 \in A, t \in T$ such that $N(a_1) = (r, t) \wedge N(a_2) = (t, r')$ Only one transition exists per pair of neighboring relation types: $\forall t, t' \in T : N(a) = (r, t) \wedge N(a') = (t, r') \wedge N(a'') = (r, t') \wedge N(a''') = (t', r') \Rightarrow t = t'$	
(v)	st is a situation type definition function, $st : (RT, (OT \times OT)) \rightarrow ST$	I is an initialization function $P \rightarrow expression$ such that $\forall p \in P : [Type(I(p)) = C(p)_{MS}]$
	Situation types become initial markings: $I(p) = \begin{cases}(o_1, o_2) & \text{if } \exists s \in ST, o_1, o_2 \in OT : st(p, (o_1, o_2)) = s \\ \emptyset & \text{otherwise}\end{cases}$	

In this section, we propose translations from object characteristics and relation interdependencies to Petri nets, thus promoting CPNs to SPNs.

Challenge A: Object characteristics exploited in guards. The first step in enhancing prediction quality, as shown in Fig. 4 and formalized in Tab. 2, aims to increase prediction precision by disabling wrong transitions between relations on the basis of object characteristics, such as permeability and moveability. In SPNs, (vi) object characteristics carried by two-colored tokens (assigned to objects by an object change characteristic function) are exploited in guards on transitions. These guards express optimization rules given by inherent characteristics of relations, such as `IsPermeable`, `IsMoveable`, and `IsScalable` [4], thereby defining firing conditions of transitions more precisely. For example, to determine whether the transition from `Disrelated` to `Partially Overlapping` should be disabled for a token even though it is placed in `Disrelated`, guard 1 in Fig. 4 checks whether either object can move. The information for this check is supplied by objects: The guard evaluates to true for `Wrong-way driver` objects, and hance the transition remains enabled. In contrast, guard 2 checks

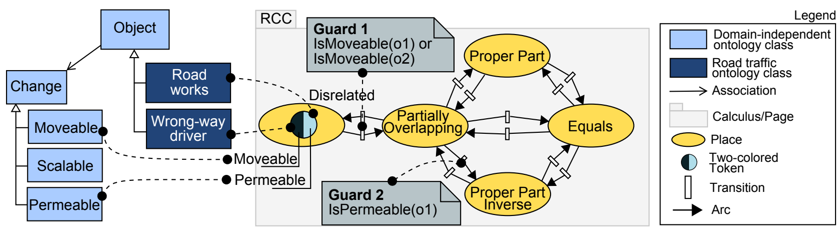

Fig. 4: Object characterististics exploited in guards.

whether the first object of the token has a permeable boundary (i. e., whether it allows the second object to enter its area), which is not the case for `Wrong-way drivers`. Consequently, the transition is disabled.

Table 2: Translations necessary for considering object characteristics.

	Ontology concept	CPN concept
(vi)	AC is a finite set of attribute change types (e. g., moveable) occ is an object change characteristic function, $occ : OT \rightarrow AC$ rcd is a relation change dependency function determining whether its from_o1 or to_o2 must fulfill the requirement, $rcd : RT \rightarrow AC$	G is a guard function, $G : T \rightarrow expression$, such that $\forall t \in T : [Type(G(t)) = \mathbb{B} \wedge Type(Var(G(t))) \subseteq \Sigma]$ $P \cup T$ is called the set of nodes Out maps each node to its output nodes, such that $Out(x) = \{x' \in X \mid \exists a \in A : N(a) = (x, x')\}$
	The transition is enabled, if the change dependencies of relation represented by a particular place are fulfilled by the incoming tokens. $G(t) = \begin{cases} \exists p \in Out(t) : p \in P \wedge rcd(p) \in occ(o_1) & \text{if } from\_o1(p) = o_1 \\ \exists p \in Out(t) : p \in P \wedge rcd(p) \in occ(o_2) & \text{if } from\_o1(p) = o_2 \\ false & \text{otherwise} \end{cases}$	

