Logic-Based Modeling Approaches for Qualitative and Hybrid Reasoning in Dynamic Spatial Systems

STEFAN MITSCH and ANDRÉ PLATZER, Carnegie Mellon University
WERNER RETSCHITZEGGER and WIELAND SCHWINGER, Johannes Kepler University

Autonomous agents that operate as components of dynamic spatial systems are becoming increasingly popular and mainstream. Applications can be found in consumer robotics, in road, rail, and air transportation, manufacturing, and military operations. Unfortunately, the approaches to modeling and analyzing the behavior of dynamic spatial systems are just as diverse as these application domains. In this paper, we discuss reasoning approaches for the medium-term control of autonomous agents in dynamic spatial systems, which requires a sufficiently detailed description of the agent's behavior and environment, but may still be conducted in a qualitative manner. We survey logic-based qualitative and hybrid modeling and commonsense reasoning approaches w.r.t. their features for describing and analyzing dynamic spatial systems in general, and the actions of autonomous agents operating therein in particular. We introduce a conceptual reference model, which summarizes the current understanding of the characteristics of dynamic spatial systems based on a catalog of evaluation criteria derived from the model. We assess the modeling features provided by logic-based qualitative commonsense and hybrid approaches for projection, planning, simulation, and verification of dynamic spatial systems. We provide a comparative summary of the modeling features, discuss lessons learned, and introduce a research roadmap for integrating different approaches of dynamic spatial system analysis to achieve coverage of all required features.

Categories and Subject Descriptors: I2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Representation languages*; I2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving—*Deduction*

General Terms: Languages, Theory, Algorithms

Additional Key Words and Phrases: Autonomous agents, logic-based reasoning, commonsense reasoning, dynamic reasoning, dynamic spatial systems, knowledge representation, hybrid systems

ACM Reference Format:
Stefan Mitsch, André Platzer, Werner Retschitzegger, and Wieland Schwinger, 2013. Logic-Based Modeling Approaches for Qualitative and Hybrid Reasoning in Dynamic Spatial Systems. *ACM Comput. Surv.* 48, 1, Article 3 (August 2015), 41 pages.
DOI:http://dx.doi.org/10.1145/2764901

This work has been funded partly by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under grants FFG FIT-IT 829598, FFG BRIDGE 832160 and 838526, FFG COIN 845947, by ÖAD/WTZ grants AR18/2013, UA07/2013, and AR10/2015, by ERC grant PIOF-GA-2012-328378, by FWF grant P 28187-N31, by NSF CNS 1035800, and by US DOT grant DTRT12-G-UTC11.

Author's addresses: S. Mitsch and A. Platzer, Computer Science Department, School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh PA, 15213, USA; W. Retschitzegger and W. Schwinger, Department of Cooperative Information Systems, Johannes Kepler University, Altenberger Str. 69, 4040 Linz, Austria.

DOI:http://dx.doi.org/10.1145/2764901

1. INTRODUCTION

Dynamic systems are systems whose states change over time [Sandewall 1994]. Dynamic spatial systems are a subclass thereof that can be primarily described in terms of the evolution of spatial states [Worboys 2001]. In such dynamic spatial systems, physical entities (e. g., infrastructure that cannot actively influence its evolution) and autonomous agents (e. g., autonomous vehicles that can actively change their behavior) may be present and evolve according to their spatial states (e. g., traffic participants change their positions according to their current velocity). Domains, which are concerned with autonomous agents in dynamic spatial systems, are anchored in the broad field of robotics and, for instance, include such diverse application areas as road and rail traffic management (e. g., autonomous vehicles [Urmson et al. 2008], traffic centers [Mitsch et al. 2012], and train control [Platzer and Quesel 2009]), aeronautics (e. g., aerial drones [Bachrach et al. 2009]), manufacturing (e. g., transportation robots [Rosenthal et al. 2010; Mitsch et al. 2013]), medical equipment [Lee et al. 2012], and consumer equipment robots (e. g., vacuum cleaning robots).

Whatever the concrete application domain at hand, autonomous agents would not be particularly useful if they made blatantly incorrect control decisions, possibly even endangering safety within a dynamic spatial system. In this paper we present a *survey of modeling concepts in logic-based qualitative and hybrid reasoning* about agent behavior in a dynamic spatial system.

Article Focus. We detail the focus of this survey along the different control tasks of autonomous agents. Autonomous agents are equipped with *controllers* to adjust their own behavior w.r.t. that of other entities and agents in the system[1]. They have to solve control tasks for different time horizons [Albus and Meystel 1996]. These control tasks—which, for instance, can be dealt with in a hierarchical control structure [Russell and Norvig 2003]—range from

- long-term strategic decisions (e. g., navigation to arrive at a destination address, typically in the magnitude of minutes to hours) over
- medium-term motion control (e. g., motion planning to turn at an intersection, typically in the magnitude of several seconds to minutes) to
- short-term computation of set-values for actuators (e. g., controllers to adjust acceleration and steering angle of a vehicle, in the magnitude of milliseconds to seconds).

In this article, we focus on the medium-term motion control tasks and their link to computing short-term set values within a dynamic spatial system. We do not focus on long-term strategic decisions here, since their effects are often too vague in a partially observable environment to ensure safety under all conditions.

Design-time techniques that aim to increase or even guarantee safety comprise (i) *simulation* of evolution in a dynamic spatial system and (ii) *verification* of the correctness of an autonomous agent. At run-time, an autonomous agent typically uses techniques to (iii) check *consistency* of sensed information to establish awareness about the current situation, and (iv) *project* the behavior of other agents to *plan* its own steps.

These design-time and run-time techniques range from purely qualitative approaches (discrete control and evolution, e. g., [Bhatt 2012; Ragni and Wölfl 2006]) and hybrid approaches (typically discrete control and continuous evolution, e. g., [Alur et al. 1995; Goebel et al. 2009; Henzinger 1996; Platzer 2010b]), to purely quantitative approaches (continuous control and evolution).

---

[1] Relevant research influencing robotics can be found in various areas, such as control theory [Aström and Murray 2008], artificial intelligence (e. g., expert systems [James 1987], decision support systems [Arnott and Pervan 2005]), and cognitive robotics [Shanahan 2000; Levesque and Lakemeyer 2008].

Diversity of Qualitative and Hybrid Approaches. Autonomous agents in dynamic spatial systems have already been investigated w.r.t. several different aspects and for different goals. Historically, approaches were first concerned with providing *static* modeling concepts (i. e., modeling *state*). This static focus distinguishes entities in dynamic spatial systems as either being physical (e. g., a vehicle) or abstract (e. g., a spatial region), and describes their properties and relationships between entities. In such models, it is important to check consistency of multiple (partial) state descriptions, resolve ambiguity, and compensate for incomplete information [Cohn and Renz 2008].

Later, the static viewpoint was extended with *dynamic* concepts for describing the events that may occur and the actions that can be initiated by entities in a dynamic spatial system. These dynamic concepts were combined with *epistemic* knowledge on the influences between states, and the necessary and sufficient conditions of events and actions that lead to evolution between states. The most interesting reasoning techniques in such models are planning and projection (e. g., [Bhatt 2012; Ragni and Wölfl 2006]), simulation (e. g., [Apt and Brand 2005]), and verification (e. g., [Henzinger 1996; Platzer 2010b]) of agent behavior.

Together, the static, dynamic, and epistemic views determine the *situation awareness* [Endsley 2000] of an autonomous agent or, more generally, the *shared* situation awareness [Stewart et al. 2008] of the entities in a dynamic spatial system. However, as a consequence of this diversity of modeling and reasoning approaches, we face not only different terminologies, but also a broad variety of modeling principles, notations, and reasoning algorithms.

Contributions. This paper presents a survey of modeling concepts of existing *logic-based qualitative* and *hybrid* reasoning approaches for autonomous agents in dynamic spatial systems. It emphasizes comparability of approaches on the basis of a detailed evaluation framework in order to facilitate a deeper understanding of commonalities and differences between existing terminology, modeling concepts, notations, and reasoning algorithms, so that researchers and practitioners can select the *right approach for the right purpose at the right level of abstraction*. With its focus on logic-based modeling approaches for qualitative and hybrid reasoning in dynamic spatial systems, this survey complements previous surveys on

- the family of (hybrid) process algebras (e. g., [Baeten 2005; Groote and Reniers 2001; Khadim 2008]),
- temporal (description) logics (e. g., [Artale and Franconi 2001; Emerson 1990; Konur 2013; Lutz et al. 2008]),
- purely algebraic approaches in geographical information systems (e. g., [Worboys 2005],
- model checking and simulation of hybrid systems (e. g., [Alur 2011; Casagrande and Piazza 2012; De Schutter et al. 2009]), and
- quantitative agent modeling (e. g., [Allan 2010; Heath et al. 2009; Nikolai and Madey 2009; Serenko and Detlor 2003])

In summary, the main contributions of this article are (i) a conceptual reference model for describing state and behavior of physical entities in dynamic spatial systems, (ii) a catalog of evaluation criteria based on the conceptual reference model, and (iii) an evaluation of logic-based modeling approaches for qualitative and hybrid reasoning in dynamic spatial systems, including a comparative summary.

Article Structure. This article is structured as follows. Sect. 1 introduces a conceptual reference model for dynamic spatial systems. Appendix A lists a UML class diagram of the complete reference model, and Appendix B complements the reference model with a summary of qualitative relation calculi and their features. Appendix C further illustrates the conceptual reference model as a modeling framework by means of examples in road traffic control and autonomous robotics. Sect. 3 turns the conceptual reference model into an evaluation framework, which is the basis for a comparative summary in Sect. 4. Appendix D provides a detailed survey of each approach, including a syntax summary and modeling examples. Finally, Sect. 5 concludes the article with a research roadmap for reasoning in dynamic spatial systems.

2. A CONCEPTUAL REFERENCE MODEL FOR DYNAMIC SPATIAL SYSTEMS

Diverse research communities contribute to logic in robotics: for example, artificial intelligence, spatio-temporal reasoning, situation awareness, data fusion, geographic information systems, simulation, and formal verification. Among other things, these communities all deal in one form or the other with modeling of and reasoning about the behavior of agents in dynamic spatial systems. In this section, we want to establish a common understanding of the different terminologies and conventions used in these communities by means of a conceptual reference model for dynamic spatial systems. While some of the terms are generally agreed upon, less agreement has been established for others. We therefore discuss the various viewpoints found in the literature, which form the basis of our conceptual model. The benefits and use cases of such a conceptual reference model include (i) identification of similar concepts, which are the prerequisite for getting approaches to work with each other and for combining multiple solutions, (ii) detection of concepts that are not yet present, which points to interesting further research opportunities, and (iii) provision of a basic modeling framework for dynamic spatial systems, which allows modeling independent of a particular approach.

The rationale behind the design of our reference model is to integrate concepts from the aforementioned research communities and domains with concepts from existing surveys on requirements (e. g., for modeling of and reasoning about dynamic spatial systems [Bhatt 2009; 2010], qualitative spatial reasoning techniques [Cohn 1997; Cohn and Hazarika 2001], and situation awareness [Baumgartner and Retschitzegger 2006]). The conceptual reference model enables us to explain the basic constituents of a dynamic spatial system and their inter-dependencies in terms of a graphical representation as a UML class diagram for easy accessibility, as well as in terms of a glossary that comprises a textual definition for each concept introduced in the class diagram. In the following paragraphs, we first introduce the major building blocks of our conceptual reference model, before each of these building blocks is detailed with a dedicated UML class diagram. Naturally, the conceptual reference model also serves as a modeling framework[2], which can be extended by means of sub-classing if further concepts must be captured. At the same time, the conceptual reference model shall serve as a basis for deriving evaluation criteria for our subsequent survey.

Overview. In the current literature, one can already find quite a large body of concepts necessary to capture information about entities and the ways in which evolution of these entities may occur. The conceptual reference model is designed in a modular manner to address varying modeling needs in four packages (see Appendix A for a single integrated model): The universe of discourse (cf. UniverseOfDiscourse) provides a vocabulary to define the properties of physical entities of a world [Niles and Pease 2001], and to relate these physical entities to notions of time and space. The static view (cf. StaticView) captures what is true at an instant of time or throughout an interval of time, whereas the dynamic view (cf. DynamicView) describes what happened between the true states, see also the SNAP/SPAN ontology [Grenon and Smith 2004]. Finally,

---

[2] It roughly builds on our previously introduced SAW task ontology [Baumgartner et al. 2010a].

the epistemic view (cf. EpistemicView) captures what an agent knows or believes to be true. Epistemic knowledge about state and evolution enables deductive reasoning (planning, projection, and verification) and simulation [Bhatt 2009; 2010].

The partition of responsibilities that we follow is in line with classical results from logic, yet specialized to the case of dynamic spatial systems here. The universe of discourse characterizes the entities, their properties, and their relations as they are relevant for modeling a dynamic spatial system. The static view characterizes what can be modeled and expressed to hold about the system, e.g., at a single state at one particular moment in time (as in first-order logic [Fitting and Mendelsohn 1999]). The dynamic view characterizes in what way the behavior of how states change over time can be expressed (see modal logics [Fitting and Mendelsohn 1999], temporal logics [Prior 1957; Pnueli 1977], and dynamic logics [Pratt 1976]). And the epistemic view characterizes what can be modeled about what agents know or do not know or believe or do not (as in epistemic logics [Von Wright 1951; Hintikka 1962]).

In Sections 2.1 to 2.4 we discuss these packages in detail. We start each package with definitions according to this article, before we discuss alternatives and different terminology found in the literature that led to these definitions. The complete conceptual reference model in a single integrated model can be found in Appendix A. To illustrate the concepts in the reference model we will use a sample scenario from an intuitively accessible domain, i. e., road traffic. This example is inspired by the Cooperative Intersection Collision Avoidance System (CICAS [Misener et al. 2010]) and described in detail in Appendix C.1. We introduce an additional example from robotics in Appendix C.2 to illustrate the applicability of the conceptual reference model to different domains. In order to avoid collisions at intersections, CICAS informs a so-called subject vehicle about possible hazards (e. g., risk of upcoming red light violation or other vehicles approaching the intersection with high velocity). To this end, CICAS exchanges information not only between an intersection and the approaching vehicles, but also in-between vehicles.

2.1. Universe of Discourse

Definition 2.1 (*Universe of Discourse*). The universe of discourse (cf. Figure 1) comprises *entities*, which are either *physical* in nature or *abstract*. Physical entities can have *properties*, which are either *constant* (non-changeable) or *fluent* (changeable). Physical entities with exclusively constant properties are called constant entities. All others are evolvable entities; some of them are *agents* with a free will. Abstract entities structure the physical appearance according to some mental abstraction. We consider especially two kinds of abstract entities: *temporal* entities, such as instants and intervals, and *spatial* entities, such as points and regions.

We anchor our subsequent discussion about the static and dynamic nature of spatial systems in this definition of a *universe of discourse*. The universe of discourse takes a similar role as the domain of discourse in many-sorted first-order logic. Its concepts are rooted in the qualitative theories for moving objects [Galton 1995; 2000], which are one of the historically first research approaches to qualitatively describe dynamic spatial systems. These qualitative theories for moving objects postulate the importance of a formal definition of a *theory of time*, *space*, *objects*, and *positions* (which maps objects into time and space) [Galton 2000] for representing behavior in dynamic spatial systems. We consider these four parts important for structuring the universe of discourse of dynamic spatial systems. However, we define objects and positions, corresponding to Niles and Pease [2001], in a broader sense of *physical entities* and *properties*, respectively. This allows us to consider additional aspects of interest (e. g., compositions of objects, such as traffic jams, or non-spatial properties, such as severity).

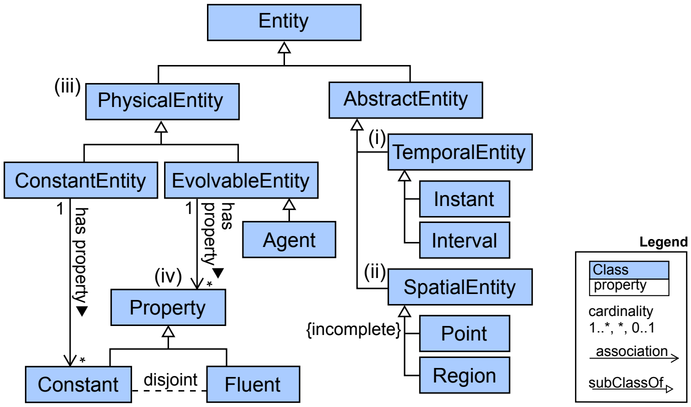


Fig. 1: The universe of discourse package (UoD)

The theory of physical entities extends the theory of objects to comprise abstract immaterial entities too, such as events and actions. This is especially important to describe virtual composite entities (e. g., a traffic jam, as opposed to every single car in the jam) and to let an agent reason about its own actions (e. g., is it safe to accelerate near an accident). The theory of properties considers evolution beyond motion, which allows us, for instance, to describe shape transformation [Davis 2001]. It maps physical entities not only into the temporal and spatial domain, but also into various discrete or dense, bounded or unbounded, as well as finite or infinite valued spaces. Each such space describes a particular aspect of a physical entity, such as its shape or color.

In a dynamic spatial system, spatial and temporal continuity is assumed, which means that evolution (e. g., in terms of motion) is modeled as a continuous function of time [Delafontaine et al. 2011]. Yet, our model of a current state is a discrete abstraction built by sampling the properties of physical entities (e. g., measuring the real position of a car) and by mapping them to abstract entities (e. g., a GPS point), possibly with sensor noise.

The UniverseOfDiscourse package describes the necessary concepts for abstracting from the observed world: it provides concepts for describing and extending the universe of discourse with (i) a theory of time and (ii) space, (iii) a theory of physical entities (i. e., objects), and (iv) a mapping of these entities into the theory of time and space in terms of describing their properties. Note that the universe of discourse only defines a vocabulary for describing state and dynamics. For example, statements about an agent existing in point p at time instant t are part of the static view described later.

A Model of Time and its Entities. In accordance with Galton [1995], a theory of time comprises a notion of *temporal entities* (cf. TemporalEntity) in terms of time instants (cf. Instant) and intervals (cf. Interval). These act as locations in the temporal dimension to describe the time for which a state description holds and duration of events, actions and processes. A reasoning technique that abstracts time to instants is unable to capture interleaving patterns when agents act concurrently in a dynamic spatial system (e. g., our own vehicle waits at the intersection *while* another vehicle passes the intersection—both extend over time intervals); one with only proper intervals cannot capture instantaneous events precisely, which, for instance, gives rise to problems in simulation and verification.

A Model of Space and its Entities. Similar to time, space comprises a notion of *spatial entities* (cf. SpatialEntity), for instance points (cf. Point), lines, and two- or three-dimensional regions (cf. Region) such as polygons and polyhedra, which act as locations in the spatial domain for different kinds of observable entities. For expressiveness reasons, one would like to have as many different kinds of spatial entities as possible. Reasoning complexity, however, typically increases with more complex spatial representations. For example, formal verification of autonomous vehicles often approximates objects as points [Loos et al. 2011; Mitsch et al. 2012; Mitsch et al. 2013]. The assumption is, that if safety cannot be guaranteed with a simple representation, then behavior of the more complex real system is even worse. Also note that additional computation may be necessary to turn low-level measurements into complex spatial entities. For example, laser scanners for measuring distance to obstacles in the environment often deliver sets of points; extracting shapes of different objects from a set of points is a computationally intensive process that we may want to avoid in an autonomous robot.

Physical Entities and their Properties. In accordance with Galton and Worboys [2005], we distinguish between two types of *physical entities* [Niles and Pease 2001] in a world (cf. PhysicalEntity): those entities that, once they exist, are constant and never subject to evolution until they cease to exist (cf. ConstantEntity), and those that are able to evolve during their lifetime (cf. EvolvableEntity). In principle, evolution may concern any kind of fluent property (e. g., position, color, or age). But with our focus on dynamic spatial systems, we restrict our discussion to spatial properties. Agents (cf. Agent as a subclass of evolvable entities) evolve as a result of their free-will decision making. If a technique is able to recognize other agents, it may become possible to negotiate a joint behavior for achieving goals.

Constant as well as evolvable entities may be characterized by certain *properties* (cf. Property, e. g., positions, lengths, or distances) [Kokar et al. 2009]. These properties again can be discerned into *constant* or rigid [Beckert and Platzer 2006] properties (cf. Constant, e. g., the position of the left-turn lane), whereas others can change and are therefore called *fluent* [Reiter 2001], *variable* [Galton and Worboys 2005], or non-rigid [Beckert and Platzer 2006] (cf. Fluent, e. g., the position of the subject vehicle). Constant entities can have constant properties only (otherwise, they would not actually be constant), whereas entities that can evolve may have properties of both kinds.

The fact that constant entities never change makes them candidates for design-time optimization (e. g., encode their position on a map or other constant facts in the knowledge base). The state of evolvable properties, however, must be sensed or communicated dynamically.

Knowledge representation about physical entities can either emphasize their states captured as a single sequential snapshot, which is the focus in the next section (static view), or, instead, emphasize the evolution that occurs between these states over time, which is the focus in Sect. 2.3 (dynamic view).

2.2. Static View

The static view captures what is true about physical entities at a time instant or throughout a time interval. The static view, compared to first-order logic, captures what is a state, what are the values of variables in a state, and what are the truth values of (interpreted) predicates. Such information is typically obtained through sensor measurements (e. g., measured with external, on-board or wearable sensors or sensor networks of an autonomous agent, e. g. [Kurschl et al. 2009]). We assemble Def. 2.2 and Figure 2 from [Grenon and Smith 2004; Bhatt and Loke 2008; Bittner 2002; Worboys and Hornsby 2004; Dylla and Bhatt 2008; Barwise and Perry 1983; Galton 1995] and relate the concepts to others from the literature.