Challenge B: Relation interdependencies expressed by configurable dependency pages. In a world in which motion is continuous, relation interdependencies between different calculi may reduce the number of transitions between relations, as described in Sect. 2. These interdependencies, however, vary across different situations: For example, in the situation "Traffic jam disrelated from and close to road works", transitions in RCC depend on spatial distance, but in other cases, they may for instance depend on orientation. To tackle challenge B, we therefore need concepts both to represent interdependencies between relations of different calculi, and to achieve configurability on the basis of situations, eliminating the need for manually composed situation-dependent calculi. In order to keep the modeling effort low by eliminating the

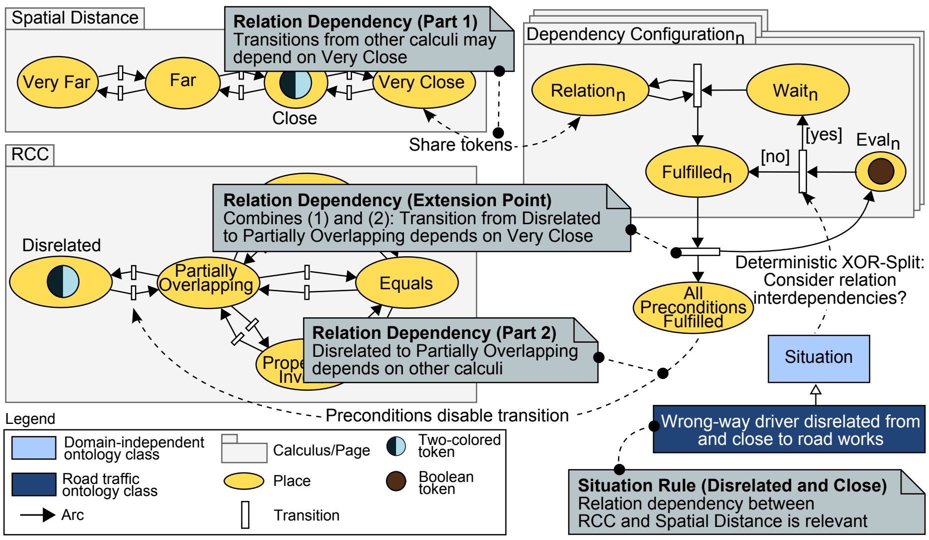

Fig. 5: Relation interdependencies expressed by configurable dependency pages.

need to create customized calculi, we split interdependencies into two parts, as shown in Fig. 5 and formalized in Tab. 3: The first part describes that other calculi may depend upon a particular relation (`Very Close` in our example), while the second part specifies that a particular transition (`Disrelated` to `Partially Overlapping` in the example) depends on certain preconditions being met (vii). A transition in-between combines these two parts and, in our example, forms the precondition "Transition from Disrelated to Partially Overlapping depends on Very Close". This transition also serves as an extension point for specifying additional dependencies, which are accumulated in the place `All Preconditions Fulfilled`. One may use this extension point, for instance, to add orientation information as a further precondition.

Now that we know how to represent dependencies between different calculi conceptually, let us turn our attention to making these dependencies configurable to accommodate different situations. For this purpose, we represent configurations on dedicated `Dependency Configuration` pages, making use of the Petri net design pattern "Deterministic XOR-Split" [21]. If the current situation makes it necessary to consider relation interdependencies (which is the case for the situation in our example), this design pattern results in a token being placed in $\texttt{Wait}_n$, meaning that we need to wait for a token to appear in $\texttt{Relation}_n$, in order to fulfill our precondition (i. e., place a token in $\texttt{Fulfilled}_n$). If the current situation does not consider relation interdependencies, this pattern fulfills the precondition directly (i. e., it is not relevant whether a token appears in $\texttt{Relation}_n$). In order to prevent tokens from accumulating in $\texttt{Fulfilled}_n$—which would make multiple transitions possible without actually evaluating the interdependencies—we follow the Petri net design pattern "Capacity Bounding" [21] with an anti-place $\texttt{Eval}_n$ restricting the capacity of $\texttt{Fulfilled}_n$ to 1.

Table 3: Translations necessary for considering relation interdependencies.

	Ontology concept	**CPN concept**
(vii)	idf is an interdependency function $idf : RT \rightarrow RT$, defining for each relaton type a finite set of depended-on relations $\forall r \in RT : RD_r = \{r' \mid idf(r) = r'\}$	A token is a pair (p, c) where $p \in P \wedge c \in C(p)$, a marking M is a multi-set over tokens in a CPN.
	In order to model Deterministic XOR Split, we extend the color set with type boolean: $\Sigma = (OT \times OT) \cup \mathbb{B}$ For each depended-on relation $n \in \bigcup RD_i$, we create a dedicated dependency configuration page: $P_n = \{Relation_n, Wait_n, Fulfilled_n, Eval_n\}$ $T_n = \{T_{n1}, T_{n2}\}$ $A_n = \{RT_{n1}, WT_{n1}, ET_{n2}, T_{n1}R, T_{n1}F, T_{n2}W, T_{n2}F\}$ $G_n(t) = \begin{cases} true \end{cases}$ $E_n(a) = \begin{cases} \exists s \in ST, p \in P, c \in C(p) : (p, c) \in M \wedge st(p, c) = s \text{ then true else empty} & \text{if } a = T_{n2}W \\ \forall s \in ST \exists p \in P, c \in C(p) : (c, p) \in M \Rightarrow st(p, c) \neq s \text{ then true else empty} & \text{if } a = T_{n2}F \\ (o_1, o_2) & \text{if } a \in \{RT_{n1}, T_{n1}R\} \\ b & \text{otherwise} \end{cases}$ $I_n(p) = \begin{cases} true & \text{if } p = Eval_n \\ \emptyset & \text{otherwise} \end{cases}$ We extend the guard functions on transitions to also check relation interdependencies. $G(t) = \begin{cases} \text{cases from above} & \\ true & \text{if } \exists (p, c) \text{ where } p = \text{All Preconditions Fulfilled} \wedge c \in \Sigma \\ false & \text{otherwise} \end{cases}$	

Challenges C and D: Distances derived from axiomatic mappings in code segments. Building upon the concepts of increasing precision introduced above, we describe methods for increasing expressiveness using distance descriptions. Let us recall the example from Sect. 2, in which temporal distance—"distant" if one object moves slowly or "soon" if both objects move fast—describes the transition from `Disrelated` to `Partially Overlapping` in more detail. For tackling challenges C and D, we need knowledge in the form of axioms, which map from object characteristics, like speed in our example, and from relations between objects to distance descriptions, as shown in Fig. 6. In road

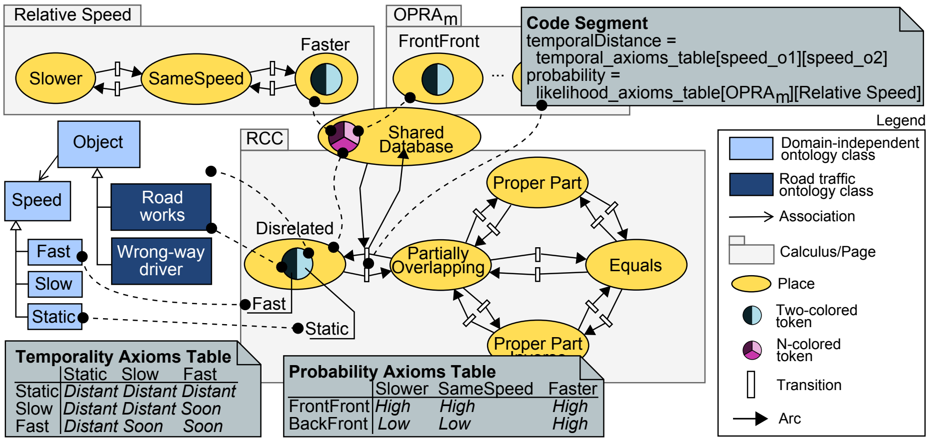

Fig. 6: Distances derived from axiomatic mappings in code segments.

traffic management, such axioms model domain knowledge providing rough estimations in the absence of real-world training data. If, in some other domain, such training data is available, learning from observed situation evolutions helps to further refine these axioms. Note that for simplicity, axioms are given in tabular form in this paper, but other representations, such as Hidden Markov Models and Bayesian nets, are also possible.