Definition 2.2 (*Static View*). A constant property can have exactly one state, while a fluent one may assume multiple states. Each state is associated with a temporal entity that defines when it holds (e. g., valid during a particular interval, or at a particular instant). We distinguish between *unary* states that capture a value of a single entity, and *n-ary* states that compare two or more entities. An important unary state in a dynamic spatial system captures the *position* of a physical entity. A *situation* is an event that is characterized by one or more states, often (but not necessarily) with an emphasis on n-ary states. We emphasize n-ary states of temporal nature to compare temporal entities (e. g., i during i') and of spatial nature to compare spatial entities (e. g., r inside r').

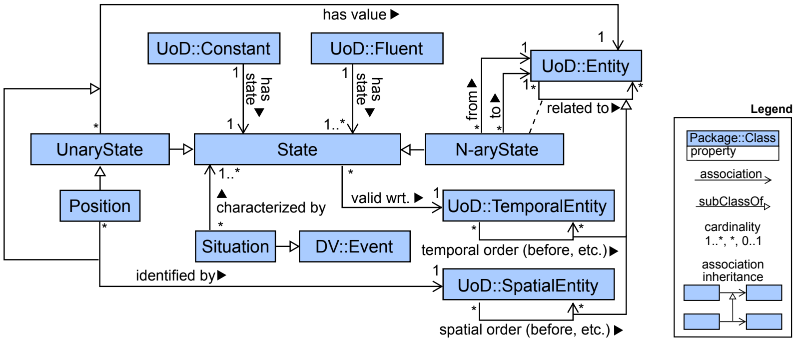


Fig. 2: Unary and n-ary states describe situations in the static view

Property States. Constant properties of physical entities never change. Therefore, constant properties have a *single state* (cf. State) only, which will be the same in all subsequent snapshots of the system. During the life of evolvable entities, their fluents, in contrast, may be subject to many changes while their constant properties stay unchanged. When we retain all previous states in addition to the most recent one, we can describe the history of an entity. This distinction between properties and their states is in accordance with the influential SNAP/SPAN ontology [Grenon and Smith 2004]. It allows us to discern entities and their life in terms of evolving states of properties.

State changes are in practice sampled at particular time instants regardless of the continuous or discrete nature of a fluent. Between sampling points, the state of a fluent is thus often considered to be stable w.r.t. a particular temporal entity (typically during a particular temporal interval, cf. State valid w.r.t. TemporalEntity). For safety guarantees, however, this simplification is at odds, because critical states could be missed by an overly coarse sampling. Hybrid approaches, therefore, model time in a continuous manner. We bridge qualitative and hybrid approaches as follows: in a hybrid approach, we explicitly model sampling intervals to guarantee safety for all possible times while the resulting constraints are still discrete for qualitative approaches.

States can be of either unary or n-ary nature [Bhatt and Loke 2008]. A unary state (cf. UnaryState) specifies a single entity (e. g., the velocity of the subject vehicle), while a n-ary state (cf. N-aryState) relates two or more entities to each other (e. g., the position of a vehicle relative to a lane). States embed physical entities in the theories of time and space, and thus correspond to the notion of a *setting* in the geospatial event model of Worboys and Hornsby [2004]. Since evolution is prevalent in dynamic spatial systems, unlike Worboys and Hornsby [2004], we neither consider purely spatial nor purely temporal settings, but in accordance with Bittner [2002] situate physical entities in combined spatio-temporal settings. This means, that every spatial state (i. e., position, cf. Position) of a physical entity is valid w.r.t. a particular temporal entity.

A specific configuration of one or more states of entities is called a *situation* (cf. Situation [McCarthy and Hayes 1969], e. g., the situation in which a vehicle is left of our own vehicle), which can be recursively modeled from entities and their properties. Characterizing situations by states is necessary to model re-occurring states, with each occurrence being a different situation, i. e., each situation is a unique node in the time structure [Dylla and Bhatt 2008]. Situations can be used, for example, to

1. summarize initial conditions and planning goals,
2. characterize initial conditions and safety criteria for simulation and verification,
3. communicate with human operators [Baumgartner et al. 2010a; Baumgartner et al. 2014], and
4. keep track of various observations and actions taken (e. g., the success rates of different actions in similar situations) in order to subsequently generalize recorded knowledge and support decision making.

For 1–3, suppressing explicit mentioning of situations may simplify statements [McCarthy and Hayes 1969] (e. g., start [suppressed: in a situation s] where a vehicle is located on a left-turn lane). To record observations and actions taken in connection with situations, it is necessary to mention situations explicitly [McCarthy and Hayes 1969] as entities that themselves can have properties with state (cf. Situation, which is a subclass of Event and, thus, in turn of ConstantEntity).

Unary and n-ary States. Relations (i. e., n-ary states) are especially relevant in human cognition for the comprehension of situations [Barwise and Perry 1983], for example describing the car on the left of the parking lot. Just like unary states, relations may or may not change over time and thereby undergo different states. Both are represented in our conceptual model as state sub-classes. Properties with n-ary state characterize entities from an extrinsic viewpoint and relate them to each other, whereas properties with unary state characterize entities from an intrinsic viewpoint. In the literature, one can observe two main research directions concerning the formal representation of such relations: those, which emphasize relations from a static viewpoint (e. g., calculi for mereotopology [Cohn et al. 1997] and orientation [Dylla and Wallgrün 2007a]), and those that emphasize the dynamic nature of a system already in the relations (e. g., relations between trajectories of moving objects [van de Weghe et al. 2005; Delafontaine et al. 2011]).

Note that n-ary states serve yet another purpose: In accordance with Galton [1995], a notion of *spatial order* must complement the spatial entities in a theory of space (cf. association spatial order). Such a spatial order comprises topological (e. g., inside) as well as positional relations [Bhatt 2010], such as distance, size, and orientation. Concrete spatial calculi can be fit into the model as sub-classes of n-ary state. Examples for such calculi include the region connection calculus [Randell et al. 1992] (topological comparison of extended regions), the 9-intersection calculus (topological comparison of lines, regions and other compound objects) [Egenhofer 2009], the oriented point relations algebra [Moratz et al. 2005] (orientation comparison of oriented points), or the cardinal directions calculus [Goyal and Egenhofer 2001] (orientation w.r.t. an external reference suitable for regions). Appendix B lists relation calculi in more detail. The choice of spatial entities from the universe of discourse directly influences the applicable n-ary spatial states:

— although points can neither be compared for size nor topologically except for equality, their distance and relative orientation can be compared in a fine-grained manner.

— although orientation between lines and regions requires additional information about which of the points in a region determines its orientation, their size and topological relations can be determined easily.

Usually, we thus approximate a complex physical entity with multiple spatial entities to model a current situation from complementing n-ary states. But this demands that we consider additional epistemic knowledge to construct only consistent situation models, as we will see in Sect. 2.4.

Analogous to spatial entities, a notion of *temporal order* must complement the temporal entities in a theory of time to enable reasoning about temporal sequence and duration (cf. association temporal order). Such a temporal order should comprise topological relations (e. g., during) as well as positional relations[3] (e. g., ten minutes ago) [Bhatt 2010]. Topological relations are necessary to reason about concurrently occurring phenomena, whereas positional relations enable reasoning about temporal distance and duration. Consider the topological and positional temporal relations in the following example: a traffic jam will *coincide* (topological) with rush hour, if it does not dissolve *soon* (positional). Such statements are similar in role to temporal logic [Prior 1957], for example linear temporal logic forms a total order on states (LTL [Pnueli 1977]) or computation tree logic forms a partial order of states, i. e., trees (CTL [Clarke and Emerson 1981]).

Galton [1995] restricts his theory of temporal ordering to the topological successor relationship (i. e., a time interval immediately or with some delay succeeds another time interval). This enables basic statements about the temporal sequence of states, but is insufficient to represent concurrently occurring phenomena, which are typical in dynamic spatial systems (e. g., an accident that happens during a traffic jam) [Sandewall 1994]. The necessary topological relations to model concurrent phenomena can be found, for instance, in the interval algebra of Allen [1983] or the temporal semi-intervals of Freksa [1992]. The interval algebra provides a comprehensive set of 13 topological base relations (e. g., before, overlaps, during, after) between intervals. One can compose base relations by disjunction to express uncertainty about a relation between temporal intervals in the interval algebra. Temporal semi-intervals are similar in nature, but use base relations that express uncertainty (e. g., older, survives, precedes). One composes base relations by conjunction to compare two states unambiguously in time. Without logical boolean connectives, a reasoning technique is unable to capture uncertainty or non-determinism, and at the same time it is unable to specify compound facts.

From a positional viewpoint, temporal distance and temporal duration relations are especially relevant [Baumgartner et al. 2007], since they allow us to refine the topological relations between states, events and actions in terms of their distance and duration. For example, a camera failure that began much *sooner* may still affect a vehicle turning left; the camera failure lasts *longer* than it takes the vehicle to make a left-turn. The conceptual reference model includes associations between temporal and spatial entities to define their temporal and spatial ordering—denoted as recursive associations temporal order and spatial order in Figure 2, respectively. These associations extend related to, which represents an n-ary state.

---

[3] Positional relation, in this case, does not refer to position in the spatial domain, but to the position on a temporal scale.

Now that we discussed how to model state, in the next section we focus on the dynamics of a dynamic spatial system.

2.3. Dynamic View

The dynamic view captures how state changes over time, i. e., how what was true is no longer and how other states become true at a later point in time (see Figure 3). It is similar in role to modal logic [Fitting and Mendelsohn 1999] and temporal logic [Prior 1957] for capturing abstract evolution over time, such as in linear temporal logic (LTL [Pnueli 1977]) or computation tree logic (CTL [Clarke and Emerson 1981]), but also includes notions of concrete evolution over time as in dynamic logic [Pratt 1976; Harel 1979].

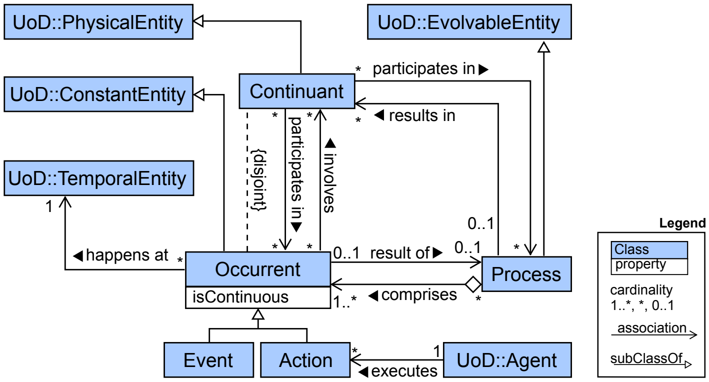


Fig. 3: Continuants, processes, and occurrents in the dynamic view

Definition 2.3 (*Dynamic View*). We distinguish between things that are (*continuants*) and things that happen (*occurrents*, such as events and agent-initiated actions). Continuants, which may or may not evolve, are physical entities. Occurrents are constant entities; they happen at a particular instant or during a particular interval in time, involve continuants and may be the result of a *process*. A process often comprises several occurrents and typically, but not necessarily, results in the creation of new continuants.

For this definition we consolidated different views of the dynamic nature of a dynamic spatial system from geographic information systems [Worboys and Hornsby 2004; Galton and Worboys 2005] and ontology engineering [Grenon and Smith 2004]. These communities are concerned with the *relationship* between multiple states to talk about *dynamic evolution* in terms of events, actions, and processes.

Events, actions, and processes are often distinguished into those modeling continuous gradual changes (e. g., continuous translational motion in both space and time), and those modeling discontinuous ones (e. g., sudden appearance of objects) [Davis 2001; Bhatt 2009]. Continuous changes usually follow the laws of physics, while discontinuous changes result in sudden jumps between states. Although continuous changes can be approximated in purely qualitative models and discontinuous ones in purely quantitative models, both together are considered only in *hybrid* models. In the following paragraphs, we discuss the complementing views of different communities on continuants, occurrents, processes, events and actions that led to our definition.

Continuants, Occurrents, and Processes. Worboys [2005], as many other ontologists (e. g., [Bittner et al. 2004; Gangemi et al. 2002; Smith and Grenon 2004; Sowa 2000]) not only distinguish between constant and evolvable entities as in Sect. 2.1; they also distinguish between *continuants* (cf. Continuant) and *occurrents* (cf. Occurrent). *Continuants*, which are sometimes also referred to as endurants [Bittner et al. 2004], represent those objects of a world that exist and endure over some interval of time (e. g., a left-turn lane, a car). *Occurrents*, sometimes called perdurants [Bittner et al. 2004], occur in a world and "are then gone" [Worboys 2005] (e. g., motion or an overtaking maneuver). From a practical viewpoint it is important to distinguish between continuants and occurrents when we want to enable autonomous agents to recognize other entities in different occurrences [Simons and Melia 2000]. For example, our own vehicle may ask itself whether or not another vehicle is the same as encountered previously in a similar critical situation. If so, our own vehicle may want to revise its own future behavior towards more cautious choices in the presence of this particular other vehicle.

As a discriminating feature, according to Simons and Melia [2000], continuants *always* have non-zero duration and can be identified independently of time and space (e. g., a car), whereas occurrents may either occur at a particular time instant or during a non-empty time interval and can only be identified w.r.t. a spatio-temporal reference (e. g., yesterday's accident on 1st Ave). Usually, as we will discuss for epistemic knowledge, an occurrent causes other occurrents or results in an observable change of state. It is generally agreed [Simons and Melia 2000; Bittner et al. 2004], that continuants may undergo evolution throughout their existence; they are thus a subclass of physical entities. For occurrents, less agreement is established; we discuss the various viewpoints found in literature, which form the basis for our conceptual model.

According to Worboys [2005], occurrents are categorized into *events* (e. g., traffic camera failure, sometimes denoted as *exogenous actions* that are not initiated by an agent [Reiter 2001]), *actions* (e. g., turn left) being agent-initiated occurrents, and *processes* being computational events. Although many other discussions use similar notation about evolution and change, particularly the notion of processes is often seen differently, for instance, in terms of a container for events and actions (e. g., [Galton 2009]).

We follow the more common view of Galton and Worboys [2005] and Galton [2009] and separate processes from occurrents. In their definition, processes are *time-varying, ongoing entities* (e. g., traffic flow may become faster or slower, but still go on during a certain time interval), whereas events and actions are *completed episodes of history* (e. g., the start of a camera failure). Note that we can go back and forth between processes and other occurrents, if necessary. For example a camera failure may have started at a particular point in time (the beginning of the failure is a completed episode). The camera's state is now faulty and a process of camera failure is going on. In that process, additional events may arise: the camera might later be repaired and the failure thus end at another point in time (the end is a completed episode too). Since now the camera failure is gone, we may no longer care for the process and thus convert it into an event that lasted for a particular temporal interval.

The distinction between occurrents and processes means, that processes can undergo evolution so their states may evolve over time, whereas events and actions cannot[4]. In this sense, we interpret events and actions in accordance with the *stages* of Bittner et al. [2004], whereas processes resemble their perdurants.

---

[4] Note that this applies to the ideal correct event or action that actually happened in real-world. In an information system, due to the correction of errors, a record that represents such an event or action can still be altered [Galton and Worboys 2005].

Events and Actions. As we have seen in the examples above, the boundary between occurrents and processes leaves room for interpretation. Depending on the granularity employed during modeling, one can view processes (cf. Process) as being made up of numerous events (cf. Event) and actions (cf. Action); conversely, events and actions can be the results of processes [Galton 2009]. For example, the process of traffic flow comprises many position change events of the contributing vehicles, and each of these position change events is the result of a motion process of a particular vehicle. This view resembles that of a business process definition [Dumas et al. 2005], which is composed of multiple activities, and each activity may in turn itself be detailed by yet another business process definition.

We treat processes as evolvable entities, which enables us to model the history of continuants and the progress of processes in a uniform, snapshot-oriented manner. The states of a process proceed in time as a result of events and actions that initiate and terminate these states. Evolution of a process, thus, is akin to evolution of continuants. Occurrents [Reiter 2001] represent those entities of a world that happen and are then gone. Such occurrents may either be *instantaneous* (i. e., last only for a time instant, e. g., the start of a camera failure), or *extended* (i. e., last for a time interval, e. g., the camera failure itself) [Grenon and Smith 2004], which is represented in the conceptual reference model through an association to a temporal entity. Since we want to consider virtual occurrents too (e. g., computational events), we do not link occurrents to spatial entities, but use an occurrent's property to do so when appropriate.

Events Occur, Actions are Initiated. In accordance with Bhatt and Loke [2008], we further distinguish occurrents into *events*, which necessarily or randomly occur in the environment whenever their preconditions are met (e. g., a vehicle collision, a traffic camera failure), and *actions* that are executed by agents, and hence, their occurrence is additionally dependent on the non-deterministic free will of an agent [Bhatt and Loke 2008] (e. g., a vehicle may or may not turn left, even if it is possible in a certain situation). Note that this definition differs from the terminology of Worboys and Hornsby [2004] and Mau et al. [2007], who define occurrents as real-world happenings and events as representations thereof in a spatio-temporal model. Our distinction between events and actions allows an agent to employ different strategies to avoid adverse occurrents: in order to avoid events (e. g., a vehicle collision), an agent must make careful decisions upfront to avoid the necessary and sufficient conditions of the event to become true. Events that may occur randomly cannot be avoided at all; we can only provide fallback mechanisms (e. g., fallback rules for an intersection in case a traffic light fails, or methods to safely stop an autonomous car when all its sensors fail at once). To avoid specific actions, an agent may have to convince other agents in a negotiation not to carry out the action.

Next, we discuss the epistemic concepts (e. g., necessary and sufficient conditions) that are required for *consistent* and expressive deductive and abductive reasoning about state and evolution phenomena.

2.4. Epistemic View

The epistemic view captures neither what is true at an instant of time (static view) nor what happened between the true states (dynamic view), but instead captures what an agent knows or believes to be true, and what an agent knows that it does not know. The epistemic view is similar in role to epistemic logic [Von Wright 1951; Hintikka 1962].

While the previous sections discussed concepts to describe observed information about our world, *epistemic knowledge* (epistemology) is concerned with describing how an agent can judge the observed information (e. g., find contradictions) and extend it when necessary. For example, having experienced or learned that parts in a container must be smaller than their container explains, why a street cannot be part of the lanes that it comprises. Another example is the frame problem [McCarthy and Hayes 1969; Reiter 2001; Shanahan 1997b]: how can our autonomous car know that moving along the road will only change its own position, and not, for example, cause a thunderstorm or turn the left-turn lane into a traffic light. Epistemic knowledge, in summary, is essential for automated reasoning tasks. Without epistemic knowledge, an agent may follow erroneous information or come to inconsistent conclusions when projecting the behavior of its environment and planning its own steps upon incomplete facts.

We assemble the following definition of epistemic concepts from artificial intelligence [Bennett 2004; Davis 2001; Reiter 2001] and spatio-temporal reasoning [Bhatt 2009; 2010; Dylla and Bhatt 2008; Galton and Worboys 2005].

Definition 2.4 (*Epistemic View*). We categorize epistemic knowledge into *qualification constraints*, *frame constraints*, and *ramification constraints*. Qualification constraints define how a state influences other states and how a state allows or prevents occurrents from happening. Frame constraints define how an occurrent initiates and terminates states and causes other occurrents to happen, and what it does neither initiate nor terminate nor cause. Ramification constraints define how states and occurrents can be composed and what indirect effects follow from them.

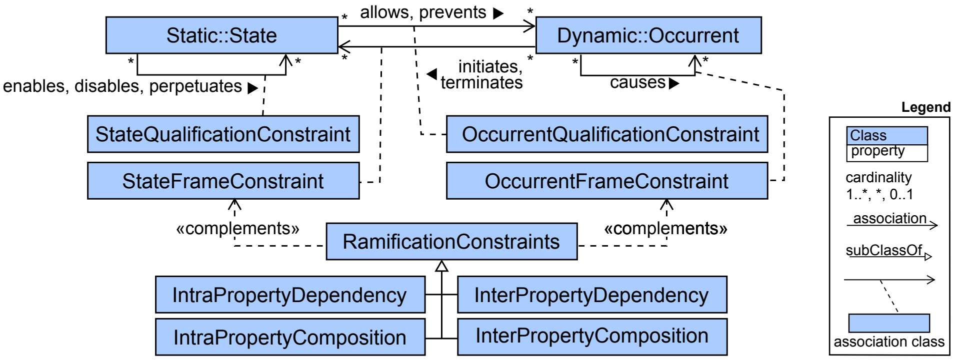


Fig. 4: Knowledge about necessary and sufficient conditions for states and occurrents

Bhatt [2009] summarizes the reasoning tasks that need to be supported for analyzing dynamic spatial systems. These tasks are in accordance with our example and comprise *deductive* and *abductive* reasoning. Deductive reasoning includes (i) planning of actions in order to reach a particular goal state in the dynamic spatial system, (ii) projection of possible future states to decide between action alternatives, and (iii) qualitative simulation of the behavior of a dynamic spatial system. Abductive reasoning explains a current state with actions that may have led to this state. This enables an agent to trace back current facts to outdated ones in the knowledge base and judge whether or not real-world behavior fits to the internal models of an agent [Mitsch and Platzer 2014].

Several problems known in the research area of artificial intelligence must be tackled for deductive and abductive reasoning. In particular, concepts are needed to capture (i) the necessary conditions of states, events and actions (termed *qualification problem* [Reiter 2001]), (ii) the causes, direct effects and non-effects of events and actions (termed *frame problem* [McCarthy and Hayes 1969; Reiter 2001; Shanahan

1997b]), and (iii) the indirect effects of these events and actions (termed *ramification problem* [Reiter 2001]).

Figure 4 models qualification and frame constraints as association classes, since they may not only depend on the states and occurrents they connect, but may also need to satisfy further safety constraints. For example, the state of our own vehicle being located at the intersection *allows* a left turn action, *but only if* other vehicles approaching the intersection yield to our car.