Using these axioms, code segments on transitions estimate distances in the prediction process. For this purpose, in addition to two-colored object tokens, the current marking in an SPN is reified as n-colored tokens (each color representing a particular place) and disseminated to transitions using the Petri net design pattern "Shared Database" [21]. The code segment in Fig. 6 derives estimations for temporal distance and probability, describing transitions between `Disrelated` and `Partially Overlapping` in more detail. Temporal distances are looked up in axiom tables using object characteristics (speed in our example), while probability is determined using the SPN's current marking.

5 Evaluation

In this section, we evaluate the applicability of SPNs using real-world data from the domain of road traffic management covering Austrian highways over a period of four weeks. These data were collected from multiple sources, such as traffic flow sensors, road maintenance schedules, and motorists reporting incidents to a call center. The recorded data set used for this evaluation consists of 3,563 distinct road traffic objects, comprising, among others, 778 traffic jams, 819 road works, 1,339 other obstructions, 460 accidents, and 64 weather warnings. As a proper starting point for situation prediction, we derived relations between traffic objects using our situation-awareness prototype BeAware! to detect situations that possibly require a human operator's attention. In order to restrict detected situations to those most relevant, we defined 13 situations in cooperation with the Austrian highways agency, of which three interesting ones[2] were selected for this evaluation. Table 4 lists these situations together with the characteristics of involved objects and the number of occurrences in our data set.

Table 4: Overview of situations that are starting points for predictions.

	Situation description and formalization, including object characteristics	#
Sit. 1	traffic jam close to another traffic jam (may merge) $TrafficJam(o1) \wedge TrafficJam(o2) \wedge Disrelated(o1, o2) \wedge Close(o1, o2) \wedge FrontBack(o1, o2)$ Traffic jam (o1): moveable, permeable, scalable, large, medium speed Traffic jam (o2): moveable, permeable, scalable, medium size, slow	17
Sit. 2	wrong-way driver heading towards road works (may cause an accident) $WrongWayDr.(o1) \wedge Roadworks(o2) \wedge Disrelated(o1, o2) \wedge Close(o1, o2) \wedge FrontFront(o1, o2)$ Wrong-way driver: moveable, non-permeable, small, fast Road works: permeable, large, static	10
Sit. 3	poor driving conditions (snow) in the area of road works (may evolve towards border) $PoorDrivingConditions(o1) \wedge Roadworks(o2) \wedge ProperPart(o1, o2) \wedge VeryClose(o1, o2)$ Poor driving conditions: moveable, permeable, medium size, slow Road works: permeable, large, static	2

Evaluation method. Based on the situations detected, we predicted possible future situations with SPNs. We discuss the predicted situations in the context of our major goals: We determined the effectiveness of increasing precision (challenges A and B) by comparing the resulting number of possible future situations to that derived from the respective unoptimized calculi. (H1: Optimizing calculi reduces the number of falsely predicted situations while retaining critical ones). The potential of distance descriptions for increasing expressiveness (challenges C and D) was evaluated by comparing the results to recorded real-world data, using duration and probability as example distances. (H2: Temporal distances and probabilities match real-world evolutions). It must be noted that, although covering a period of four weeks, the data with which we were provided were updated very infrequently. Hence, we obtained only a small number of observed evolutions. Although the first evaluation indicates that the approach we propose to challenges C and D is applicable, further (real-world) observations are needed to confirm H2. To this end, we are continuously extending our data set.