Qualification Constraints. Knowledge about qualification constraints for states and occurrents complements the static and dynamic view. Various dependencies are possible between states (e. g., a state may enable, disable, or perpetuate the existence of another state), which is especially relevant for reasoning about consistency of states [Galton and Worboys 2005]. For example, our own vehicle may measure its position with multiple sensors (e. g., GPS and wheel encoders), and those may report contradictory measurements (e. g., GPS tells us that we are still far from the intersection, while the wheel encoders report that we are already close). The subject vehicle has to detect the inconsistent reports and choose a resolution strategy: it may decide in favor of one of the measurements, retrieve more evidence, or switch into a failure mode to safely stop. Without qualification constraints, an autonomous agent may be unable to detect inconsistencies, or it may devise infeasible plans based on inconsistent states.

Common to both kinds of occurrents, events and actions, is that there are some necessary conditions under which an occurrent may happen, i. e., qualification constraints [Galton and Worboys 2005]. Typically, the necessary conditions are certain states, which *allow* or *prevent* occurrents. For example, the state of the camera being functional allows the event of the camera to fail, or sufficient safety distance to other vehicles allows our own vehicle to initiate a left turn action. Knowing qualification constraints for occurrents allows us to design safe agents (e. g., with control algorithms that actively avoid the qualification constraints of adverse events becoming true) and create plans.

Both, state and occurrent qualification constraints, depend on unary (e. g., a rigid entity cannot contain other entities) [Bhatt 2010] or n-ary states (e. g., relations being *joint exhaustive and partially disjoint*, JEPD, meaning that between any two objects exactly one relation must hold true). When we model the type of an entity as a property of its class, we can even formulate qualification constraints w.r.t. the entities involved (e. g., being a left-turn lane allows the state of a vehicle being located on that lane) as proposed by Apt and Brand [2005].

Frame Constraints. For reasoning about evolution (e. g., situation projection as in [Baumgartner et al. 2009; 2010b]) it is particularly important to know what remains stable and what is caused to change as a result of occurrents [Bhatt 2010]. In artificial intelligence, such knowledge on the direct effects and non-effects of actions and events is known under the term *frame problem* [Reiter 2001; Shanahan 1997b]. Actions and events result in perceivable state change in the environment [Dylla and Bhatt 2008] (i. e., occurrents initiate or terminate states, e. g., the event of a camera failure terminates the state of the camera being functional and initiates the state of the camera being malfunctioning). We interpret occurrents as transitions between states in accordance with Bennett [2004]. Without frame constraints, events and actions would have undefined (i. e., arbitrary) effects, which makes reasoning impractical.

Another interesting distinction of occurrents can be made according to their nature of continuity [Davis 2001]: *continuous* occurrents (e. g., motion) result in gradual state changes, which are discretized with snapshots, whereas *discontinuous* ones (e. g., appearance or disappearance of objects, switching states of a traffic light) are already of discrete nature in the real world. It is important to know the limitations of qualitative reasoning techniques that arise from their coarse approximations of continuous occurrents, in order to refine their results when computing actuator set values at run-time.

The effects of occurrents can be modeled from a static viewpoint or from a dynamic viewpoint. From a static viewpoint, the effects of occurrents can be formulated either in a *unary* manner to directly influence the unary properties of entities (e. g., a shrinking event decreases the size of an entity), or in an *n-ary* manner on n-ary properties [Bhatt and Loke 2008] (e. g., a shrinking event terminates the equal relationship and initiates a smaller relationship). Both, unary and n-ary frame constraints, are subsumed as state frame constraints in Figure 4. Mau et al. [2007] discuss the effects on properties in more detail in the form of *evolving impacts* (sudden or gradual increase or decrease of property values) and distinguish between *delayed* and *immediate* impacts. In hybrid systems [Henzinger et al. 1997; Platzer 2010b], effects are modeled even more detailed using differential equations. The more fine-grained we model the effects of occurrents, the better predictions of future states can be made; often, however, at the expense of computation time (e. g., enumerating paths in a rather small graph of neighboring relations vs. solving a differential equation).

From a dynamic viewpoint, the effects of actions and events are captured in Figure 4 using occurrent frame constraints. Such occurrent frame constraints model a *cause*, that means, a sufficient condition for a particular occurrent to happen. Typically, such a cause is another occurrent: for example, the event of a lightning causes the event of camera failure; the event of distance between two cars becoming too small demands a braking action of the follower car. From an implementation viewpoint, these sufficient conditions of actions define the switching constraints of a controller. This brings us directly to a discussion of the kind of control that we may want to implement: “Time-triggered systems exhibit autonomous control and interact with the environment according to an internal predefined schedule, whereas event-triggered systems are under the control of the environment and must respond to stimuli as they occur.” [Obermaisser 2005, p. 1]. If safety is key, event-triggered control is not a practical approach to implement: when we use sensors to detect events in dense-time systems, we would need to be able to sample infinitely often in order to react to an event in a dependable manner. Thus, time-triggered control is prevalent in safety-critical systems, while event-triggered control is popular for non-safety critical applications [Obermaisser 2005]. This means, however, that for safety-critical systems we need to transform the sufficient conditions into switching conditions that respect the delayed sensing and actuation of time-triggered control.

Ramification Constraints. Certain indirect effects may also occur in addition to the direct effects and non-effects of actions and events considered in frame constraints [Bhatt 2010]. For example, consider a green traffic signal indicating right of way for a particular street. By ramification, this green traffic signal indicates right of way for the lanes being part of this street too. These indirect effects can be attributed to one of the following causes (cf. also the sub-classes of ramification constraints in Figure 4):

(1) *Intra-property dependencies—between states of a single property*; for instance if a vehicle is located on a left-turn lane, then the left-turn lane accommodates the vehicle. From the viewpoint of n-ary states (i. e., relations), intra-property dependencies define: (i) whether or not a relation is symmetric (e. g., equality $=$ is symmetric); (ii) what is the inverse of a relation (e. g., the inverse of $<$ is $\geq$); (iii) what other relations are subsumed by a relation (e. g., $<$ subsumes $\leq$); (iv) what relations are disjoint from a relation (e. g., if $<$ is true, then $>$ cannot be true). A reasoning technique without intra-property dependencies must be fed with additional facts to compensate for the lack of reasoning capabilities.

(2) *Intra-property composition* of the values of the same property; for instance, if our own vehicle is west of the intersection, and the intersection is west of another vehicle, than we can say that our own vehicle must also be west of the other vehicle. An important intra-property composition is transitivity of relations (e. g., equality is transitive, whereas inequality is not). Without composition, a reasoning technique is unable to combine multiple facts.

(3) *Inter-property dependencies—between states of different properties*; for instance, if a vehicle has not yet entered an intersection, and from a distance viewpoint is far away, then as an indirect effect of entering the intersection (which is, topologically speaking, a single step), the distance between the vehicle and the intersection must have been reduced as well. Such inter-property dependencies are necessary whenever: (i) we use logical connectives between two different kinds of relational operators; (ii) these relational operators map the same underlying space into two different spaces; (iii) at least one of the mappings implicitly makes assumptions about the other mapping. For example, let us consider two kinds of relational operators on sets: size comparison ($|s| \ \gamma \ |s'|$, with $\gamma \in \{<, =, >\}$) and topological comparison in $\mathcal{RCC}$ ($s \ \gamma \ s'$ with $\gamma \in \{DC, PO, PP, PPi, EQ\}$[5] and $EQ \equiv (s \backslash s' = \emptyset) \wedge (s' \backslash s = \emptyset)$, ...). Without inter-property dependencies or evaluating the relations in the underlying space, a reasoning technique may, for instance, devise an impossible plan that demands two objects to be topologically equal while one is smaller than the other ($|s| < |s'| \wedge s \ EQ \ s'$).

(4) *Inter-property composition* of the values of different properties; for example, if our own vehicle is far from an intersection, and another vehicle is stopped at the intersection, then we can say that our own vehicle must also be far from the other vehicle; if our own vehicle then moves towards the intersection, then the distance between the vehicles must change as well.

Note that these indirect effects are typically entailed by the basic action theory [Reiter 2001], i. e., they will ultimately be modeled as qualification and frame constraints. However, we want a clean separation, which may enable us to specify transformations that turn ramification constraints into corresponding qualification and frame constraints systematically.

In summary, the previous sections introduced modeling concepts for representing entities, their state, evolution between states, and epistemic knowledge to judge or complement all of the aforementioned.

3. EVALUATION FRAMEWORK

In this section, we transform our conceptual reference model into a criteria catalog in order to create a viable basis for comparing the modeling concepts of different approaches for reasoning in dynamic spatial systems. The corresponding criteria in the catalog introduce measurable indicators of the concepts. We use the following schema for defining our criteria:

(1) a *name* and *abbreviation*, which allow us to cross-reference a criterion throughout the survey, and a *reference to the source* if a criterion has been adopted from others.

(2) a *definition* specifying the criterion, together with a discussion of potential difficulties in defining the criterion due to, for instance, conflicts with other definitions.

(3) an appropriate *measure*, such as a list of *values* or a particular scale, enabling us to compare and rate different approaches with respect to each other.

---

[5] The operators of $\mathcal{RCC}$ [Randell et al. 1992] denote topological relationship between sets: disconnected (DC), partly overlapping (PO), proper part (PP) and its inverse (PPi), and equality (EQ).

Table I: Evaluation criteria for modeling concepts

Abbrv.	Criterion	Measure
	Universe of Discourse	
UD.T	Temporal entities	Instant and/or interval
UD.S	Spatial entities	Point, line, and/or region (polygon, polyhedron, ellipsoid, etc.)
UD.E	Physical entities	Formalized as first-class concept yes/no
UD.P	Properties	Fluent and/or constant
	Static View	
SV.ST	State	Unary and/or n-ary, time-dependent yes/no
SV.TO	Temporal ordering	Topological and/or positional
SV.SO	Spatial ordering	Topological and/or positional; intrinsic/extrinsic/deictic ref. frame
SV.S	Situations	Explicitly formalized or implicitly derived
	Dynamic View	
DV.TS	Temporal structure	Linear, branching, or cyclic time; discrete or dense set
DV.CO	Continuous occurrents	Unary/n-ary, fixed/extensible, informal description/formal def.
DV.DO	Discontinuous occurrents	Appear, disappear, and/or others (listed in survey)
DV.EX	Expressiveness	Temporal and/or others (listed in survey)
	Epistemic View	
EV.QC	Qualification constraints	Listing of necessary conditions
EV.FC	Frame constraints	Listing of sufficient conditions and effects
EV.RC	Ramification constraints	Listing of intra-/inter-property dependencies/composition

The criteria are categorized according to the packages of the conceptual reference model, summarized in Table I, and defined in detail in Sections 3.1 to 3.4.

3.1. Universe of Discourse Criteria

This category contains criteria considering the modeling of real-world physical entities, together with their anchoring in time and space. We focus our discussion on the concepts of instants and intervals (the two major representations in temporal space [Galton 1995]), as well as shape, position, orientation, and size, which are the four major spatial properties of physical entities [Egenhofer 2010].

Temporal Entities (UD.T). We distinguish between approaches that do not refer to a particular kind of temporal entities (NA), those that use temporal instants (Instant) [Galton 1995], those that use temporal intervals (Interval) [Galton 1995], and those that combine both. Different interleaving patterns of concurrently acting agents cannot be captured when time is abstracted solely to instants. In contrast, instantaneous events cannot be precisely represented with only proper intervals (e. g., a ball that bounces back at an instant of time when it hits the ground).

Spatial Entities (UD.S). The spatial entities criterion determines how an approach abstracts from physical entities in terms of their manifestation in space [Galton 1995]. For example, a road can be described as a two-dimensional region by projection onto a two-dimensional plane, which approximates the Earth's surface in a restricted area. Since a spatial calculus, such as $\mathcal{RCC}$, is applicable for some spatial entities but not for others, the actual choice of employed abstraction makes a big difference for reasoning about state consistency and evolution. For example, points can neither be topologically compared, except for equality, nor can they scale; but their distance can be determined unambiguously. We evaluate whether and on which level of abstraction a modeling approach includes spatial entities. According to the *suggested upper merged ontology* (SUMO) [Niles and Pease 2001] objects can be represented in decreasing order of their level of abstraction (i. e., in increasing order of expressiveness) as geometric figures in the form of points, one-dimensional straight lines, two-dimensional planar regions

(e. g., circles, polygons) or three-dimensional regions (e. g., spheres, ellipsoids, polyhedra). We additionally list whether these regions are abstract topological regions or can have concrete geometrical shape. On the one hand, we want to have many different spatial entities for best expressiveness. On the other hand, reasoning complexity typically increases with more complex spatial representations, and additional computation may be necessary to turn low-level measurements into complex spatial entities at run-time.

Physical Entities (UD.E). This criterion evaluates whether or not an approach structures physical entities into (possibly complementing) sub-class hierarchies according to their spatial properties and evolution capabilities, thus formalizing them as first-class concepts. With respect to our reference model, these hierarchies distinguish at least between those objects that are constant and those that may evolve. Thus, they refine in essence the notion of the *theory of objects* of Galton [1995]. In the course of evaluating this criterion, the concrete classification will be listed to gain further insights. Note that non-spatial characterizations (e. g., self-connected, transparent, and autonomous entities distinguished in SUMO [Niles and Pease 2001]) will not be considered during evaluation of this criterion, since they do not influence spatial evolution. If a technique is able to distinguish between physical entities and agents, it may be used to negotiate a joint behavior for achieving goals.

Properties (UD.P). This criterion evaluates the nature of properties used to describe physical entities. We distinguish between *constant* and *fluent* properties. Only approaches with fluent properties are able to capture a history of states. States of fluent properties must be sensed at run-time. In contrast, constant properties never change, which makes them candidates for design-time optimization.

3.2. Static View Criteria

State (SV.ST). In order to represent the history of a physical entity, fluent properties have to capture multiple states, which describe the property change over time [Grenon and Smith 2004]. This criterion measures whether or not states are time-dependent (i. e., anchored in time). A state of a property may either be unary (i. e., describe an entity from an intrinsic viewpoint) or n-ary (i. e., in relationship to other entities). Since n-ary states, as already mentioned, are particularly important for ordering temporal and spatial entities, we introduce dedicated criteria below.

Temporal Ordering (SV.TO). The temporal ordering criterion [Galton 1995] measures the extent to which temporal entities can be compared. Temporal comparison operators are necessary to represent temporal dependencies between states and occurrents (e. g., the primary other vehicle must pass the intersection, *before* the subject vehicle may turn left). The criterion measures the expressiveness of temporal ordering relations (in increasing expressiveness): (i) no ordering supported, (ii) definable (i. e., one may define relational operators), (iii) successor ordering supported, (iv) fully supported. We measure this criterion separately for topological and positional ordering. If ordering is supported, we list the specific calculi that come with a reasoning technique (e. g., interval algebra of Allen [1983]).

Spatial Ordering (SV.SO). The spatial ordering criterion measures how an approach describes the states of spatial n-ary properties (e. g., an approach may use topological and positional ordering). Since we are dealing with partial views on dynamic spatial systems, we are primarily interested in qualitative theories of spatial ordering here [Galton 2000]. Although approaches with only a single aspect of spatial ordering may be easier to use and less computationally expensive, those approaches that use multiple aspects of mixed topological and positional nature can capture possible evolution with less ambiguity, and are thus favored. In order to gain further insights into the potentially achievable expressiveness, we list the specific relation calculi used for spatial ordering. We further distinguish the reference frame [Carlson-Radvansky and Irwin 1993] assumed by these calculi, which anchors relations between physical entities in an *intrinsic* (without external reference frame, e. g., “inside”), *extrinsic* (with universally applicable external reference frame, e. g., “west of”), or a *deictic* manner (with external reference frame from the perspective of the viewer, e. g., “left of”). According to Carlson-Radvansky and Irwin [1993], the reference frame used “is important because the axes of different reference frames are not always oriented in the same way”, which can make reasoning over different reference frames difficult. For example, if we know that accident a is east of intersection x, we still do not know whether it is safe to turn left unless we learn that going left means turning west.

Situations (SV.S). The situations criterion distinguishes between approaches not supporting the concept of situations, those describing them implicitly in terms of the occurrents since some initial state, and those representing situations explicitly as identifiable entities. Only approaches that explicitly represent situations as identifiable entities are able to capture further details about situations with dedicated properties (e. g., duration or severity), and do not require potentially expensive computations to derive situations from their implicit descriptions [Thielscher 2005].

3.3. Dynamic View Criteria

Temporal Structure (DV.TS). The temporal structure criterion measures how time is modeled: linear (e. g., [Pnueli 1977]), branching (e. g., [Clarke and Emerson 1981]), or cyclic structure, defined over either discrete or dense sets of temporal entities [Furia et al. 2010]. *Branching* and *cyclic* [Hornsby et al. 1999] time structures can represent different evolution variants and even cyclic phenomena (e. g., water tide). They are more appropriate for planning, projection, and simulation than *linear* time structures, which always result in a unique future. The same is true for reasoning about explanations for an observed history of states up until a current state: since the same observed sequence of states might be caused by different events and actions, we want to be able to represent the past in a branching structure as well. This allows us to represent alternative courses of occurrents in the same time structure. For example, past sensor information may only fit the epistemic constraints and the current sensor information of our own vehicle, if another vehicle took one of several possible actions at an intermediate state. Conclusions about these possible past actions may enable us to infer likely predictions of future actions.

Continuous Occurrents (DV.CO). This criterion distinguishes between unary and n-ary continuous occurrents. The former define effects for unary properties (e. g., as a fixed set of topological occurrents [Egenhofer 2009]), whereas the latter rely solely on a continuity structure of the employed spatial ordering. We additionally evaluate whether or not the effects of continuous occurrents are *defined formally* with respect to a quantitative reference frame. This gives interesting insights into the applicability of reasoning techniques, since n-ary continuous occurrents will often be defined as transitions in *conceptual neighborhood graphs* (CNG) [Freksa 1991]. A CNG defines a continuity structure [Randell and Witkowski 2004] by imposing constraints on the existence of direct transitions between relations (e. g., two disrelated entities first must overlap, before one may become a part of the other one).

Although transitions in a CNG enable some partial consistency checking between states even in the presence of evolution, they make it harder to plan, project, simulate, or verify the behavior of a dynamic spatial system in detail. In case such a definition w.r.t. a quantitative reference frame is missing, we list whether or not at least an informal discussion of detailed occurrents that go beyond transitions in a CNG is given. Finally, we evaluate whether or not additional continuous occurrents besides motion are definable, which makes it easier to create modeling primitives for domain-specific concepts.

Discontinuous Occurrents (DV.DO). In order to handle sudden changes, such as suddenly appearing or disappearing entities, a modeling approach has to provide concepts for discontinuous occurrents [Bhatt 2009; Davis 2001]. In case discontinuous occurrents are not supported, only entities that were already present in the initial state are available for reasoning. Such an approach, for instance, cannot handle agents that enter the system after it was started, or devise plans that involve creating new entities. This criterion measures whether or not such discontinuous occurrents are supported, and lists the kinds of supported occurrents.

Expressiveness of Occurrents (DV.EX). To select between alternative actions, compare the likelihood of different possible evolution paths, or solve further similar dynamic reasoning tasks, we need to know properties of occurrents, such as their trajectory, duration, probability of occurrence, or costs. Expressiveness of occurrents evaluates whether or not occurrents can be annotated with temporal entities (enables reasoning about their duration) and other properties (e. g., probability, costs, or risk of damage). Approaches without expressiveness information can only select between alternative actions in a random manner and may, thus, devise sub-optimal plans, project unlikely evolution, or simulate uninteresting cases.

3.4. Epistemic View Criteria

The criteria of this category evaluate the extent to which an approach integrates epistemic knowledge about state and evolution phenomena in a reusable manner—in particular qualification, frame and ramification constraints [Bhatt 2009; 2010].

Qualification Constraints (EV.QC). This criterion evaluates, which types of qualification constraints—*state qualification constraints* and *occurrent qualification constraints*—are supported. For both, we list whether or not those necessary conditions can be defined in terms of unary or n-ary states (cf. Figure 4 on page 14: states enable or disable other states, states allow or prevent occurrents). Approaches that only support necessary conditions for n-ary states are unable to consider facts about single entities when reasoning about evolution: e. g., they can express that two entities must be equally sized in order to become topologically equal, but not that one of them has to move when they are at different locations. Approaches that only support necessary conditions for unary states are practically useful for single-entity systems only.

In spatio-temporal reasoning, necessary conditions for states are often modeled as *transitions between relational states* in a CNG [Freksa 1991]. Since these transitions are a form of events, their necessary conditions can be described in terms of the states they connect. Their sufficient conditions are defined by other events, such as motion [Galton and Worboys 2005]. For each kind of qualification constraint, the particular terminology of an evaluated approach is explained as well.

Frame Constraints (EV.FC). This criterion measures whether effects can be provided for unary and n-ary state evolution, as well as for occurrents (with a focus on spatial properties). According to our reference model, occurrents *initiate or terminate* a state and *cause* other occurrents as their effect. Conceptual neighborhood graphs define these effects for n-ary states in spatio-temporal reasoning [Freksa 1991]. Approaches where occurrents initiate or terminate states, but do not cause other occurrents, are unable to integrate multiple relation calculi. Those where occurrents cause only other occurrents have no observable influence on the continuant things in the real world.

Ramification Constraints (EV.RC). This criterion evaluates how an approach handles indirect effects of occurrents. We distinguish between indirect effects resulting from *intra-* and *inter-property dependencies* or from *intra-* and *inter-property composition*. A reasoning technique without intra-property dependencies must compensate for the lack of reasoning capabilities with additional facts. Without inter-property dependencies, logical connectives between two different kinds of relational operators may describe inconsistent facts. Without intra- and inter-property composition, a reasoning technique cannot combine multiple facts (within one or across multiple calculi).

4. COMPARATIVE SUMMARY

In this section, we compare logic-based modeling approaches in terms of a summary of the assessed evaluation criteria. Detailed evaluations with syntax overview, modeling examples, and summaries for each approach can be found in Appendix D. Note that we use the notion *supported* only when a particular aspect can be expressed with dedicated modeling concepts built into an approach. We explicitly state when some aspect can be defined manually from existing operators. If an approach would require substantial extension (e. g., through extending the logic itself), we rate it *not supported*.

Selection of Approaches. Modeling concepts of logic-based commonsense and hybrid system techniques for dynamic spatial systems can be found in various fields:

- In fields concerned with checking *consistency* between state descriptions, for instance with a theory of dynamic spatial systems such as moving objects [Galton 1995; 2000], geographic information systems [Egenhofer and Al Taha 1992; Egenhofer and Mark 1995; Egenhofer and Wilmsen 2006; Egenhofer 2009; 2010], or in fields targeting situation awareness [Kokar et al. 2009; Matheus et al. 2003; Matheus et al. 2005a].
- In fields concerned with *planning* feasible steps in *projected* evolution towards a desired goal situation, for instance using generic methods such as the situation calculus [Bhatt 2012; Reiter 2001], event calculus [Shanahan 1997], fluent calculus [Thielscher 2005], or domain-dependent planning in robot control [Dylla and Moratz 2005; Dylla and Wallgrün 2007b; Miene et al. 2004].
- In fields concerned with analyzing a dynamic spatial system at design time by *simulating* its behavior in a qualitative manner [Apt and Brand 2005; Cui et al. 1992], or by *verifying* the correctness of its behavior using logic-based formal verification of real-time systems [Chaochen and Hansen 2004; Chaochen et al. 1999; Hansen and Hung 2007] or of hybrid systems [Platzer 2008; Platzer and Quesel 2008; Fulton et al. 2015].

Upfront, we would like to particularly stress one practical lesson learned during application of the modeling approaches of this survey. The support for modelers in terms of development environments, modeling patterns, or model evolution is rather limited in comparison to the support one is accustomed to from traditional programming languages. Especially step-wise refinement of models and managing the changes entailed in dependent artifacts is better supported in classical programming languages. This may be in part due to the fact that the domain of dynamical spatial systems has a number of challenging features, but also largely points to an interesting avenue for promising future research in making modeling and analysis of dynamic spatial systems more practical and scalable.

Reading Guide. The comparative summary is structured along the four packages—universe of discourse, static view, dynamic view, epistemic view—of the conceptual reference model. Tables II–V summarize the results and can be interpreted as follows.

What kinds of entities are expressible. See Table II. For spatial entities pay attention to column “kind”: if abstract and qualitative topological entities suffice any approach rated “T” is useful, but if concrete geometrical entities with geometric shapes are needed then look for “G” since approaches supporting “T” are not enough. For physical entities: formally rigorous approaches need checkmarks in column “formal def.”, and if both constant and fluent properties (changeable) are needed with the additional safety of checking that constants are never changed, look for “C,F” in column “properties”.

How to represent knowledge about states and reason about state consistency. See Table III. For temporal order: concurrent phenomena require topological relations, reasoning about duration and temporal distance require positional relations. For spatial order: reasoning about containment requires topological relations, about distance and orientation positional relations; additionally cross-reference with Table II, column “kind” to determine whether purely qualitative relations (kind “topological”) or quantitative (kind “geometrical”) relations are supported.

How to reason about evolution. See Table IV. For time: if a single course of events should be projected, approaches rated “L” for linear time suffice; otherwise “B” for branching time is needed; reasoning in dense time requires domain “$\mathbb{R}$”, in discrete time domain “$\mathbb{N}$” suffices. For occurrents: column “formal def.” should contain checkmarks if occurrents should have a clear and unambiguous meaning, otherwise entries in “informal description” suffice; a checkmark in column “(Dis)appear” is needed if entities can be created/destroyed in a course of events; columns “extensible” and “other” should have checkmarks if custom occurrents should be definable.

How to reason about knowledge. See Table V. Qualification constraints should be supported for situations where an agent wants to determine when an action can be taken. Frame constraints should be supported if effects and non-effects of actions should be definable (some approaches support fine-grained explicit modeling of the dynamics with differential equations and/or clocks, see footnotes). Ramification constraints should be supported if an agent wants to know indirect effects of actions and resolve interdependencies between different reasoning calculi for states and evolution. Further knowledge concepts (e. g., which agent knows about what) are listed in the detailed survey (see Appendix D).

4.1. Universe of Discourse

The summary of the evaluation with respect to the universe of discourse criteria can be found in Table II, and lessons learned are listed in the following paragraphs.

Temporal entities are comprehensively covered by existing approaches. The majority of the approaches discuss a theory of temporal entities with both temporal instants and intervals, cf. [Allen 1983; Apt and Brand 2005; Galton 1995; 2009; 2000; Galton and Worboys 2005; Grenon and Smith 2004; Matheus et al. 2003; Ragni and Wölfl 2006; 2008]. Hence, they support the view of Galton [2009], that either temporal instants or intervals used in isolation are insufficient. Only some approaches, in particular [Bhatt and Loke 2008; Hornsby and Egenhofer 1997] focus on a less complex theory of temporal entities in terms of temporal instants, although in principle their region-based nature allows them to express relations between temporal entities similar to relations between spatial entities. However, such facts are unrelated except for being true in the same situation and it requires additional effort (e. g., using axioms of interaction in [Bhatt and Loke 2008]) to express extended states (e. g., an accident may be positioned at a particular location for several hours) and duration of occurrents (e. g., a left-turn action may not be instantaneous).

Table II: Universe of discourse

		Temporal Entities		Spatial Entities					Physical Entities		Properties
		Instant	Interval	Point	Line	Region	Other	Kind	Description	Formal Def.	Nature
State Consistency	**Theory of Movement** (Galton et al.)	✓	✓	✓	–	✓	–	T	rigid, non-discrete, non-concrete	–	F
	GIS (Egenhofer et al.)	✓	–	–	✓	✓	✓	T	Size	–	C
	SAWA (Kokar et al.)	✓	✓	–	–	–	–	–	–	–	F
Planning and Projection	**Situation and Fluent Calculus** (Reiter, Bhatt et al., Thielscher)	✓	∼	∼	∼	✓	∼	T	dynamic physical properties, fluents	✓	F
	Event Calculus (Shanahan et al.)	✓	✓	✓	✓	✓	–	T	fluents	✓	F
	Robot Navigation (Ragni et al.)	✓	✓	✓	–	✓	–	T	changeable, size-persistent	–	F
Simulation and Verification	**Qualitative Simulation** (Cohn et al.)	✓	✓	–	–	✓	–	T	–	–	C,F
	Differential Dynamic Logic (Platzer)	✓	∼	✓	∼	∼	∼	T,G	∼[i]	✓	C,F
	Duration Calculus (Chaochen et al.)	✓	✓	✓	∼[ii]	∼[ii]	∼[ii]	T,G	–	–	C,F

Legend Supported: yes (✓), definable (∼), no (–)
Nature: constant (C), fluent (F)
Region kind: topological (T), geometrical (G)

[i] Sorts in quantified differential dynamic logic [Platzer 2010b] [ii] in the Shape Calculus extension [Schäfer 2006]

Spatial entities are richer in approaches for concrete domains. Domain approaches, for instance for geographic information systems, provide rich theories of spatial entities in terms of lines [Egenhofer and Mark 1995; Reis et al. 2008], abstract topological (shapeless) regions [Egenhofer and Al Taha 1992; Egenhofer and Mark 1995; Egenhofer and Wilmsen 2006], spheres [Egenhofer 2010], compound objects [Egenhofer 2009], and holes in those different spatial entities, and are even introducing concepts for representing fuzziness (e. g., thick borders and broad-boundary lines [Reis et al. 2008]) in order to enable the representation of various different real-world objects. In contrast, approaches focusing on reasoning tasks seem to prefer either of the following strategies.

- Use abstract, comprehensible notions of space (e. g., points [Ragni and Wölfl 2006] and topological regions [Ragni and Wölfl 2005; 2008] in qualitative planning).
- Burden the definition of spatial entities on the modeler, such as in qualitative simulation [Apt and Brand 2005], hybrid verification [Platzer 2010b], and the duration calculus [Hansen and Hung 2007] (e. g., these approaches can express lines and regions with multiple variables, which must be kept consistent manually). Especially [Platzer 2010b] is very flexible in what spatial entities are definable—even concrete geometrical regions in addition to purely abstract topological regions—at the expense of manual consistency management.

— Do not consider spatial entities as modeling concepts, e. g. SAWA [Kokar et al. 2009].

Physical entities are barely formalized. Qualitative modeling for geographic information systems does not consider a theory of physical entities, since geographic information systems focus on spatial abstractions in the form of their geographic shape. Physical entities are implicitly sketched only as part of an informal discussion [Egenhofer 2010] of the relations that may potentially hold between particular regions. Likewise, a theory of physical entities is not supported in SAWA [Matheus et al. 2003; Matheus et al. 2005a; Kokar et al. 2009] and qualitative simulation [Apt and Brand 2005]. The situation calculus, event calculus, and fluent calculus provide modeling primitives (e. g., in terms of dynamic physical properties of continuants [Bhatt and Loke 2008]), which allow the definition of such a theory of entities. Qualitative planning for robot navigation provides different CNGs according to fixed aspects of an entity [Ragni and Wölfl 2005] and requires physical entities to be size-persistent [Ragni and Wölfl 2008]. Hybrid approaches to dynamic spatial systems, such as $\mathsf{d}\mathcal{L}$ [Platzer 2010b], and real-time approaches, such as the duration calculus [Hansen and Hung 2007], are primarily based on first-order languages. Thus, compound entities arise from multiple variables implicitly (e. g., by means of comments in the specification). Although modeling constructs for compound objects are not absolutely essential to express a dynamic spatial system, they would increase modeling comfort and foster well-established engineering practice in the form of decomposition and reuse. This result points out an opportunity to enhance existing logic-based approaches.

4.2. Static View

The lessons learned in the static view category are summarized in Table III.

Unary and n-ary states are supported and both anchored in time. Most qualitative spatio-temporal reasoning approaches support modeling of both unary and n-ary states. The qualitative planning approach of Ragni and Wölfl [2005] does not provide n-ary states as modeling primitives. The same is true in hybrid verification [Platzer 2010b]: the use of first-order logic requires that n-ary states (i. e., relations between higher objects) are derived manually.

Almost all approaches anchor states in time, and thus support modeling evolution. Only qualitative reasoning approaches for geographical information systems describe states irrespective of time, which reflects the findings of Worboys [2005] that current geographic information systems do not yet model evolution to a satisfactory extent.

Topological temporal ordering is essential for modeling concurrently occurring phenomena. Situation-calculus-based modeling [Bhatt et al. 2005; Bhatt and Loke 2008] and the theory of movement [Galton 1995] provide a notion of positional temporal ordering, which enables the definition of successor relationships between temporal entities (e. g., an accident occurred before a traffic jam emerged). Topological orderings, which are essential for modeling concurrently occurring phenomena in detail (e. g., an accident occurs temporally during a traffic jam), are integrated in terms of interval extensions to the situation calculus (e. g., [Finzi and Pirri 2005]), Allen's $\mathcal{IA}$ in the event calculus [Shanahan 2000], qualitative planning [Ragni and Wölfl 2005; 2006], simulation [Apt and Brand 2005; Cui et al. 1992], and in the duration calculus. In other approaches, such topological temporal relations are definable. For example, $\mathsf{d}\mathcal{L}$ supports temporal instants as modeling primitive. Nevertheless, intervals can be specified manually using two instant variables to represent the lower and upper bound of the interval; topological relations can then be defined using logical connectives and the usual relations on $\mathbb{R}$.

Table III: Summary of static view criteria

		State		Temp. Order		Spatial Order			Situations	
		Arity	Time-dependent	Topological	Positional	Topological	Positional	Reference frame	Implicit	Explicit
State Consistency	**Theory of Movement** (Galton et al.)	1,+	✓	–	$\mathcal{PA}$	$\mathcal{RCC}$	✓[i]	I,E	–	–
	GIS (Egenhofer et al.)	1,+	–	–	–	$9\mathcal{I}$	–	I	–	–
	SAWA (Kokar et al.)	1,+	✓	∼	∼	∼	∼	I,E,D	–	∼
Projection and Planning	**Situation and Fluent Calculus** (Reiter, Bhatt et al., Thielscher)	1,+	✓	∼	$\mathcal{PA}$	$\mathcal{RCC}$	∼	I,E,D	action sequence	initial, ✓
	Event Calculus (Shanahan et al.)	1,+	✓	$\mathcal{IA}$	$\mathcal{PA}$	∼	∼	I,E,D	action sequence	initial
	Qualitative Planning (Ragni et al.)	1	✓	$\mathcal{IA}$	$\mathcal{PA}$	$\mathcal{RCC}$	$\mathcal{CD}$	I,E	–	initial, final
Simulation and Verification	**Qualitative Simulation** (Cohn et al.)	1,+	✓	$\mathcal{IA}$	–	$\mathcal{RCC}$	∼	I	transition sequence	initial
	Differential Dynamic Logic (Platzer)	1	✓	∼	✓	∼	✓	I	–	–
	Duration Calculus (Chaochen et al.)	1,+	✓	$\mathcal{IA}$	✓	✓[ii]	✓[ii]	I,E,D	–	–

Legend
Supported: yes (✓), definable (∼), no (–)
Temporal: point algebra ($\mathcal{PA}$), interval algebra ($\mathcal{IA}$)
Spatial: region connection calculus ($\mathcal{RCC}$)
9-intersection calculus ($9\mathcal{I}$)
cardinal directions calculus ($\mathcal{CD}$)
intrinsic (I), extrinsic (E), or deictic (D)

[i] graph distance [ii] in the Shape Calculus extension [Schäfer 2006]

Spatial ordering is often fixed to a single aspect. Most approaches concentrate on a topological ordering of their spatial entities, although positional ordering can be expressed in almost all approaches. For instance, geographic information systems use the 4-intersection [Egenhofer and Wilmsen 2006] or 9-intersection topological calculus [Egenhofer and Al Taha 1992; Egenhofer and Mark 1995; Reis et al. 2008; Egenhofer 2009; 2010], whereas qualitative planning uses $\mathcal{RCC}$ [Ragni and Wölfl 2005; 2008]. Qualitative planning for robot navigation [Ragni and Wölfl 2006; 2008] additionally considers orientation. Concepts for integrating arbitrary spatial relation calculi are only provided in SAWA [Matheus et al. 2003; Matheus et al. 2005a; Kokar et al. 2009], qualitative simulation [Apt and Brand 2005], and in situation-calculus-based modeling [Bhatt and Loke 2008]. Concerning the nature of these spatial relations, most approaches rely on *conceptual neighborhood* between relations to describe evolution in terms of transitions between the relations. Only a small number of approaches [Bhatt et al. 2005; Bhatt and Loke 2008; Kokar et al. 2009] breaks this convention and expresses dynamics with n-ary states (e. g., approaches, chases). Real-time approaches, such as the duration calculus [Chaochen and Hansen 2004], use variables with constant slope 1 (clocks) to express evolution. Hybrid approaches (e. g., $d\mathcal{L}$ [Platzer 2010b]) even refine this combination of state and dynamics with differential equations. To represent states, real-time and hybrid approaches support points and their positional relationship as modeling primitives. However, these approaches can be extended manually when defining more complex spatial entities and their topological and positional relations (even using any reference frame) from built-in operators.

Situations are mostly histories of events and actions. The surveyed approaches split into two main strands of representing situations: (i) Qualitative simulation [Apt and Brand 2005; Cui et al. 1992], the situation calculus [Reiter 2001], the event calculus [Shanahan 1997], the fluent calculus [Thielscher 2005], qualitative robot planning [Ragni and Wölfl 2006], and differential dynamic logic [Platzer 2010b] identify and describe situations as sequences of events and actions being rooted in a so-called *initial situation* (i. e., situations are histories [Reiter 2001]). In qualitative planning for robot navigation [Ragni and Wölfl 2006], a planning goal in terms of a *final situation* can be additionally specified. (ii) SAWA [Matheus et al. 2003; Matheus et al. 2005a; Kokar et al. 2009] and the fluent calculus [Thielscher 2005] use *named* situations, which can form the basis for relating situations to each other. Still, situations in SAWA focus on static real-world descriptions, and are not integrated with evolution concepts, such as events, actions, and processes.

Property states are typically anchored in time. As can be seen in Table III, almost all of the approaches distinguish between properties and their states. This is in accordance with the SNAP/SPAN ontology [Grenon and Smith 2004], and enables representation of the history of entities in terms of their states.

4.3. Dynamic View

This section summarizes the dynamic view evaluation (cf. overview in Table IV).

Temporal structure depends on task focus. Most approaches work over a linear time structure when it suits the reasoning task: the theory of movement of Galton [1995] focuses on the representation of motion of a single object; qualitative reasoning for geographic information systems [Hornsby and Egenhofer 1997] focus on the history of geographic entities; the event calculus [Shanahan 1997] and qualitative planning for robot navigation [Ragni and Wölfl 2005] focus on providing the shortest path between an initial and a goal situation; the duration calculus [Hansen and Hung 2007] for real-time systems and differential dynamic logic [Platzer 2010b] for hybrid systems focus on verification. Differential dynamic logic additionally supports branching time structure by nested modal operators. Qualitative simulation [Cui et al. 1992] the situation calculus extension of Bhatt and Loke [2008], and the fluent calculus [Thielscher 1999] use a branching time structure in order to enable the representation of different potential evolutions.

Abstractness of continuous occurrents prevents modeling of evolution constraints. Most approaches [Apt and Brand 2005; Bhatt and Loke 2008; Cui et al. 1992; Egenhofer 2009; Reis et al. 2008] abstract from the observed real-world in terms of spatial relations for imposing a spatial ordering on their entities, and therefore rely on the abstract notion of conceptual neighborhood introduced by Freksa [1991] as the prime means for modeling evolution. However, conceptual neighborhood bases on a generic *continuous deformation* occurrent, which abstracts from relevant occurrents [Egenhofer 2009], such as translational motion, scaling, and rotation (also known as affine linear transformations [Davis 2001]). As a consequence, constraints in the form of necessary and sufficient conditions cannot be imposed on transitions between relations. Translational motion, scaling, and rotation are discussed only in geographic information systems [Egenhofer and Al Taha 1992; Egenhofer and Mark 1995; Egenhofer and Wilmsen 2006; Egenhofer 2010; Hornsby and Egenhofer 1997], in a theory of movement [Galton 1995], and in situation-calculus-based modeling [Bhatt et al. 2005]. Qualitative planning for robot navigation [Ragni and Wölfl 2006; 2008] consid-

Table IV: Summary of dynamic view criteria

		Time		**Occurrents Continuous and Discontinuous**						**Expres-siveness**	
		Structure	Domain	Cardinality	Informal description	Formal def.	Extensible	(Dis)appear	Other	Temporal	Other
State Consistency	**Theory of Movement** (Galton et al.)	L	ℝ	+	motion, enter, connect	–	–	–	–	∼	–
	GIS (Egenhofer et al.)	L	–	+	move, scale, rotate, shape	–	✓	✓	split, merge	–	–
	SAWA (Kokar et al.)	–	ℕ	(+)[1]	–	–	✓	–	–	–	–
Projection and Planning	**Situation and Fluent Calculus** (Reiter, Bhatt et al., Thielscher)	B	ℕ	+	approach, recede, split, coalesce	–	✓	✓	sensing, knowledge update	–	–
	Event Calculus (Shanahan et al.)	L	–	+	∼	–	✓	–	∼	✓	–
	Qualitative Planning (Ragni et al.)	L	ℕ	(+)[1]	motion	–	✓	–	–	–	–
Simulation and Verification	**Qualitative Simulation** (Cohn et al.)	B	ℕ	+	–	–	✓	✓	–	–	–
	Differential Dynamic Logic (Platzer)	L,B[i]	ℝ	1	–	✓[ii]	✓	✓[iii]	✓	✓[iv]	–
	Duration Calculus (Chaochen et al.)	L	ℝ	1	–	✓[v]	✓	–	–	✓	–

Legend
Supported: yes (✓), definable (∼), no (–)
Time: linear (L), branching (B), discrete (ℕ), dense (ℝ)
Occurrent cardinality: unary (1), n-ary (+)

[i] by nested [] and ⟨⟩ [ii] linear and non-linear differential algebraic equations [iii] quantified differential dynamic logic [Platzer 2012a] [iv] differential temporal dynamic logic [Platzer 2010b] [v] variables evolve as clocks; an extension with linear differential equations was proposed [Chaochen et al. 1993]

ers solely translational motion. Although these approaches discuss specific continuous occurrents, they do not provide exact formal definitions of the underlying continuous occurrents that may cause discrete transitions between relational states. This abstraction from continuous occurrents, on the one hand, allows efficient reasoning without detailed information (e. g., path planning without knowledge about obstacles being movable or not). On the other hand, it prohibits automated customization of reasoning constraints when detailed sensor information is available. Arbitrary occurrents are definable in SAWA [Matheus et al. 2003] and in situation-calculus-based modeling [Bhatt and Loke 2008]. Those two approaches, however, lack integrated topological occurrents; therefore, users have to rebuild all knowledge about such occurrents with the operators of the modeling framework. Differential equations in hybrid formalisms (e. g., [Platzer 2008]) allow more detailed descriptions. Such level of detail is necessary whenever we want to compute, describe, or analyze precise motion dynamics. Coarse qualitative statements about motion are still possible with hybrid approaches, but reasoning may be more efficient with purely qualitative approaches.