Evaluation setup. In our evaluation, we employed the guards `IsPermeable`, `IsMoveable`, and `IsScalable`, which use object characteristics, in conjunction with interdependencies between the calculi mereotopology, spatial distance, and

[2] Showing strengths, and shortcomings indicating potential improvements in SPNs.

size. In particular, transitions in RCC require objects to be very close to each other; transitions to `Proper Part (Inverse)` and `Equals` check relative sizes; spatial distances when being `Partially Overlapping`, `Proper Part (Inv.)`, or `Equals` are restricted on the basis of object size to increase precision. For increasing expressiveness, we used temporal axioms mapping object speed to durations (on the basis of domain bindings defining spatial distances), as well as probability axioms mapping orientation of objects towards each other to probabilities. Figure 7 summarizes the achieved prediction quality enhancement. It

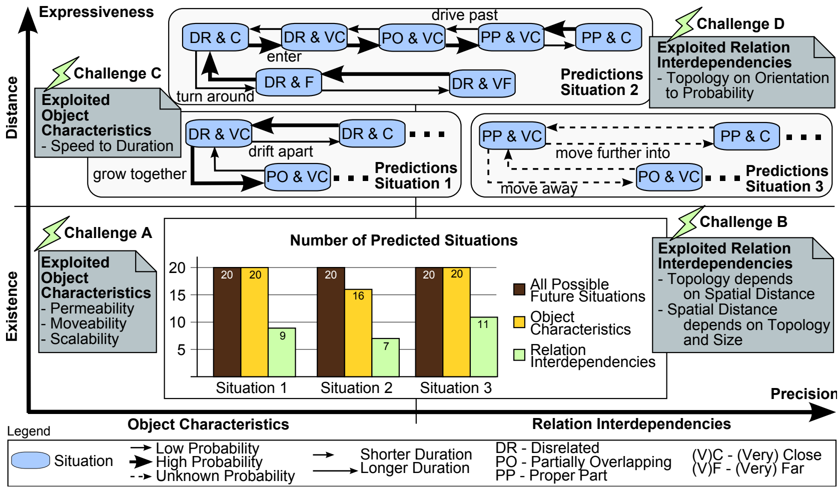

Fig. 7: Evaluation overview.

can be seen that object characteristics alone are not effective, but in combination with relation interdependencies they exclude many false predictions, and that the predicted distances correspond to the expected overall evolution. Below, we discuss these results in more detail.

Situation 1: Traffic jams close to each other. In the first situation, two traffic jams are very close (about 0.5km apart, as stated in the real-world data), but still disrelated from each other (meaning that they will probably merge) with the rear one being larger, and growing faster, than the front one (real-world data, cf. Table 4). Traffic jams, which can move, scale, and have permeable boundaries, do not allow us to exclude situations by inspecting object characteristics. In this case, prediction precision can only be increased by additionally taking relation interdependencies—as described above—into account, reducing the number of predicted situations to nine (excluding evolutions such as traffic jams partially overlapping but far from each other). In these predicted situations, two crucial evolutions are preserved: the traffic jams may drift apart or merge (confirms H1). Distances for duration and probability are based on the current motion of traffic jams and state that merging is more likely than drifting apart, although it will take a considerable amount of time. This prediction was confirmed by our data set, which showed that the observed traffic jams indeed merged into a single large one about 90 minutes after detection.

Situation 2: A wrong-way driver heads towards road works. In the second scenario, a wrong-way driver (small, fast) is detected to head towards road works (large, static). The fact that a wrong-way driver's boundary is not permeable, discards predictions involving `Proper Part Inverse` and reduces their number from 20 to 16 solely on the basis of object characteristics. With relation interdependencies, a further reduction to 7 (size relationship between the two objects) results in the following predicted evolutions: the wrong-way driver may enter and then drive past the area of road works, or he/she may turn around (never observed in our data set). The most likely scenario is that, due to the wrong-way driver's current orientation and speed, he/she may enter the area of road works. This prediction partially matches our data set, in which, ten minutes after being detected, the wrong-way driver entered the area of road works (becoming `Proper Part` in accordance with our prediction) and then, luckily and against all odds, managed to drive past the road works (which was assigned only low probability).

Situation 3: Snow in the area of road works. In our final scenario, poor driving conditions (a medium-sized area of snowfall) are detected within a large area of road works. The non-deterministic nature of weather conditions, together with the limited amount of information currently available to our system (e. g., directions of weather movements are not provided), makes it impossible to exclude situations on the basis of object characteristics. It also does not allow us to exclude a large number of evolutions when considering relation interdependencies, and makes deriving probabilities impossible (no direction given). Only approximate durations (distant and very distant) may be given on the basis of the domain knowledge that weather conditions typically change slowly. These predicted durations are consistent with observations in our real-world data reporting poor driving conditions over a period of three hours in one case and over about a day in another case.