Discontinuous occurrents are necessary to represent evolution. In contrast to continuous occurrents, discontinuous occurrents—for instance, introducing new or removing existing continuants—have not been widely in focus yet. Regardless of domain and task, most approaches assume that all objects relevant to the reasoning task are al-

Table V: Summary of epistemic view criteria

		Constraints							
		Qualification		**Frame**		**Ramification**			
		State	Occurrent	State	Occurrent	Intra-property dependency	Intra-property composition	Inter-property dependency	Inter-property composition
State Consist.	**Theory of Movement** (Galton et al.)	$1,+^{i}$	$+^{i}$	✓^{ii}	✓^{ii}	$\mathcal{RCC}$	$\mathcal{RCC}$	$\sim^{iii}$	–
	GIS (Egenhofer et al.)	$+^{iv}$	–	–	–	$9\mathcal{I}$	$9\mathcal{I}$	$\sim^{v}$	–
	SAWA (Kokar et al.)	–	–	–	–	–	–	–	–
Planning & Projection	**Situation and Fluent Calculus** (Reiter, Bhatt et al., Thielscher)	$1,+^{i,vi}$	$+^{i,vii}$	✓^{viii,ix}	✓^{x}	$\mathcal{RCC}$	$\sim^{xi}$	$\sim^{xii}$	$\sim^{x}$
	Event Calculus (Shanahan et al.)	1,+	1,+	✓^{xiii}	–	$\sim$	$\sim$	$\sim$	$\sim$
	Qualitative Planning (Ragni et al.)	$1,+^{xiv}$	$+^{xiv}$	✓^{xv}	✓^{xvi}	$\mathcal{RCC}$,$\mathcal{CD}$	$\mathcal{RCC}$,$\mathcal{CD}$	–	✓
Simulation & Verification	**Qualitative Simulation** (Cohn et al.)	$1,+^{xvii}$	$1,+^{xviii}$	–	–	✓^{xvii}	✓^{xviii}	–	–
	Differential Dynamic Logic (Platzer)	1^{xix}	1^{xx}	✓^{xxi}	✓^{xxi}	$\sim$	$\sim$	$\sim$	$\sim$
	Duration Calculus (Chaochen et al.)	1,+	1,+	–	✓^{xxi}	$\sim$	$\sim$	$\sim$	$\sim$

Legend Supported: yes (✓), definable ($\sim$), no (–)
Cardinality: unary (1), n-ary (+)
Spatial: region connection calculus ($\mathcal{RCC}$)
9-intersection calculus ($9\mathcal{I}$)
cardinal directions calculus ($\mathcal{CD}$)

[i] $\mathcal{RCC}$ [ii] event [iii] state enables event [iv] $9\mathcal{I}$ [v] feasible relations [vi] existence of state axioms [vii] action precondition axioms [viii] direct effect axioms [ix] knowledge update and action effects [x] occurrence axioms [xi] state constraints [xii] axioms of interaction [xiii] action effects [xiv] $\mathcal{RCC}$ and $\mathcal{CD}$ [xv] motion [xvi] operators (e. g., move) [xvii] intra-state constraints [xviii] inter-state constraints [xix] evolution domain [xx] test/check [xxi] differential equations

ready present in the initial description of the real world. Appearance and disappearance of objects including their specializations (e. g., merge) are explicitly considered as occurrents only in dynamic spatial systems [Bhatt and Loke 2008], geographic information systems [Hornsby and Egenhofer 1997], qualitative simulation [Cui et al. 1992], and in hybrid verification [Platzer 2012a; Loos et al. 2011].

Expressiveness of occurrents is not yet widely considered. As can be seen in Table IV, expressiveness of processes and occurrents in terms of their duration and probability has not yet been the focus. Especially w.r.t. deductive reasoning tasks, such as projection, planning, and verification, representing the temporal dimension of occurrents, however, is of major importance. Only then is it possible for autonomous agents to devise detailed plans and make informed decisions (e. g., is it safe for our own vehicle to cross the intersection before another vehicle). Real-time approaches (e. g., the duration calculus [Chaochen et al. 1999]) make a first step in this direction by modeling evolution with clocks. Hybrid approaches, such as d$\mathcal{L}$ [Platzer 2010b] support detailed dynamics with differential equations and, thus, deliver the necessary level of detail.

4.4. Epistemic View

In Table V, the results for the epistemic view criteria are summarized.

State and occurrent qualification constraints are prime for qualitative reasoning. Only SAWA [Kokar et al. 2009; Matheus et al. 2003; Matheus et al. 2005a] does not consider state qualification constraints, since it comes without a formal theory of spatial ordering. All other evaluated approaches are based upon qualitative or quantitative spatial calculi as a theory of spatial ordering, and hence, utilize constraints formulated on the basis of *relational states*. In geographic information systems, unary states (in particular the size of objects [Egenhofer 2010]) are informally discussed as qualification constraints for the validity of an n-ary state. Bhatt and Loke [2008] support state qualification constraints (called *dynamic physical constraints*) on the basis of unary states. These exploit the dynamic physical properties of the approach for building a theory of entities. However, Bhatt and Loke [2008] do not define a taxonomy of properties and constraints. As a consequence, properties and constraints must be provided by users, instead of being integrated as modeling concepts in a reusable fashion. Entities, besides unary and n-ary states, are considered for formulating constraints in two approaches: (i) in qualitative simulation in the form of so-called *intra-state constraints* [Apt and Brand 2005], and (ii) in qualitative planning for robot navigation in terms of so-called *constraint networks* [Ragni and Wölfl 2006].

Conceptual neighborhood between relations, as utilized in most approaches, constrains the occurrence of transitions (i. e., occurrent qualification constraints) between neighboring relations with coarse-grained necessary conditions only. This means, that most approaches formulate occurrent qualification constraints in terms of relational states and do not consider additional semantics [Ragni and Wölfl 2006]. Sometimes, occurrent qualification constraints actually reflect state qualification constraints. For example, Egenhofer [2010] formulate ‘smaller’ as an occurrent qualification constraint for a transition, while in fact it is a state qualification constraint for the relational state the transition leads to. More fine-grained modeling of occurrent qualification constraints is supported in some situation-calculus based approaches: for instance, Bhatt et al. [2005] formulate occurrent qualification constraints on the basis of unary states. This circumstance indicates that state and occurrent qualification constraints are a prime means for qualitative reasoning.

Sufficient conditions and frame constraints are mostly coarse-grained. As mentioned above, conceptual neighborhood between relations constrains the occurrence of transitions between neighboring relations but does not fully specify the behavior of agents that causes those transitions. Hence, fine-grained sufficient conditions and frame constraints (i. e., constraints basing on other information than the relational states modeled in spatio-temporal calculi) are mostly unconsidered. This means, that approaches basing on conceptual neighborhood graphs [Freksa 1991] merely use informal text descriptions to give an intuition about the nature of transitions (e. g., whether a reduction in distance was caused by motion or growing). Only a small number of approaches—in particular Ragni and Wölfl [2006], d$\mathcal{L}$ [Platzer 2010b], and the duration calculus [Hansen and Hung 2007]—provide formal definitions of fine-grained continuous occurrents in terms of effects on a grid encoding and in the form of differential equations, respectively. These approaches, in principle, allow for modeling of both, fine-grained sufficient conditions and fine-grained frame constraints.

Ramification constraints are mostly restricted to intra-property dependencies and composition. Only some approaches, [Apt and Brand 2005; Bhatt and Loke 2008], put the burden of defining intra-property dependencies and composition on users, whereas most other approaches favor pre-defined composition tables for this purpose. SAWA

[Kokar et al. 2009; Matheus et al. 2003; Matheus et al. 2005a] does not support any ramification constraints, since it focuses on static phenomena only. Although many works in the domain of GIS (e. g., [Egenhofer and Wilmsen 2006]) do not directly mention intra-property composition in terms of composition tables, they build upon qualitative spatial calculi that are accompanied by such composition tables. Besides intra-property composition, however, especially inter-property dependency and inter-property composition as further reusable ramification constraints are often omitted. Hence, when needed these constraints must be defined, which entails additional modeling effort to compose spatial calculi (cf., for instance, the qualitative planning approach of Ragni and Wölfl [2008], who combine $\mathcal{RCC}$ and $\mathcal{CD}$, or Moratz and Wallgrün [2012] who combine $\mathcal{OPRA}_m$ and $\mathcal{CD}$ with distance information).

Summary. In order to exhaustively serve the diverse modeling and reasoning demands of autonomous agents in dynamic spatial systems, integration of concepts from multiple approaches is necessary. Additionally, good engineering practices, such as modeling patterns and systematic support for incremental development of models, are essential to manage the complexity of today's models. A vision and research roadmap of such an integrated dynamic spatial systems modeling and reasoning approach with modeling patterns and support for model evolution is discussed next.

5. CONCLUSION AND OUTLOOK

Our evaluation of the modeling concepts of logic-based commonsense and hybrid reasoning approaches for dynamic spatial systems revealed the strengths and application focus of different approaches (consistency checking, planning, projection, simulation, and verification). Qualitative approaches are good for creating high-level models of a dynamic spatial system and for reasoning about consistency of states, for deriving high-level plans, and for projecting and simulating high-level behavior, possibly in the presence of uncertainty. Hybrid and quantitative approaches complement these qualitative high-level approaches with detailed system models to provide detailed predictions. They could also model high-level qualitative behavior, but not necessarily as efficiently as tailored qualitative approaches.

There is a divide between approaches for knowledge representation (rich modeling features, basic reasoning) and approaches for reasoning (emphasize reasoning engine yet support less rich modeling features). Integration of approaches is, thus, a promising direction for future research, because they could be successful at modeling every part of a dynamic spatial system at the most efficient level that can still address it. The conceptual reference model introduced in this paper suggests that such an integration is indeed feasible, because there are more similarities than the diversity of communities and backgrounds as well as superficial dissonances in terminologies would make one suspect there would be.

In summary, the approach of Bhatt and Loke [2008] offers the most comprehensive set of modeling primitives concerning qualitative reasoning. However, this approach may additionally benefit from more sophisticated event and action descriptions, as introduced with temporal duration for occurrents in the event calculus. Also, verification capabilities and even more detailed quantitative modeling would be highly desirable, such as in the real-time duration calculus [Chaochen et al. 1993] and in differential dynamic logic and its variants [Platzer 2010b] for hybrid systems.

In practice, we want to have different formalisms that are each most suitable for a particular application area and a particular stage in the development life-cycle. What we would like to avoid, however, is repeated modeling effort and manual translation between models as a likely source of error and a cause for inefficiencies. It is necessary to identify their corresponding features in a conceptual reference model to allow such a faithful translation. For example, at design time all the processing power of an extensive, stationary computation infrastructure can be used for verification and simulation. This enables us to model a dynamic spatial system in-depth from many different aspects. At run-time, we still want to have a provably safe navigation algorithm that behaves as simulated. It should, however, be more suitable to the execution infrastructure in an autonomous vehicle, mobile robot and the like. We want to have an end-to-end integration from high-level qualitative models that can be used to capture initial requirements, to detailed hybrid models in verification and simulation, to finally provably correct control code running on the actual platform to check consistency of sensor values with the knowledge base and to plan future steps.

In the following paragraphs we sketch a road-map for future research, which outlines the vision of a comprehensive dynamic spatial systems workbench with the goal of using integrated models with different approaches and helping develop autonomous agents in dynamic spatial systems in a safe and traceable manner. As a first step, we introduced the verification-driven engineering workbench Sphinx [Mitsch et al. 2013; 2014a] to promote collaboration between verification team members with diverse expertise. Going beyond that, such a workbench should promote the application of well-established principles of (software) engineering to leverage reuse and cope with system evolution in order to remain feasible in the context of larger development efforts. In particular, a workbench such as Sphinx, should: (i) integrate and transform among models of different reasoning approaches, (ii) provide a library of reusable and recurring modeling examples and patterns that have proven to be useful in practice, and (iii) support model evolution and co-evolution of dependent artifacts to enable incremental development processes. These three research avenues are discussed in more detail below.

5.1. A Roadmap for Integration of and Transformation between Different Approaches

Modeling languages are often tailored to fit the purpose of one particular approach: for example, in hybrid systems [Platzer 2010b] one often assumes that the physics of one object depends on its own control variables only; we may define classes of objects and their properties in another approach to actually model such architectural assumptions explicitly [Ruchkin et al. 2015]. Hence, if we want to be able to efficiently and safely utilize the advantages of different approaches, we must not only bridge the syntax between different approaches, but first and foremost create sound semantics-preserving transformations between models.

A starting point for model integration (e. g., by model transformation) are efforts in unifying the semantics of different reasoning approaches (e. g., [Belleghem et al. 1997; Thielscher 2011]). Whenever multiple qualitative calculi are used together, a common quantitative representation that provides translatable semantics in an intermediate language, such as introduced in [Mitsch et al. 2011; Bhatt et al. 2011; Schultz and Bhatt 2012] is necessary to integrate spatial aspects in a consistent manner. Additionally, quantitative constraints (i. e., constraints over real arithmetic) can be solved using alternative techniques, such as quantifier elimination over reals [Bhatt et al. 2011]; they are the basis for adapting the scale of abstraction [de Weghe et al. 2014], and integrate nicely with hybrid system modeling and verification [Platzer 2010b].

Many logic-based approaches have overly coarse approximations of continuous occurrents in a qualitative manner and they do not consider discontinuous occurrents (e. g., appearance of entities). Such details may be supplied in the form of rich control models and detailed physical evolution using differential equations as in hybrid systems [Platzer 2010b], and discontinuous occurrents as in [Platzer 2012a; Bhatt and Loke 2008].

As a consequence, models at different levels of abstraction are needed that help analyze a dynamic spatial system w.r.t. various aspects. For verification and simulation at design time, we can rely on powerful computers and create detailed models, such as different spatial entities, detailed differential equations as frame constraints, and dense time structure. At run-time, we may prefer to turn those verified models into quicker safety checks [Mitsch and Platzer 2014] and complement them with qualitative high-level planners that are capable of quickly coping with the intricacies of the real world, such as sensor and actuator uncertainty. From another viewpoint, we may want to start off with an approximated and qualitative system model, later refine it into a hybrid model to verify safety and liveness including continuous physics, and finally turn those models into actual control code. For this, we may borrow the structure of OMG's MDA[6] comprising platform-independent models (PIM) and platform-specific models (PSM). In dynamic spatial systems, a PIM focuses on qualitative aspects by abstracting from the underlying sensors, actuators and entailed physics, whereas a PSM adds quantitative details.

5.2. A Roadmap for Design Patterns and Reusable Model Library

As elaborated in the comparative summary in Sect. 4, physical entities are barely formalized as modeling primitives of the surveyed approaches. Thus, most approaches lack an abstraction mechanism, which would enable reuse by inheritance. As a result, the same modeling questions arise over and over again in many different application contexts (e. g., how should we model a movable object, what are its properties and the actions it can take, what are the necessary and sufficient constraints for that actions).

Since abstraction and inheritance was not the focus in most of the surveyed approaches, we sketch a different method: We envision to conduct a comprehensive analysis of models of different dynamic spatial systems, factor their commonalities and variations and create a library of reusable modeling patterns—a *dynamical and hybrid systems pattern language*—very much in the sense of design patterns of object-oriented software [Gamma et al. 1994]. These modeling patterns could then be instantiated to create the corresponding concrete model fragments. As a basis for creating a modeling pattern library, examples included in publications (e. g., [Apt and Brand 2005; Bhatt and Flanagan 2010; Choi and Amir 2009; Dylla and Bhatt 2008]), benchmark problems (e. g., [Baltes 2000; Fehnker and Ivancic 2004; Madhavan et al. 2009]), and tutorials and case studies included in tools (e. g., KeYmaera [Fulton et al. 2015; Platzer and Quesel 2008; Platzer 2012c; Quesel et al. 2015], SpaceEx [Frehse et al. 2011], or HSolver [Ratschan and She 2007]) should be analyzed.

5.3. A Roadmap for Proof-Aware Model Evolution

In order to support developers in maintaining models of autonomous agents and their environment, development support in the form of refactoring operations and co-evolution of dependent artifacts requires further research. A particularly challenging area here is evolution of models and co-evolution of verification results (i. e., proofs) in a step-wise refinement process [Mitsch et al. 2014a].

For example, common practice is to start with a simple model of the system (e. g., apply only maximum braking power), prove its correctness in a typically laborious process, and incrementally extend the model (e. g., choose between maximum and moderate braking power) to better reflect the real-world system (i. e., *refactor* the model while ensuring preservation of safety constraints). This step-wise refinement process is typically significantly more successful than a one shot attempt. In existing technology, however, much care is needed to ensure that the verification overhead incurred by

---

[6] www.omg.org/mda/

multiple successive proofs of related models is minimized. Initial steps in this direction show that specific refactoring operations can help to limit the verification effort after refactoring [Mitsch et al. 2014a].

The overview of the modeling concepts in this survey may help to design further refactoring operations, so that refactoring operations can either be defined in a generic manner that makes them applicable to many different approaches, or in different flavors that target models at different levels of abstraction (e. g., one refactoring operation for qualitative models and a refined version for hybrid models).

REFERENCES

J. S. Albus and A. M. Meystel. 1996. A Reference Model Architecture for Design and Implementation of Intelligent Control in Large and Complex Systems. *J. Intell. Control and Systems* 1, 1 (1996), 15–30.

R. J. Allan. 2010. *Survey of Agent Based Modelling and Simulation Tools*. Technical Report DL-TR-2010-007. Computational Science and Engineering Department, STFC Daresbury Laboratory.

J. F. Allen. 1983. Maintaining Knowledge about Temporal Intervals. *Commun. ACM* 26, 11 (1983), 832–843.

R. Alur. 2011. Formal verification of hybrid systems. In *Proc. of the 11th Int. Conf. on Embedded Software*, S. Chakraborty, A. Jerraya, S. K. Baruah, and S. Fischmeister (Eds.). ACM, 273–278.

R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. 1995. The algorithmic analysis of hybrid systems. *Theor. Comput. Sci.* 138, 1 (1995), 3–34.

K. R. Apt and S. Brand. 2005. Constraint-Based Qualitative Simulation. In *Proc. of the 12th Int. Symp. on Temporal Representation and Reasoning*. IEEE, 26–34.

D. Arnott and G. Pervan. 2005. A critical analysis of decision support systems research. *J. of Information Technology* 20, 2 (2005), 67–87.

A. Artale and E. Franconi. 2001. A survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence* 30, 1-4 (2001), 171–210.

K. J. Aström and R. M. Murray. 2008. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press.

A. Bachrach, R. He, and N. Roy. 2009. Autonomous Flight in Unknown Indoor Environments. *Micro Air Vehicles* 1, 4 (2009), 217–228.

J. C. M. Baeten. 2005. A brief history of process algebra. *Theor. Comp. Sci.* 335, 2-3 (2005), 131–146.

J. Baltes. 2000. A benchmark suite for mobile robots. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vol. 2. IEEE, 1101–1106.

J. Barwise and J. Perry. 1983. *Situations and Attitudes*. MIT Press.

N. Baumgartner, W. Gottesheim, S. Mitsch, W. Retschitzegger, and W. Schwinger. 2009. On Optimization of Predictions in Ontology-Driven Situation Awareness. In *Proc. of the 3rd Int. Conf. on Knowledge Science, Engineering and Management (LNCS)*, D. Karagiannis and Z. Jin (Eds.), Vol. 5914. Springer, 297–309.

N. Baumgartner, W. Gottesheim, S. Mitsch, W. Retschitzegger, and W. Schwinger. 2010a. BeAware!—Situation Awareness, the Ontology-driven Way. *Data and Knowledge Eng.* 69, 11 (2010), 1181–1193.

N. Baumgartner, W. Gottesheim, S. Mitsch, W. Retschitzegger, and W. Schwinger. 2010b. Situation Prediction Nets - Playing the Token Game for Ontology-Driven Situation Awareness. In *Proc. of the 29th Int. Conf. on Conceptual Modeling (LNCS)*, J. Parsons, M. Saeki, P. Shoval, C. C. Woo, and Y. Wand (Eds.), Vol. 6412. Springer, 202–218.

N. Baumgartner, S. Mitsch, A. Müller, A. Salfinger, W. Retschitzegger, and W. Schwinger. 2014. A Tour of BeAware: A Situation Awareness Framework for Control Centers. *Information Fusion* 20 (2014), 155–173.

N. Baumgartner and W. Retschitzegger. 2006. A Survey of Upper Ontologies for Situation Awareness. In *Proc. of the 4th Int. Conf. on Knowledge Sharing and Collaborative Engineering*. ACTA Press, 1–9.

N. Baumgartner, W. Retschitzegger, and W. Schwinger. 2007. Lost in Time, Space, and Meaning—An Ontology-Based Approach to Road Traffic Situation Awareness. In *Proc. of the 3rd Workshop on Context Awareness for Proactive Systems*. University of Surrey.

B. Beckert and A. Platzer. 2006. Dynamic Logic with Non-rigid Functions: A Basis for Object-oriented Program Verification.. In *IJCAR (LNCS)*, U. Furbach and N. Shankar (Eds.), Vol. 4130. Springer, 266–280.

K. Van Belleghem, M. Denecker, and D. De Schreye. 1997. On the Relation Between Situation Calculus and Event Calculus. *J. Log. Program.* 31, 1-3 (1997), 3–37.

B. Bennett. 2004. A Unifying Semantics for Time and Events. *AI* 153, 1-2 (2004), 13–48.

M. Bhatt. 2009. Commonsense Inference in Dynamic Spatial Systems: Phenomenal and Reasoning Requirements. In *Proc. of the 23rd Int. Qualitative Reasoning Workshop*. University of Ljubljana, 1–6.

M. Bhatt. 2010. Commonsense Inference in Dynamic Spatial Systems: Epistemological Requirements. In *Proc. of the 23rd Int. Florida Artificial Intelligence Research Society Conf.* AAAI Press.

M. Bhatt. 2012. *Reasoning about Space, Actions and Change*. IGI Global, Chapter 9, 284–320.

M. Bhatt and G. Flanagan. 2010. Spatio-Temporal Abduction for Scenario and Narrative Completion. In *Proc. of the Int. Workshop on Spatio-Temporal Dynamics*. IOS Press, 31–36.

M. Bhatt, J. H. Lee, and C. Schultz. 2011. CLP(QS): A Declarative Spatial Reasoning Framework. In *COSIT (LNCS)*, M. J. Egenhofer, N. A. Giudice, R. Moratz, and M. F. Worboys (Eds.), Vol. 6899. Springer, 210–230.

M. Bhatt and S. Loke. 2008. Modelling Dynamic Spatial Systems in the Situation Calculus. *Spatial Cognition and Computation* 8, 1 (2008), 86–130.

M. Bhatt, W. Rahayu, and G. Sterling. 2005. Qualitative Simulation: Towards a Situation Calculus based Unifying Semantics for Space, Time and Actions. In *Conf. on Spatial Information Theory*. Springer.

T. Bittner. 2002. Judgements about Spatio-temporal Relations. In *Proc. of the 8th Int. Conf. on Principles and Knowledge Representation and Reasoning*. Morgan Kaufmann, 521–532.

T. Bittner, M. Donnelly, and B. Smith. 2004. Endurants and perdurants in directly depicting ontologies. *AI Commun.* 17, 4 (Dec. 2004), 247–258.

L. A. Carlson-Radvansky and D. E. Irwin. 1993. Frames of reference in vision and language: Where is above? *Cognition* 46, 3 (1993), 223–244.

A. Casagrande and C. Piazza. 2012. Model Checking on Hybrid Automata. In *15th Euromicro Conf. on Digital System Design*. 493–500.

Z. Chaochen, D. Guelev, and Z. Naijun. 1999. A Higher-Order Duration Calculus. In *Millenial Perspectives in Computer Science. Proc. of the 1999 Oxford-Microsoft Symposium in Honour of Professor Sir Anthony Hoare, Palgrave*. Springer, 13–15.

Z. Chaochen and M. R. Hansen. 2004. *Duration Calculus - A Formal Approach to Real-Time Systems*. Springer.

Z. Chaochen, A. P. Ravn, and M. R. Hansen. 1993. An extended duration calculus for hybrid real-time systems. In *Hybrid Systems*, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel (Eds.). Springer, 36–59.

J. Choi and E. Amir. 2009. Combining Motion Planning with an Action Formalism. In *Proc. of the 9th Int. Symp. on Logical Formalizations of Commonsense Reasoning*, G. Lakemeyer, L. Morgenstern, and M.-A. Williams (Eds.). UTSePress (University of Technology, Sydney), 19–26.

H. Choset, K. M. Lnych, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. 2005. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press.

E. M. Clarke and E. A. Emerson. 1981. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *Logics of Programs, Workshop, Yorktown Heights, New York, May 1981 (LNCS)*, D. Kozen (Ed.), Vol. 131. Springer, 52–71.

E. Clementini, P. Di Felice, and P. van Oosterom. 1993. A Small Set of Formal Topological Relationships Suitable for End-User Interaction. In *SSD (LNCS)*, D. J. Abel and B. C. Ooi (Eds.), Vol. 692. Springer, 277–295.

A. G. Cohn. 1997. Qualitative Spatial Representation and Reasoning Techniques. In *Proc. of the 21st Annual German Conf. on Artificial Intelligence: Advances in Artificial Intelligence*. Springer, 1–30.

A. G. Cohn, B. Bennett, J. M. Gooday, and N. Gotts. 1997. Qualitative Spatial Representation and Reasoning with the Region Connection Calculus. *GeoInformatica* 1, 3 (1997), 275–316.

A. G. Cohn and S. M. Hazarika. 2001. Qualitative Spatial Representation and Reasoning: An Overview. *Fundamenta Informaticae* 46, 1-2 (2001), 1–29.

A. G. Cohn and J. Renz. 2008. Qualitative Spatial Representation and Reasoning. In *Handbook of Knowledge Representation*. Elsevier, 551–596.

Z. Cui, A. G. Cohn, and D. A. Randell. 1992. Qualitative Simulation Based on a Logical Formalism of Space and Time. In *Proc. of the 10th National Conf. on Artificial Intelligence*. AAAI Press, 679–684.

DARPA Tactical Technology Office. 2012. *DARPA Robotics Challenge*. Broad Agency Announcement DARPA-BAA-12-39. Defense Advanced Research Projects Agenca (DARPA), Tactical Technology Office.

E. Davis. 2001. Continuous Shape Transformation & Metrics on Regions. *Fund. Inform.* 46, 1 (2001), 31–54.

G. De Giacomo, Y. Lespérance, and F. Patrizi. 2013. Bounded Epistemic Situation Calculus Theories. In *IJCAI*.

H. De Jong. 2004. Qualitative simulation and related approaches for the analysis of dynamic systems. *Knowl. Eng. Rev.* 19, 2 (June 2004), 93–132.

B. De Schutter, W. P. M. H. Heemels, J. Lunze, and C. Prieur. 2009. Survey of modeling, analysis, and control of hybrid systems. In *Handbook of Hybrid Systems Control – Theory, Tools, Applications*, J. Lunze and F. Lamnabhi-Lagarrigue (Eds.). Cambridge University Press, Chapter 2, 31–55.

N. Van de Weghe, B. de Roo, Y. Qiang, M. Versichele, T. Neutens, and P. De Maeyer. 2014. The continuous spatio-temporal model (CSTM) as an exhaustive framework for multi-scale spatio-temporal analysis. *International Journal of Geographical Information Science* 28, 5 (2014), 1047–1060.

M. Delafontaine, A. G. Cohn, and N. Van de Weghe. 2011. Implementing a Qualitative Calculus to Analyse Moving Point Objects. *Expert Systems with Applications* 38, 5 (2011), 5187–5196.

M. Dumas, W. M. P. van der Aalst, and A. H. M. ter Hofstede. 2005. *Process-Aware Information Systems*. John Wiley and Sons.

F. Dylla and M. Bhatt. 2008. Qualitative Spatial Scene Modeling for Ambient Intelligence Environments. In *Proc. of the 1st Int. Conf. on Intelligent Robotics and Applications*. Springer, 716–725.

F. Dylla and R. Moratz. 2005. Exploiting Qualitative Spatial Neighborhoods in the Situation Calculus. In *Proc of the 4th Int. Conf. on Spatial Cognition: Reasoning, Action, Interaction*. Springer, 304–322.

F. Dylla and J. O. Wallgrün. 2007a. On Generalizing Orientation Information in OPRA$_m$. In *Proc. of the 29th Annual German Conf. on Artificial Intelligence*. Springer, 274–288.

F. Dylla and J. O. Wallgrün. 2007b. Qualitative Spatial Reasoning with Conceptual Neighborhoods for Agent Control. *Intelligent Robotics Systems* 48, 1 (2007), 55–78.

M. Egenhofer. 1989. A Formal Definition of Binary Topological Relationships. In *Proc. of the 3rd Int. Conf. on Foundations of Data Organization and Algorithms*, Vol. 367. Springer, 457–472.

M. Egenhofer. 2009. A Reference System for Topological Relations between Compound Spatial Objects. In *Proc. of the 3rd Int. Workshop on Semantic and Conceptual Issues in GIS*. Springer, 307–316.

M. Egenhofer. 2010. The Family of Conceptual Neighborhood Graphs for Region-Region Relations. In *Proc. of the 6th Int. Conf. on Geographic Information Science*. Springer, 42–55.

M. Egenhofer and K. K. Al Taha. 1992. Reasoning about Gradual Changes of Topological Relationships. In *From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning*. Springer, 196–219.

M. Egenhofer and J. R. Herring. 1991. *Categorizing Binary Topological Relationships between Regions, Lines and Points in Geographic Databases*. Technical Report 7. National Center for Geographic Information and Analysis.

M. Egenhofer and D. Mark. 1995. Modeling Conceptual Neighborhoods of Topological Line-Region Relations. *J. of Geographical Information Systems* 9, 5 (1995), 555–565.

M. Egenhofer and D. Wilmsen. 2006. Changes in Topological Relations when Splitting and Merging Regions. In *Proc. of the 12th Int. Symposium on Spatial Data Handling*. Springer, 339–352.

E. A. Emerson. 1990. Handbook of theoretical computer science. Vol. B. MIT Press, Chapter Temporal and modal logic, 995–1072.

M. R. Endsley. 2000. *Theoretical Underpinnings of Situation Awareness: A Critical Review*. Lawrence Erlbaum Associates, 3–33.

A. Fehnker and F. Ivancic. 2004. Benchmarks for Hybrid Systems Verification. In *Proc. of the 7th Int. Workshop on Hybrid Systems: Computation and Control (LNCS)*, R. Alur and G. J. Pappas (Eds.), Vol. 2993. Springer, 326–341.

A. Finzi and F. Pirri. 2005. Representing Flexible Temporal Behaviors in the Situation Calculus. In *IJCAI*, L. Pack Kaelbling and A. Saffiotti (Eds.). Professional Book Center, 436–441.

Melvin Fitting and Richard L. Mendelsohn. 1999. *First-Order Modal Logic*. Kluwer, Norwell, MA, USA.

A. U. Frank. 1996. Qualitative Spatial Reasoning: Cardinal Directions as an Example. *J. of Geographical Information Science* 10, 3 (1996), 269–290.

G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. 2011. SpaceEx: Scalable Verification of Hybrid Systems. In *CAV (LNCS)*, G. Gopalakrishnan and S. Qadeer (Eds.), Vol. 6806. Springer, 379–395.

C. Freksa. 1991. Conceptual Neighborhood and Its Role in Temporal and Spatial Reasoning. In *Proc. of the Imacs Int. Workshop on Decision Support Systems and Qualitative Reasoning*, M. G. Singh and L. Trave-Massuyes (Eds.). Elsevier, 181–187.

C. Freksa. 1992. Temporal Reasoning Based on Semi-Intervals. *AI* 54, 1 (1992), 199–227.

N. Fulton, S. Mitsch, J.-D. Quesel, M. Völp, and A. Platzer. 2015. KeYmaera X: An Axiomatic Tactical Theorem Prover for Hybrid Systems. In *CADE (LNCS)*, A. Felty and A. Middeldorp (Eds.). Springer.

J. Funge. 1999. Representing Knowledge within the Situation Calculus Using Interval-Valued Epistemic Fluents. *Reliable Computing* 5, 1 (1999), 35–61.

C. A. Furia, D. Mandrioli, A. Morzenti, and M. Rossi. 2010. Modeling Time in Computing: A Taxonomy and a Comparative Survey. *Computing Surveys* 42, 2 (2010), 1–59.

A. Gabaldon and G. Lakemeyer. 2007. ESP: A Logic of Only-Knowing, Noisy Sensing and Acting. In *AAAI*. AAAI Press, 974–979.

A. Galton. 1995. Towards a Qualitative Theory of Movement. In *Proc. of the Int. Conf. on Spatial Information Theory: A Theoretical Basis for GIS*. Springer, 377–396.

A. Galton. 2000. Continuous Motion in Discrete Space. In *Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 26–37.

A. Galton. 2009. Spatial and Temporal Knowledge Representation. *Earth Science Inf.* 2, 3 (2009), 169–187.

A. Galton and M. F. Worboys. 2005. Processes and Events in Dynamic Geo-Networks. In *Proc. of the 1st Int. Conf. on Geospatial Semantics*. Springer, 45–99.

E. Gamma, R. Helm, R. Johnson, and J. Vlissides. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. 2002. Sweetening Ontologies with DOLCE. In *Proc. of the 13th Int. Conf. on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*. Springer, 166–181.

R. Goebel, R. Sanfelice, and A. Teel. 2009. Hybrid dynamical systems. *Control Systems* 29, 2 (2009), 28–93.

R. Goyal and M. Egenhofer. 2001. Similarity of Cardinal Directions. In *Advances in Spatial and Temporal Databases*, C. Jensen, M. Schneider, B. Seeger, and V. Tsotras (Eds.). LNCS, Vol. 2121. Springer, 36–55.

P. Grenon and B. Smith. 2004. SNAP and SPAN: Towards Dynamic Spatial Ontology. *Spatial Cognition and Computation* 4, 1 (2004), 69–104.

J. F. Groote and M. A. Reniers. 2001. *Algebraic Process Verification*. Elsevier, Chapter 17, 1151–1208.

M. R. Hansen and D. Hung. 2007. A Theory of Duration Calculus with Application. In *Domain Modeling and the Duration Calculus*, C. W. George, Z. Liu, and J. Woodcock (Eds.). Springer, 119–176.

D. Harel. 1979. *First-Order Dynamic Logic*. LNCS, Vol. 68. Springer.

B. Heath, R. Hill, and F. Ciarallo. 2009. A Survey of Agent-Based Modeling Practices (January 1998 to July 2008). *J. of Artificial Societies and Social Simulation* 12, 4 (2009), 35.

T. A. Henzinger. 1996. The Theory of Hybrid Automata. In *Proc. of the 11th Symposium on Logic in Computer Science*. IEEE, 278–292.

T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. 1997. HyTech: A model checker for hybrid systems. In *Computer Aided Verification*, O. Grumberg (Ed.). LNCS, Vol. 1254. Springer, 460–463.

J. Hintikka. 1962. *Knowledge and Belief - An Introduction to the Logic of the Two Notions*. Cornell University Press.

K. Hornsby and M. Egenhofer. 1997. Qualitative Representation of Change. In *Proc. of the 2nd Int. Conf. on Spatial Information Theory: A Theoretical Basis for GIS*. Springer, 15–33.

K. Hornsby, M. Egenhofer, and P. J. Hayes. 1999. Modeling Cyclic Change. In *Proc. of the 1st Int. Workshop on Evolution and Change in Data Management*. Springer, 98–109.

J. R. James. 1987. A Survey of Knowledge-Based Systems for Computer-Aided Control System Design. In *Proc. of the American Control Conference*. IEEE, 2156–2161.

R. F. Kelly and A. R. Pearce. 2007. Knowledge and observations in the situation calculus. In *AAMAS*, E. H. Durfee, M. Yokoo, M. N. Huhns, and O. Shehory (Eds.). IFAAMAS, 124.

U. Khadim. 2008. *Process Algebras for Hybrid Systems: Comparison and Development*. Ph.D. Dissertation. Eindhoven University of Technology.

M. M. Kokar, C. J. Matheus, and K. Baclawski. 2009. Ontology-based Situation Awareness. *Information Fusion* 10, 1 (2009), 83–98.

S. Konur. 2013. A survey on temporal logics for specifying and verifying real-time systems. *Frontiers of Computer Science* (2013), 1–34.

Y. Kurata and M. J. Egenhofer. 2006. The Head-Body-Tail Intersection for Spatial Relations Between Directed Line Segments. In *GIScience (LNCS)*, M. Raubal, H. J. Miller, A. U. Frank, and M. F. Goodchild (Eds.), Vol. 4197. Springer, 269–286.

Y. Kurata and M. J. Egenhofer. 2007. The 9+-Intersection for Topological Relations between a Directed Line Segment and a Region. In *BMI (CEUR Workshop Proceedings)*, B. Gottfried (Ed.), Vol. 296. CEUR-WS.org, 62–76.

W. Kurschl, S. Mitsch, and J. Schönböck. 2009. Modeling Distributed Signal Processing Applications. In *Proc. of the 6th Int. Workshop on Wearable and Implantable Body Sensor Networks*, B. P. L. Lo and P. Mitcheson (Eds.). IEEE, 103–108.

G. Lakemeyer. 1996. Only Knowing in the Situation Calculus. In *KR*, L. C. Aiello, J. Doyle, and S. C. Shapiro (Eds.). Morgan Kaufmann, 14–25.

G. Lakemeyer and H. J. Levesque. 1998. AOL: A logic of Acting, Sensing, Knowing, and Only Knowing. In *KR*, Anthony G. Cohn, Lenhart K. Schubert, and Stuart C. Shapiro (Eds.). Morgan Kaufmann, 316–329.

G. Lakemeyer and H. J. Levesque. 2004. Situations, Si! Situation Terms, No!. In *KR*, D. Dubois, C. A. Welty, and M.-A. Williams (Eds.). AAAI Press, 516–526.

I. Lee, O. Sokolsky, S. Chen, J. Hatcliff, E. Jee, B. Kim, A. L. King, M. Mullen-Fortino, S. Park, A. Roederer, and K. K. Venkatasubramanian. 2012. Challenges and Research Directions in Medical Cyber-Physical Systems. *Proc. IEEE* 100, 1 (2012), 75–90.

H. J. Levesque and G. Lakemeyer. 2008. Cognitive Robotics. In *Handbook of Knowledge Representation*, F. van Harmelen, V. Lifschitz, and B. Porter (Eds.). Foundations of Artificial Intelligence, Vol. 3. Elsevier, 869 – 886.

S. M. Loos, A. Platzer, and L. Nistor. 2011. Adaptive Cruise Control: Hybrid, Distributed, and Now Formally Verified. In *FM (LNCS)*, M. Butler and W. Schulte (Eds.), Vol. 6664. Springer, 42–56.

C. Lutz, F. Wolter, and M. Zakharyaschev. 2008. Temporal Description Logics: A Survey. In *Proc. of the 15th Int. Symposium on Temporal Representation and Reasoning*. IEEE, 3–14.

R. Madhavan, R. Lakaemper, and T. Kalmar-Nagy. 2009. Benchmarking and standardization of intelligent robotic systems. In *Proc. of the 14th Int. Conf. on Advanced Robotics*. IEEE, 1–7.

C. J. Matheus, M. M. Kokar, and K. Baclawski. 2003. A Core Ontology for Situation Awareness. In *Proc. of the 6th Int. Conf. on Information Fusion*. IEEE, 545–552.

C. J. Matheus, M. M. Kokar, K. Baclawski, and J. A. Letkowski. 2005. An Application of Semantic Web Technologies to Situation Awareness. In *4th Int. Semantic Web Conference*. Springer, 944–958.

I. Mau, K. Hornsby, and I. D. Bishop. 2007. Modeling Geospatial Events and Impacts Through Qualitative Change. In *5th Int. Conf. on Spatial Cognition: Reasoning, Action, Interaction*. Springer, 156–174.

J. McCarthy and P. J. Hayes. 1969. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In *Machine Intelligence 4*. Edinburgh University Press, 463–502.

A. Miene, U. Visser, and O. Herzog. 2004. Recognition and Prediction of Motion Situations Based on a Qualitative Motion Description. In *RoboCup 2003: Robot Soccer World Cup VII*. Springer, 77–88.

J. Misener, M. Barnes, C.-Y. Chan, D. Cody, S. Dickey, R. Goodsell, T. Gordon, Z. W. Kim, T. Kuhn, T. Lian, D. Nelson, C. Nowakowski, K. Nubukawa, A. Sharafsaleh, S. Shladover, J. Spring, J. VanderWerf, W.-B. Zhang, L. Zhang, and K. Zhou. 2010. *Cooperative Intersection Collision Avoidance System (CICAS): Signalized Left Turn Assist and Traffic Signal Adaptation*. Technical Report UCB-ITS-PRR-2010-20. University of California, Berkeley.

S. Mitsch, K. Ghorbal, and A. Platzer. 2013. On Provably Safe Obstacle Avoidance for Autonomous Robotic Ground Vehicles. In *Robotics: Science and Systems*, P. Newman, D. Fox, and D. Hsu (Eds.). TU Berlin.

S. Mitsch, S. M. Loos, and A. Platzer. 2012. Towards formal verification of freeway traffic control. In *ICCPS*, C. Lu (Ed.). IEEE, 171–180.

S. Mitsch, G. O. Passmore, and A. Platzer. 2013. A Vision of Collaborative Verification-Driven Engineering of Hybrid Systems. In *Do-Form: Enabling Domain Experts to use Formalised Reasoning*. AISB, 8–17.

S. Mitsch, G. O. Passmore, and A. Platzer. 2014a. Collaborative Verification-Driven Engineering of Hybrid Systems. *Mathematics in Computer Science* 8, 1 (2014), 71–97.

S. Mitsch and A. Platzer. 2014. ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models. In *Runtime Verification - 5th International Conference, RV 2014, Toronto, ON, Canada, September 22-25, 2014. Proceedings (LNCS)*, B. Bonakdarpour and S. A. Smolka (Eds.), Vol. 8734. Springer, 199–214.

S. Mitsch, J.-D. Quesel, and A. Platzer. 2014b. Refactoring, Refinement, and Reasoning - A Logical Characterization for Hybrid Systems. In *FM 2014: Formal Methods - 19th International Symposium, Singapore, May 12-16, 2014. Proceedings (LNCS)*, C. B. Jones, P. Pihlajasaari, and J. Sun (Eds.), Vol. 8442. Springer, 481–496.

S. Mitsch, W. Retschitzegger, and W. Schwinger. 2011. Towards Modeling Dynamic Behavior with Integrated Qualitative Spatial Relations. In *Advances in Conceptual Modeling. Recent Developments and New Directions*, O. Troyer, C. Bauzer Medeiros, R. Billen, P. Hallot, A. Simitsis, and H. Mingroot (Eds.). LNCS, Vol. 6999. Springer, 271–280.

R. Moratz, F. Dylla, and L. Frommberger. 2005. A Relative Orientation Algebra with Adjustable Granularity. In *Proc. of the Workshop on Agents in Real-Time and Dynamic Environments*, L. P. Kaelbling and A. Saffiotti (Eds.). Professional Book Center.

R. Moratz and J. O. Wallgrün. 2012. Spatial reasoning with augmented points: Extending cardinal directions with local distances. *J. Spatial Information Science* 5, 1 (2012), 1–30.

C. Nikolai and G. Madey. 2009. Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *Artificial Societies and Social Simulation* 12, 2 (2009), 2–39.

I. Niles and A. Pease. 2001. Towards a Standard Upper Ontology. In *Proc. of the 2nd Int. Conf. on Formal Ontology in Information Systems*. ACM, 2–9.

R. Obermaisser. 2005. *Event-Triggered and Time-Triggered Control Paradigms*. Springer.

A. Platzer. 2008. Differential Dynamic Logic for Hybrid Systems. *J. Autom. Reasoning* 41, 2 (2008), 143–189.

A. Platzer. 2010. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer.

A. Platzer. 2012a. A Complete Axiomatization of Quantified Differential Dynamic Logic for Distributed Hybrid Systems. *Logical Methods in Computer Science* 8, 4 (2012), 1–44.

A. Platzer. 2012b. Logics of Dynamical Systems. In *Proc. of the 27th Symposium on Logic in Computer Science*, N. Dershowitz (Ed.). IEEE, 13–24.