6 Lessons Learned and Future Work

In this section, we present lessons learned from implementing and using Situation Prediction Nets and, based on these findings, indicate directions for future work.

Object characteristics are only effective in combination with relation interdependencies. Situations described with objects and relations are the basis for neighborhood-based predictions in situation awareness. The potential for enhancing prediction quality when using object characteristics in isolation may, depending on the domain, therefore be rather limited (as shown for road traffic management by our evaluation). Only in combination with relation interdependencies, one can achieve substantial improvements in such a case.

CPNs are suitable for deriving, but not for keeping track of, distance descriptions. Distance descriptions must be retained for later examination by human operators. In a naive approach, distances are attached to tokens or accumulated in dedicated places, resulting in prediction state space being no longer bounded by the number of possible combinations between relations. Persistent storage outside the CPN seems to be better suited to preventing this.

Distances in axiomatic mappings should be learned. Distances and probabilities modeled a-priori in axiomatic mappings are only a starting point for the system. In order to keep them up-to-date, a learning component should analyze events occurring in the domain (e. g., to learn that something that was considered unlikely actually occurs more often than assumed).

Recursive aggregation of situations in predictions facilitates re-use. In our previous work [5], we encouraged re-use of situations as objects in recursively defined higher-level situations. For example, a wrong-way driver could head towards road works in snowfall. We could re-use the situation "Poor driving conditions in the area of road works", and relate the wrong-way driver with this situation. SPNs, therefore, need to be extended with concepts for aggregating simulations of one net into the simulation of another one, for instance by representing the marking of one Petri net in a two-colored token of another one.

Predictions using partial information require planning. Neighborhood-based prediction assumes that starting points for predictions already comprise all relevant objects. This is particularly problematic if causal relations between objects are described. For example, the emergence of the critical situation "Accident causes traffic jam" should clearly be indicated by the corresponding initial situation—the occurrence of an accident. However, the absence of relations prevents our current approach from predicting any situation in this case. We intend to extend SPNs with ideas from qualitative planning [22]. Critical situations could then be represented as goals, and the planning approach should yield the necessary steps (e. g., emergence of a traffic jam) for reaching them.

References

1. K. R. Apt and S. Brand. Constraint-based qualitative simulation. In *Proc. of the 12th Intl. Symp. on Temporal Rep. and Reasoning*, pages 26–34. IEEE, 2005.
2. C. Bailey-Kellogg and F. Zhao. Qualitative spatial reasoning - extracting and reasoning with spatial aggregates. *AI Magazine*, 24(4):47–60, 2003.
3. J. Barwise and J. Perry. *Situations and Attitudes.* MIT Press, 1983.
4. N. Baumgartner, W. Gottesheim, S. Mitsch, W. Retschitzegger, and W. Schwinger. On optimization of predictions in ontology-driven situation awareness. In *Proc. of the 3rd Intl. Conf. on Knowl., Science, Eng. and Mgmt.*, pages 297–309, 2009.
5. N. Baumgartner, W. Gottesheim, S. Mitsch, W. Retschitzegger, and W. Schwinger. BeAware!—situation awareness, the ontology-driven way. *Accepted for publication in: Data and Knowledge Engineering (Journal)*, 2010.
6. N. Baumgartner, W. Retschitzegger, W. Schwinger, G. Kotsis, and C. Schwietering. Of situations and their neighbors—Evolution and Similarity in Ontology-Based Approaches to Situation Awareness. In *Proc. of the 6th Intl. and Interdis. Conf. on Modeling and Using Context*, pages 29–42, Roskilde, Denmark, 2007. Springer.
7. M. Bhatt, W. Rahayu, and G. Sterling. Qualitative simulation: Towards a situation calculus based unifying semantics for space, time and actions. In *Proc. of the Conf. on Spatial Information Theory*, Ellicottville, NY, USA, 2005.