A. Platzer and J.-D. Quesel. 2008. KeYmaera: A Hybrid Theorem Prover for Hybrid Systems.. In *Proc. of the 4th Int. Joint Conf. on Automated Reasoning*. Springer, 171–178.

A. Platzer and J.-D. Quesel. 2009. European Train Control System: A Case Study in Formal Verification. In *Proc. of the 11th Int. Conf. on Formal Engineering Methods (ICFEM) (LNCS)*, K. Breitman and A. Cavalcanti (Eds.), Vol. 5885. Springer, Rio de Janeiro, Brasil, 246–265.

A. Pnueli. 1977. The Temporal Logic of Programs. In *Proc. of the 18th Annual Symposium on Foundations of Computer Science*. IEEE, 46–57.

V. R. Pratt. 1976. Semantical Considerations on Floyd-Hoare Logic. In *17th Annual Symposium on Foundations of Computer Science, Houston, Texas, USA, 25-27 October 1976*. IEEE Computer Society, 109–121.

A. Prior. 1957. *Time and Modality*. Oxford University Press.

J.-D. Quesel, S. Mitsch, S. Loos, N. Aréchiga, and A. Platzer. 2015. How to Model and Prove Hybrid Systems with KeYmaera: A Tutorial on Safety. *STTT* (2015). to appear.

M. Ragni and S. Wölfl. 2005. Temporalizing Spatial Calculi: On Generalized Neighborhood Graphs. In *Proc. of the 28th Annual German Conf. on Artificial Intelligence*. Springer, 64–78.

M. Ragni and S. Wölfl. 2006. Temporalizing Cardinal Directions: From Constraint Satisfaction to Planning. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Rep. and Reasoning*. AAAI, 472–480.

M. Ragni and S. Wölfl. 2008. Reasoning about Topological and Positional Information in Dynamic Settings. In *Proc. of the 21st Int. Florida Artificial Intelligence Research Society Conf.* AAAI, 606–611.

D. A. Randell, Z. Cui, and A. G. Cohn. 1992. A Spatial Logic based on Regions and Connection. In *Proc. of the 3rd Int. Conf. on Knowledge Rep. and Reasoning*. Morgan Kaufmann, 165–176.

S. Ratschan and Z. She. 2007. Safety Verification of Hybrid Systems by Constraint Propagation Based Abstraction Refinement. *ACM Transactions in Embedded Computing Systems* 6, 1 (2007), 1–23.

R. Reis, M. Egenhofer, and J. Matos. 2008. Conceptual Neighborhoods of Topological Relations between Lines. In *13th Int. Symp. on Spatial Data Handling*, A. Ruas and C. Gold (Eds.). Springer, 557–574.

R. Reiter. 2001. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press.

S. Rosenthal, J. Biswas, and M. M. Veloso. 2010. An effective personal mobile robot agent through symbiotic human-robot interaction. In *Proc. of the 9th Int. Conf. on Autonomous Agents and Multiagent Systems*, W. van der Hoek, G. A. Kaminka, Y. Lespérance, M. Luck, and S. Sen (Eds.). IFAAMAS, 915–922.

I. Ruchkin, B. R. Schmerl, and D. Garlan. 2015. Architectural Abstractions for Hybrid Programs. In *Proceedings of the 18th International ACM SIGSOFT Symposium on Component-Based Software Engineering, CBSE 2015, Montreal, QC, Canada, May 4-8, 2015*, P. Kruchten, S. Becker, and J.-G. Schneider (Eds.). ACM, 65–74.

S. J. Russell and P. Norvig. 2003. *Artificial Intelligence — A Modern Approach* (2nd ed.). Pearson Education.

E. Sandewall. 1994. *Features and Fluents: The Representation of Knowledge about Dynamical Systems*. Oxford University Press.

A. Schäfer. 2006. *Specification and Verification of Mobile Real-Time Systems*. Ph.D. Dissertation. Universität Oldenburg.

R. B. Scherl and H. J. Levesque. 1993. The Frame Problem and Knowledge-Producing Actions. In *AAAI*, R. Fikes and W. G. Lehnert (Eds.). AAAI Press / The MIT Press, 689–695.

R. B. Scherl, T. C. Son, and C. Baral. 2009. State-Based Regression with Sensing and Knowledge. *Int. J. Software and Informatics* 3, 1 (2009), 3–30.

C. Schultz and M. Bhatt. 2012. Towards a Declarative Spatial Reasoning System. In *ECAI (Frontiers in Artificial Intelligence and Applications)*, L. De Raedt, C. Bessière, D. Dubois, P. Doherty, P. Frasconi, F. Heintz, and P. J. F. Lucas (Eds.), Vol. 242. IOS Press, 925–926.

A. Serenko and B. Detlor. 2003. Agent toolkit satisfaction and use in higher education. *J. of Computing in Higher Education* 15 (2003), 65–88. Issue 1.

M. Shanahan. 1997a. Event Calculus Planning Revisited. In *Proc. of the 4th European Conf. on Planning (LNCS)*, S. Steel and R. Alami (Eds.), Vol. 1348. Springer, 390–402.

M. Shanahan. 1997b. *Solving the frame problem - a mathematical investigation of the common sense law of inertia*. MIT Press.

M. Shanahan. 2000. An abductive event calculus planner. *J. of Logic Programming* 44, 1-3 (2000), 207–240.

P. Simons and J. Melia. 2000. Continuants and Occurrents. *Aristotelian Society Supplementary Volume* 74, 1 (2000), 59–75.

B. Smith and P. Grenon. 2004. The Cornucopia of Formal-Ontological Relations. *Dialectica* 58, 3 (2004), 279–296.

J. F. Sowa. 2000. *Knowledge Representation—Logical, Philosophical, and Comp. Foundations*. Brooks/Cole,.

R. Stewart, N. A. Stanton, D. Harris, C. Baber, P. Salmon, M. Mock, K. Tatlock, L. Wells, and A. Kay. 2008. Distributed situation awareness in an airborne warning and control system: application of novel ergonomics methodology. *Cognition, Technology & Work* 10, 3 (1 July 2008), 221–229.

M. Thielscher. 1999. From Situation Calculus to Fluent Calculus: State Update Axioms as a Solution to the Inferential Frame Problem. *Artificial Intelligence* 111, 1-2 (1999), 277–299.

M. Thielscher. 2005. FLUX: A logic programming method for reasoning agents. *Theory Pract. Log. Program.* 5, 4-5 (July 2005), 533–565.

M. Thielscher. 2011. A unifying action calculus. *Artificial Intelligence* 175, 1 (03 Jan. 2011), 120–141.

C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.-W. Seo, S. Singh, J. Snider, A. Stentz, W. Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson. 2008. Autonomous driving in urban env.: Boss and the Urban Challenge. *J. Field Robot.* 25, 8 (2008), 425–466.

N. van de Weghe, A. G. Cohn, P. De Maeyer, and F. Witlox. 2005. Representing Moving Objects in Computer-based Expert Systems: The Overtake Event Example. *Expert Systems with App.* 29, 4 (2005), 977–983.

G. H. Von Wright. 1951. *An Essay on Modal Logic*. North-Holland.

M. F. Worboys. 2001. *Modelling changes and events in dynamic spatial systems with reference to socio-economic units*. Vol. 8. Taylor and Francis, Chapter 9, 129–138.

M. F. Worboys. 2005. Event-oriented Approaches to Geographic Phenomena. *Geographical Information Science* 19, 1 (2005), 1–28.

M. F. Worboys and K. Hornsby. 2004. From Objects to Events: GEM, the Geospatial Event Model. In *Proc/ of the 3rd Int. Conf. on Geographic Information Science*. Springer, 327–343.

Online Appendix to: Logic-Based Modeling Approaches for Qualitative and Hybrid Reasoning in Dynamic Spatial Systems

STEFAN MITSCH and ANDRÉ PLATZER, Carnegie Mellon University
WERNER RETSCHITZEGGER and WIELAND SCHWINGER, Johannes Kepler University

This appendix complements the main article with: (i) the complete conceptual reference model, (ii) spatio-temporal relation calculi to express relationships between entities in a qualitative manner, (iii) sample scenarios to instantiate the conceptual reference model and evaluate specific modeling approaches, and (iv) detailed evaluations of modeling approaches for dynamic spatial systems.

A. THE CONCEPTUAL REFERENCE MODEL

In Figure 5, we summarize the four packages of the conceptual reference model in a single UML class diagram.

B. AN INTRODUCTION TO SPATIO-TEMPORAL RELATION CALCULI

In this section we give a short overview of *relation calculi* from the field of qualitative spatio-temporal reasoning, since they are of prime importance in many of the approaches in our survey. These relation calculi define mappings between different spaces to reduce reasoning complexity for qualitative reasoning. They are also used in the subsequent sample scenarios in Appendix C.

In qualitative spatio-temporal reasoning, relations (i. e., n-ary states) between temporal and spatial entities are expressed by employing *relation calculi*. Each of them abstracts from numerical details by focusing on a certain spatio-temporal aspect, such as mereotopology [Randell et al. 1992] or orientation [Dylla and Wallgrün 2007a]. Relation calculi either employ a *topological* view on the related entities, or structure the respective entities in a *positional* sense [Ragni and Wölfl 2008]. Table VI summarizes important qualitative relation calculi; an overview of the algebraic properties of spatial calculi can be found in [Dylla et al. 2013], while examples and more detailed descriptions of some of the calculi are given in [Renz and Nebel 2007]. In this article, we will use

- the topological calculus $\mathcal{RCC}$ [Randell et al. 1992] (e. g., proper part of),
- the positional calculus $\mathcal{CD}$ [Goyal and Egenhofer 2001] (e. g., west of) for specifying directions with an extrinsic reference frame, and
- the positional calculus $\mathcal{OPRA}_m$ [Dylla and Wallgrün 2007a] for comparing the orientation of entities from an intrinsic reference frame (e. g., heads towards).

The categorization of relation calculi into topological and positional calculi is important to determine whether a particular relation holds between two entities or not: topological relations are defined on a topological space, meaning that entities must be representable as sets (e. g., regions), whereas positional ones are defined on points. Besides defining a vocabulary for expressing relational states, such relation calculi provide basic epistemic knowledge about possible evolution between relations. Qualification and frame constraints are often formulated in so-called *conceptual neighborhood graphs* (CNG) [Freksa 1991]. A CNG defines a continuity structure [Randell and Witkowski 2004] by imposing constraints on the existence of direct transitions between

---

DOI: http://dx.doi.org/10.1145/2764901

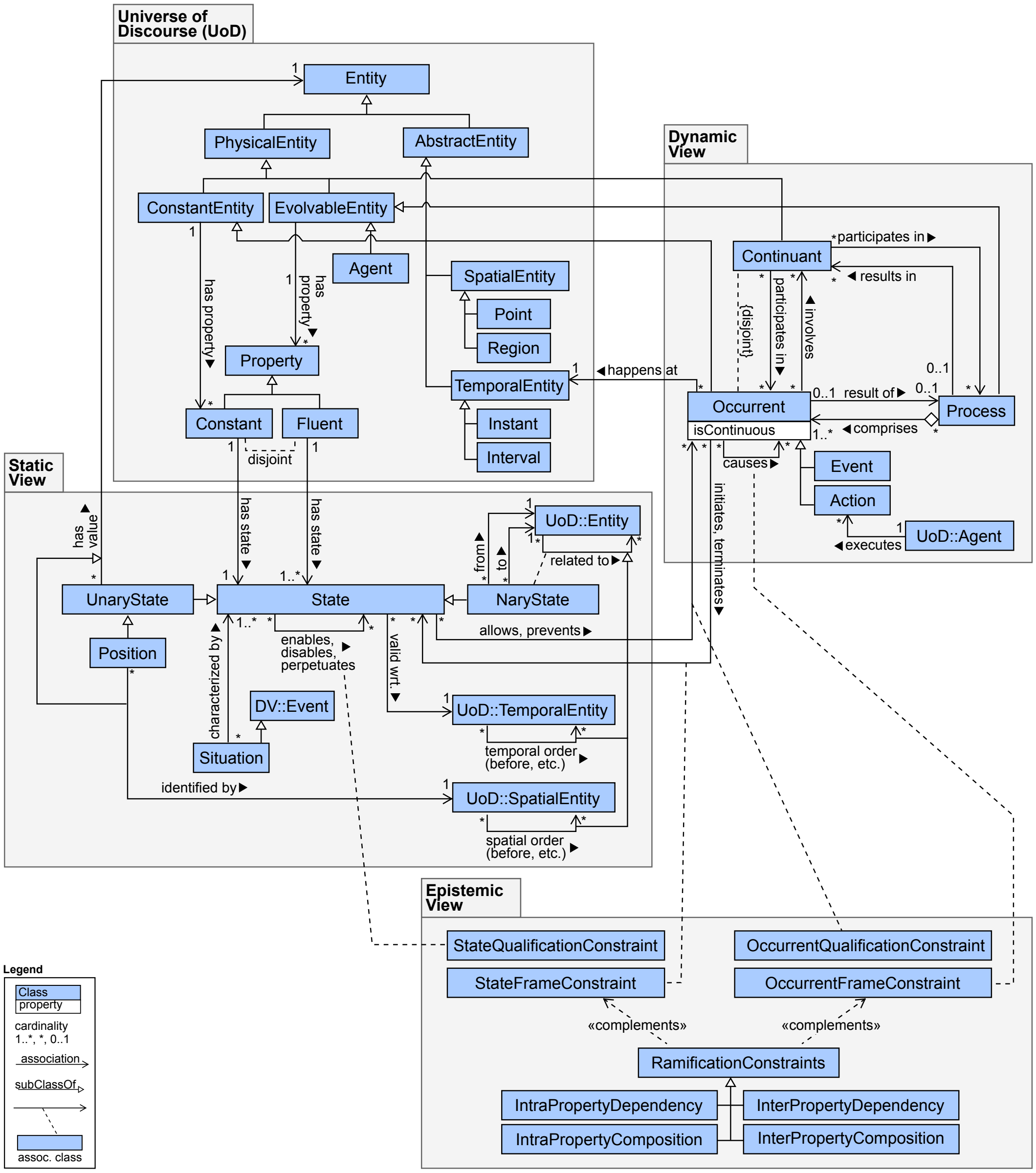


Fig. 5: The conceptual reference model of dynamic spatial systems

relations (e. g., two disrelated entities first must overlap, before one may become a part of the other one). In particular, relations in a CNG are modeled as vertices and evolution in-between as edges connecting these vertices (i. e., vertices resemble qualification constraints, while edges describe frame constraints). Ramification constraints are typically defined by symmetry, inverseness, and transitivity of relations. In the following paragraphs, we introduce the spatial relation calculi $\mathcal{RCC}$ and $\mathcal{OPRA}_m$.

Table VI: Overview of qualitative reasoning calculi

Name	**Description**	**Entities**	**Details**
$\mathcal{IA}$	Topological relations between intervals	I	[Allen 1983]
Semi-intervals	Topological relations between semi-intervals	I	[Freksa 1992a]
$\mathcal{RCC}$	Topological relations between regions	R	[Randell et al. 1992]
4-intersection	Topological relations between regions based on boundaries and interiors	R	[Egenhofer 1989]
9-intersection	Topological relations between regions[i]	R	[Egenhofer and Franzosa 1991]
Ternary projective relations	Ternary projective relations between regions[ii]	R	[Billen and Clementini 2004; Clementini 2013],
Qualitative distances	Distances between points	P	[Hernández et al. 1995; Clementini et al. 1997]
$\mathcal{CD}$	Orientation w.r.t. an extrinsic reference frame[iii]	OP	[Goyal and Egenhofer 2001]
Line segment relations	Ordering (orientation) between line segments	OI	[Schlieder 1995]
Dipole calculus	Orientation between line segments	OI	[Moratz et al. 2000]
$\mathcal{OPRA}_m$	Orientation w.r.t. an intrinsic reference frame	OP	[Dylla and Wallgrün 2007a; Mossakowski and Moratz 2012]
Single/double cross	Orientation w.r.t. an intrinsic reference frame[iv]	OP	[Freksa 1992b; Zimmermann and Freksa 1996; Scivos and Nebel 2001],
Star, StarVars	Orientation w.r.t. an intrinsic reference frame	OP	[Renz and Mitra 2004; Lee et al. 2013]
$\mathcal{EPRA}_m$	Combines orientation ($\mathcal{CD}$, $\mathcal{OPRA}_m$) and distance	OP	[Moratz and Wallgrün 2012]
TPCC	Combines single cross orientation and binary distance (distant, close)	OP	[Moratz and Ragni 2008]
QTC	Qualitative relations between trajectories of moving points[v]	P	[de Weghe et al. 2005a]
Legend	Entities: points (P)–oriented (OP), intervals (I)–oriented (OI), regions (R)		

[i] Variations for $\mathbb{Z}^2$ [Egenhofer and Sharma 1993], for regions with holes [Egenhofer et al. 1994; Vasardani and Egenhofer 2009], for splitting ratios when used with lines [Nedas and Egenhofer 2004], for spheres [Egenhofer 2005], for oriented lines [Kurata and Egenhofer 2006; 2007]. Related formalizations were made by Clementini et al. [1993], for instance, for approximate topological relations [Clementini and Felice 1997]. A comparison between early formalizations can be found in [Clementini and Di Felice 1995]. [ii] Extension to 3D environments [Billen and Clementini 2006] [iii] Generalized to extended objects [Liu et al. 2010] [iv] Generalized to compare shapes [de Weghe et al. 2005b] [v] Generalized to networks [Bogaert et al. 2006; de Weghe et al. 2007; Delafontaine et al. 2008]

Topological Relations in $\mathcal{RCC}$. The relations of $\mathcal{RCC}$ [Randell et al. 1992], which are equivalent to the relations between regions of Egenhofer and Herring [1991], describe topological aspects of relationship between entities. Figure 6 depicts the five relations of $\mathcal{RCC}$ in terms of circular regions: (i) DR describes two disrelated regions sharing neither boundary nor interior; (ii) PO denotes two regions that overlap partially (i. e., at least their boundaries must intersect); (iii) PP denotes a region being part of another one (i. e., the first region's interior must be completely contained in the second region); (iv) PPI is the inverse relation of PP (i. e., a region that contains another one); (v) EQ describes two equal regions (i. e., their boundaries and interiors must be equal). The qualification and frame constraints of $\mathcal{RCC}$ [Randell et al. 1992] are depicted as CNG in Figure 6. In particular, its continuously connected parts can only transition according to Figure 6, not differently. Ramification constraints (in particular, intra-

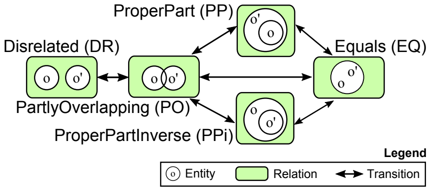


Fig. 6: Conceptual neighborhood between spatial relations in $\mathcal{RCC}$

property dependencies) of $\mathcal{RCC}$ are enumerated in Table VII in terms of inverseness, symmetry, and transitivity of relations.

Table VII: Topological relations between regions (cf. [Randell et al. 1992])

Name	**Symbol**	**Inverse**	**Inverse Symbol**	**Symmetric**	**Transitive**
o disrelated from o'	DR			✓	
o partly overlapping o'	PO			✓	
o proper part o'	PP	o' proper part inverse o	PPI		✓
o equals o'	EQ			✓	✓

Extrinsic Positional Relations in $\mathcal{CD}$. We use the cardinal direction calculus $\mathcal{CD}$ [Goyal and Egenhofer 2001] for describing positional relationships in terms of *extrinsic* orientation [Carroll 1993]. Extrinsic orientation relations compare the orientation of entities towards each other in an external reference frame. The cardinal direction calculus describes relations between a reference o and a target object o' by placing them into a 3×3 grid. The cells in the grid are enumerated with symbols from $\{N, \boldsymbol{NE}, E, \boldsymbol{SE}, S, \boldsymbol{SW}, W, \boldsymbol{NW}, 0\}$, cf. Figure 7. In Table VIII, we summarize inverseness, symmetry, and transitivity of relations in $\mathcal{CD}$ according to Frank [1996]; composition of relations in $\mathcal{CD}$ is also discussed in detail in [Skiadopoulos and Koubarakis 2001].

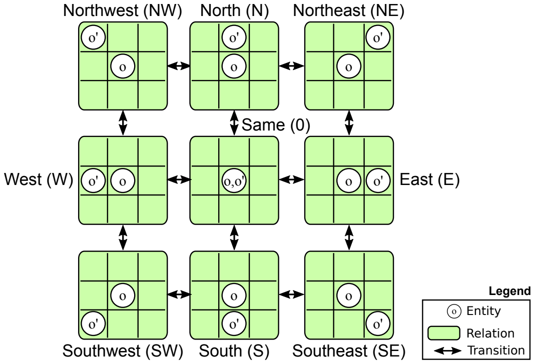
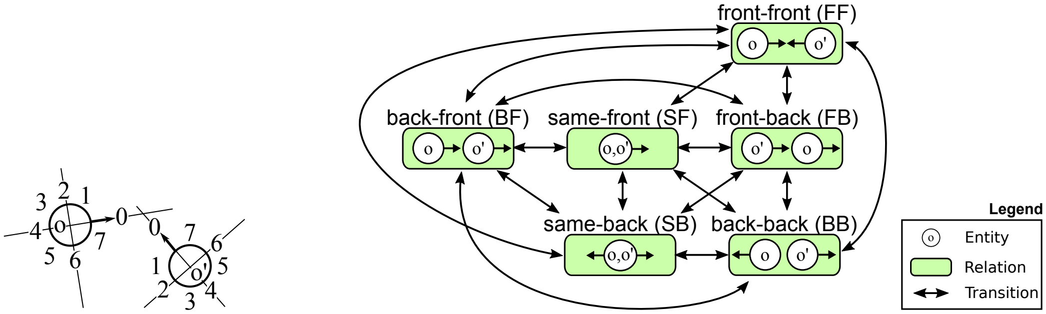


Fig. 7: Conceptual neighborhood in $\mathcal{CD}$ (cf. [Goyal and Egenhofer 2001])

Table VIII: Cardinal direction relations between regions (cf. [Frank 1996])

Name	**Symbol**	**Inverse**	**Inverse Symbol**	**Symmetric**	**Transitive**
o north o'	N	o' south o	S		✓
o north-east o'	NE	o' south-west o	SW		✓
o east o'	E	o' west o	W		✓
o south-east o'	SE	o' north-west o	NW		✓
o same o'	0			✓	✓

Intrinsic Positional Relations in $\mathcal{OPRA}_m$. We use $\mathcal{OPRA}_m$ [Dylla and Wallgrün 2007a] for describing positional relationships in terms of *intrinsic* orientation [Carroll 1993]. Intrinsic orientation relations compare the orientation of entities towards each other without external reference frame. $\mathcal{OPRA}_m$ is defined in a two-dimensional plane and comprises 20 relations [Dylla and Wallgrün 2007b]. These 20 relations are determined by segregating the two-dimensional plane at each oriented point with two orthogonal lines (crossing at the oriented point, and one being in line with the point's orientation) into four planar regions (quadrants) and four linear regions (line segments) [Dylla and Wallgrün 2007a], as depicted in Figure 8a.