8. F. Bonchi, A. Brogi, S. Corfini, and F. Gadducci. Compositional specification of web services via behavioural equivalence of nets: A case study. In *Proc. of the 29th Intl. Conf. on Application and Theory of Petri Nets and other models of concurrency*, pages 52–71, Xi'an, China, 2008. Springer.
9. A. G. Cohn and J. Renz. *Handbook of Knowledge Representation*, chapter Qualitative Spatial Representation and Reasoning, pages 551–596. Elsevier, 2008.
10. Z. Cui, A. G. Cohn, and D. A. Randell. Qualitative simulation based on a logical formalism of space and time. In *Proc. AAAI-92*, pages 679–684. AAAI Press, 1992.
11. F. Dylla and J. O. Wallgrün. On generalizing orientation information in OPRA_m. In *Proc. of the 29th Annual German Conference on AI*, pages 274–288, Bremen, Germany, 2007. Springer.
12. F. Dylla and J. O. Wallgrün. Qualitative spatial reasoning with conceptual neighborhoods for agent control. *Intelligent Robotics Systems*, 48(1):55–78, 2007.
13. C. Freksa. Conceptual neighborhood and its role in temporal and spatial reasoning. In *Proc. of the Imacs Intl. Workshop on Decision Support Systems and Qualitative Reasoning*, pages 181–187, 1991.
14. Z. M. Ibrahim and A. Y. Tawfik. An abstract theory and ontology of motion based on the regions connection calculus. In *Proc. of the 7th Intl. Symp. on Abstraction, Reformulation, and Approx.*, pages 230–242, Whistler, Canada, 2007. Springer.
15. K. Jensen. *Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use*. Springer, 2nd edition, 1997.
16. K. Jensen, L. M. Kristensen, and L. Wells. Coloured petri nets and CPN Tools for modeling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9:213–254, 2007.
17. M. M. Kokar, C. J. Matheus, and K. Baclawski. Ontology-based situation awareness. *International Journal of Information Fusion*, 10(1):83–98, 2009.
18. S. Laborie. Spatio-temporal proximities for multimedia document adaptation. In *Proc. of the 12th Intl. Conf. on Artificial Intelligence: Methodology, Systems, and Applications*, pages 128–137, Varna, Bulgaria, 2006. Springer.
19. B. Li and F. Fonseca. TDD - a comprehensive model for qualitative spatial similarity assessment. *Journal on Spatial Cognition & Computation*, 6(1):31–62, 2006.
20. G. Ligozat. Towards a general characterization of conceptual neighborhoods in temporal and spatial reasoning. In *Proceedings of the AAAI-94 Workshop on Spatial and Temporal Resoning*, pages 55–59, Seattle, WA, USA, 1994. AAAI.
21. N. Mulyar and W. van der Aalst. Towards a pattern language for colored petri nets. In *Proc. of the 6th Workshop on Practical Use of Coloured Petri Nets and the CPN Tools*, pages 39–58, Aarhus, Denmark, 2005. University of Aarhus.
22. M. Ragni and S. Wölfl. Temporalizing cardinal directions: From constraint satisfaction to planning. In *Proc. of 10th Intl. Conf. on Principles of Knowledge Representation and Reasoning*, pages 472–480. AAAI Press, 2006.
23. Randell, D.A., Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning*, pages 1–34. Morgan Kaufmann, 1992.
24. W. Reisig and G. Rozenberg. *Lectures on Petri Nets I: Basic Models - Advances in Petri Nets*, chapter Informal intro. to Petri nets, pages 1–11. Springer, 1998.
25. N. van de Weghe and P. D. Maeyer. Conceptual neighborhood diagrams for representing moving objects. In *Proceedings of the ER Workshop on Perspectives in Conceptual Modeling*, pages 228–238, Klagenfurt, Austria, 2005. Springer.
26. G. Zhang, F. Meng, C. Jiang, and J. Pang. Using petri net to reason with rule and owl. In *Proc. of the 6th Intl. Conf. on Computer and Information Technology*, pages 42–47, Seoul, Korea, 2006. IEEE.