(a) Relative orientation between points

(b) Conceptual neighborhood graph

Fig. 8: Overview of $\mathcal{OPRA}_m$ (cf. [Dylla and Wallgrün 2007a])

Beginning at the line segment in direction of point orientation, the linear and planar regions are then numbered from 0 to 7. A relation between two oriented points is named according to the index of the position the other oriented point is placed at. For example, in Figure 8a o and o' are in a relation $o\ {}_2\angle_7^1\ o'$, which means that o' is in sector 7 of o ($\angle_7$), and o is in sector 1 of o' ($\angle^1$) when at each point the space is segregated with two orthogonal lines (${}_2\angle$). If the points coincide, relations are defined as $o\ {}_2\angle i\ o'$. For better readability, the most important six relations are renamed by applying the naming scheme of Moratz et al. [2005] to front-front (FF, $o\ {}_2\angle_0^0\ o'$), front-back (FB, $o\ {}_2\angle_0^4\ o'$), back-front (BF, $o\ {}_2\angle_4^0\ o'$), back-back (BB, $o\ {}_2\angle_4^4\ o'$), same-front (SF, $o\ {}_2\angle 0\ o'$), and same-back (SB, $o\ {}_2\angle 4\ o'$). Figure 8b shows the qualification and frame constraints of $\mathcal{OPRA}_m$ in the CNG between these six relations, following the definition of Dylla and Wallgrün [2007b]. Table IX summarizes the ramification constraints of $\mathcal{OPRA}_m$.

Table IX: Relations between oriented points (cf. [Dylla and Wallgrün 2007a])

Name	Symbol	Inverse	Inverse Symbol	Symmetric	Transitive
o front-front o'	FF			✓	
o front-back o'	FB	o' back-front o	BF		✓
o back-back o'	BB			✓	
o same-front o'	SF			✓	✓
o same-back o'	SB			✓	

C. APPLICATION EXAMPLES

C.1. Running Example: Autonomous Vehicle

To illustrate the modeling features of different reasoning techniques, we introduce a sample scenario that will be used throughout the detailed evaluation in Appendix D. This scenario reflects the complexity of dynamic spatial systems, while at the same time it strives for a minimal setup based on examples from an intuitively accessible domain, i. e., road traffic.

The road traffic scenario comprises (i) *physical entities*, such as traffic lights, and (ii) cars as *agents* that (iii) have only *partial but possibly complementary views* on a situation (sensors have limited range). These agents (iv) may sense or receive *uncertain information*, since sensors are not perfect and sensor data may be outdated. Both, physical entities and autonomous agents, are described with a combination of (v) *constant* non-changeable states (e. g., license plate) and (vi) *fluent* changeable states (e. g., position). These states are of either (vii) *unary* nature (e. g., position in GPS coordinates) or (viii) *n-ary* nature (e. g., position relative to the traffic light). In a dynamic spatial system, (ix) *events* may occur, such as traffic light failure, and (x) agents can take *actions* (e. g., move). Epistemic knowledge may specify further details, such as (ix) necessary and sufficient conditions under which events may occur and actions can be initiated. Finally (xii), agents must satisfy *safety constraints* (e. g., not passing a red light) when taking actions.

Since it is easily accessible, we use a running example inspired by a CICAS scenario. In order to avoid collisions at intersections, CICAS informs all vehicles that approach an intersection about possible hazards (e. g., risk of upcoming red light violation or other vehicles approaching the intersection with high velocity). To this end, CICAS exchanges information not only between an intersection and approaching vehicles, but also in-between vehicles [Misener et al. 2010]. Here, we concentrate on the *situation awareness* capabilities that are necessary to avoid crashes at intersections from a software viewpoint (especially an autonomous vehicle's ability to devise plans that will work in a changing environment).

Let us suppose that an autonomous vehicle (the so-called *subject vehicle* [Misener et al. 2010]) wants to turn left at the intersection from 1st Avenue onto 2nd Street, see Figure 9. Priority at this intersection is established by right of way from the right (i. e., the subject vehicle has to yield to those vehicles that approach the intersection from the east). To turn safely, the vehicle must not only consider *constant* information, such as the presence of turn lanes, but also *fluent* information, such as the positions and behavior of other vehicles at the intersection (the so-called *primary other vehicles*). In our scenario, the subject vehicle sv prepares for the left turn and, hence, changes to the left-turn lane south of the intersection x. A primary other vehicle pov approaches the intersection x on 2nd Street from the east, which is unrecognized by the subject vehicle sv due to restricted lines of sight at the intersection. However, the roadside infrastructure (e. g., a camera $cctv$) recognizes the approaching primary other vehicle pov and estimates that the pov is close to or even very close to the intersection. Conse-

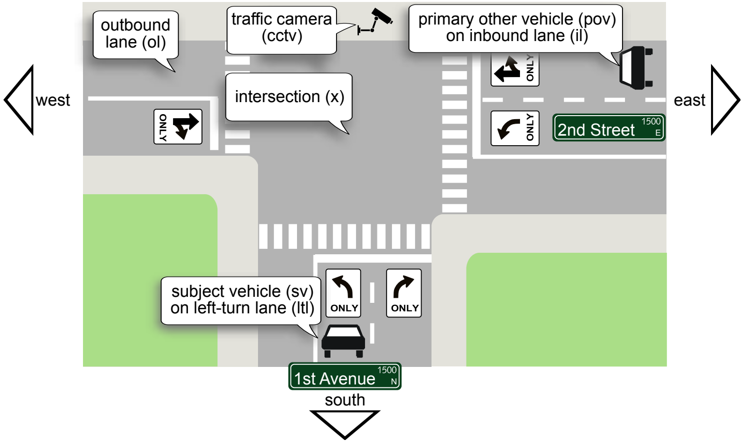


Fig. 9: Intersection scenario: an autonomous vehicle has to plan a left turn

Table X: Overview of scenario elements

	Requirement	**Scenario element**
(i)	Physical entity	**Left-turn, outbound, inbound** lane (ltl, ol, il), **intersection** (x)
(ii)	Agent	**Subject** and **primary other** vehicle (sv, pov)
(iii)	Partial view	sv **unable to sense** pov
(iv)	Uncertain information	pov **close or very close to** x
(v)+(vii)	Constant unary state	x **intersection width**
(v)+(viii)	Constant n-ary state	ltl **south of** x
(vi)+(vii)	Fluent unary state	**Positions** of sv and pov
(vi)+(viii)	Fluent n-ary state	pov **to the right of** sv
(ix)	Event	Camera ($cctv$) **failure**
(x)	Action	**Turn left**
(xi)	Necessary condition	**In order to** turn left: **positioned on** ltl
(xii)	Safety constraint	**Safe left-turn**, sv turns after pov passed x

quently, CICAS instructs the subject vehicle sv to stop, since the position and velocity of the primary other vehicle may not allow making a safe turn. After the primary other vehicle passed, the subject vehicle may enter the intersection and turn west onto the outbound lane ol on 2nd Street, because CICAS issues clearance. Note, that several variations of CICAS exist (e. g., completely without infrastructure) [Misener et al. 2010]. We just focus on one for clarity.

Table X summarizes the elements of the sample scenario with respect to the scenario ingredients discussed above. The subject vehicle in this example, from a *static* viewpoint, has the task to assemble and validate information on the current situation at the intersection. This information stems from its own sensors and from those of the intersection. From a *dynamic* viewpoint, the subject vehicle has to find a *plan* to safely turn left at the intersection considering the current fluent state. It should *project* the behavior of other dynamic entities (the primary other vehicle pov) when devising the plan. Both, static and dynamic viewpoints are described in terms of the constituents of dynamic spatial systems, which establish our *universe of discourse*. The reasoning techniques employed during design-time or run-time may need to consider *epistemic knowledge* (e. g., necessary conditions). Also, to ensure safety of such an autonomous vehicle, we want to be able to *simulate* and *verify* the correctness of the resulting dynamic spatial system and of the interaction among its agents.

Object Model of the Autonomous Vehicle Scenario. Let us now exemplify the conceptual reference model by means of a UML object diagram of this sample scenario. This object diagram models an excerpt of our sample scenario to instantiate the key concepts of the conceptual reference model (cf. Figure 10). This demonstrates that the concepts included in the conceptual reference model are feasible to qualitatively model autonomous agents in dynamic spatial systems.

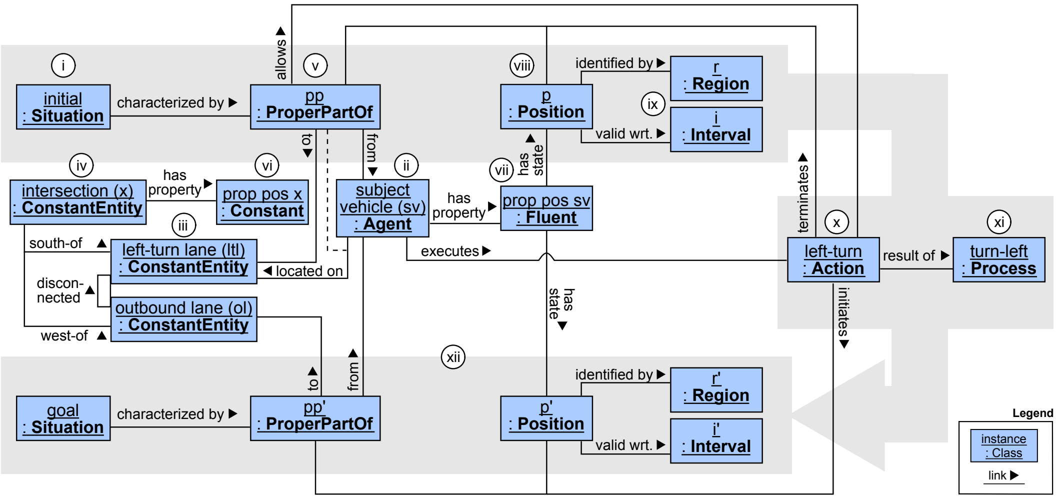


Fig. 10: The sample scenario as an object diagram of the conceptual reference model

In summary, this object diagram models that the subject vehicle is located on the left-turn lane south of the intersection and wants to turn to the outbound lane west of the intersection. The arrow in the background illustrates the transition from the initial situation at the top of the figure to the situation at the bottom through actions. In the initial situation (i) of the sample scenario, the subject vehicle sv (ii) is located on the left-turn lane (iii) south of an intersection x (iv). In other words, from the viewpoint of a topological relational state, this initial situation is characterized by the subject vehicle being a *proper part of* (v) the left-turn lane. The constant unary property $prop\_pos\_x$ (vi) holds the position of the intersection, whereas a fluent unary property $prop\_pos\_sv$ (vii) describes the subject vehicle's position. This means, the position has multiple consecutive states. In the initial situation, the subject vehicle is located at position p (viii) identified by a spatial region r and valid w.r.t. a temporal interval i (ix). Also, being a proper part of the left-turn lane allows the subject vehicle to execute a left-turn action (x), which is the result of a turn-left process that is not further detailed (xi). This left-turn action, which models instant motion for simplicity, has two immediate consequences: First, it terminates the n-ary state pp (of type *ProperPartOf*) and the unary state p (of type $Position$) of the initial situation, i. e., the subject vehicle is no longer located on the left-turn lane. Second, it initiates two corresponding new states (pp' and p'), i. e., after executing the left-turn action the subject vehicle is located on the outbound lane west of the intersection, and thus, reached its goal situation (xii).

C.2. Emergency Response Robot

To illustrate the applicability of the CRM for further domaines, we introduce an example inspired by the current DARPA robotics challenge [DARPA Tactical Technology Office 2012]. In the DARPA robotics challenge, a disaster robot must demonstrate its capabilities to cope with disaster response scenarios, such as an accident in an industrial plant involving a leaking gas pipe. For this, the challenge demands that a robot (i) drives a vehicle to a disaster site, dismounts from the vehicle, and travels across rubble to its operational area, (ii) unblocks and opens a door, (iii) climbs a ladder, and (iv) uses a tool to tear down a concrete obstacle, locates and closes a valve, and finally replaces a cooling pump.

On a rather high level of abstraction, this variation of the DARPA robotics challenge is a variant of the well-known "piano mover's" motion planning problem [Choset et al. 2005]. The piano mover's problem is used in robotics, vehicle automation, and other fields concerned with autonomous agents to exemplify the task of finding a series of executable steps that enable an autonomous agent to transition from a current situation towards a desired goal situation, while respecting certain constraints on the way (e. g., find a series of motion steps to bring a disaster response robot to its operational area while avoiding obstacles). In order to cope with the dynamic nature of disaster sites (e. g., collapsing walls or spreading fire), we have to consider navigation through changing environments or devising a plan for a partially unexplored environment [Choset et al. 2005], instead of assuming perfect knowledge about obstacles in a static environment as in the classical piano mover's problem.

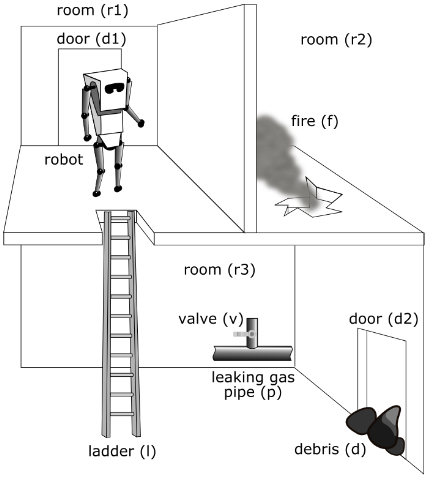


Fig. 11: Disaster response scenario: a disaster response robot must shut off a leaking gas pipe

Let us suppose, as depicted in Figure 11, that a gas pipe in an industrial plant leaks, and as a consequence, an explosion occurs in an area adjacent to the pipe. In order to prevent further damage to the environment and the plant, the gas leak must be stopped immediately. Personnel evacuated from the plant, however, report that the disaster area near the pipe is too hostile for humans; thus, a disaster response robot should shut off the leaking gas pipe. They are also able to partly provide information about possible obstacles caused by the explosion. For example, the door on the lower level to the leaking gas pipe is blocked by debris, and a fire spreads in the area above the gas pipe, and thus, a ladder to access the pipe may become inaccessible in the near future.

To reach its operational area, first, a disaster robot has to assemble a coherent description of the current state of the plant from a *static* viewpoint (the *current situation*), basing on the incomplete information supplied by different plant personnel members. Next, the robot has to decide about the best possible means to reach the leaking pipe as fast as possible, considering possible evolution of the situation in the plant from a *dynamic* viewpoint. From a static viewpoint, the robot may conclude that it would be best to access the leaking gas pipe through the shortest path with the least obstacles known to be present in the current situation (i. e., using the ladder). However, the fire spreading in the plant bears the risk of blocking the ladder before the robot will be able to pass it. As an alternative solution, the robot could try to enter through the door on the lower level, which requires removing debris from the blocked door.

In summary, the robot's task in this example is (i) from a *static* viewpoint to validate and assemble information describing the current situation in the plant, and (ii) from a *dynamic* viewpoint to find a plan to reach the leaking gas pipe in a changing environment, as well as simulate or project the behavior of dynamic entities in the plant (e. g., the fire) when devising the plan. Also, to ensure safety of a disaster robot, we want to verify the correctness of its behavior and interaction with other autonomous agents within the plant.

Object Model of the Emergency Response Robot Scenario. The object diagram in Figure 12 models an excerpt of the spatial aspect of the robot sample scenario similar to the spatial aspect of the traffic scenario above. The arrow in the background illustrates the transition from the initial situation at the top of the figure to the situation at the bottom through actions. We further complement this spatial description with the state description and manipulation of the valve in Figure 13.

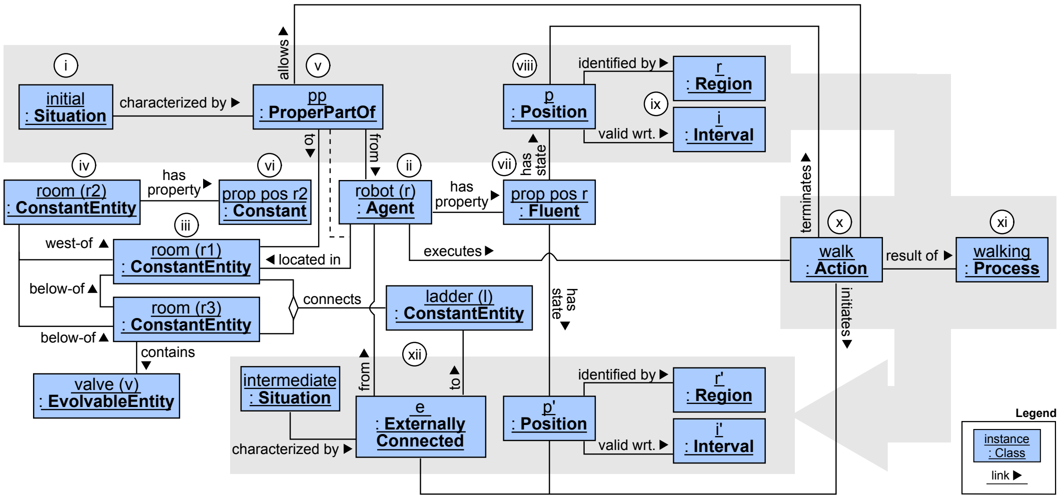


Fig. 12: The robot scenario as an object diagram of the conceptual reference model

In summary, Figure 12 models the scenario depicted in Figure 11, i. e., the robot is located in room $r1$, which is west of room $r2$; the robot wants to walk to the ladder connecting rooms $r1$ and $r3$ in order to reach room $r3$, which contains the valve. Note, that in this scenario extrinsic orientation relations (e. g., west of) are not as meaningful as in the traffic navigation scenario; a graph structure is more useful for indoor navigation, such as established through the connection relation between room $r1$ and $r3$, which are connected by the ladder l. Doors and walls connect rooms similar to ladders, but all three require different actions in order to transition from one room into another (ladders need to be climbed, doors opened and walked through, walls broken down). In the initial situation (i), the robot r (ii) is located in room $r1$ (iii) west of room $r2$ (iv). In terms of a topological relational state, this initial situation is characterized by the robot being a *proper part of* (v) room $r1$. The unary property $prop\_pos\_r2$ (vi) that holds the position of room $r2$ is constant, whereas the unary property $prop\_pos\_r$ (vii) describing the robot's position needs to be fluent. In the initial situation, the robot is located at position p (viii) identified by a spatial region r and valid w.r.t. a temporal interval i (ix). Also, being a proper part of room $r1$ allows the robot to execute a walk action (x), which is the result of a walking process that is not further detailed (xi). This walk action, which models instant motion for simplicity, has two immediate consequences: First, it terminates the unary state p (of type $Position$) of the initial situation, i. e., the robot is no longer located in its initial position. Second, it initiates two new states (e and p'), i. e., after executing the walk action the robot is still located in room $r1$, but in a new location p' and additionally externally connected (i. e., adjacent to) the ladder l (xii).

Once the robot climbs down the ladder and is close to the valve, it can continue with its primary task of shutting down the valve. In Figure 13 we illustrate how the conceptual reference model can be extended with custom fluents and states that allow us to describe the task of closing the valve.

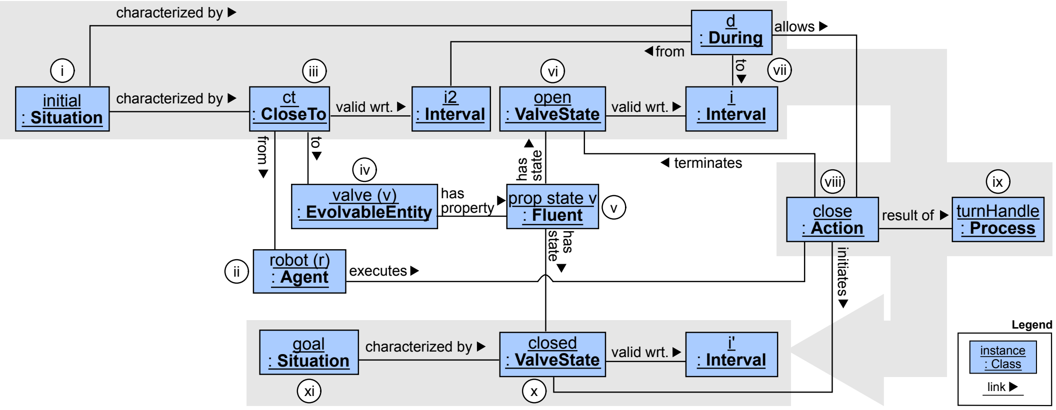


Fig. 13: A robot closes a valve when being close to it while the valve is open

In the initial situation (i), the robot r (ii) is close to (iii) the valve v (iv). The current state (v) of the valve is open (vi), which together with being close to the valve at the same time (vii) allows the robot to execute the close action (viii). The close action itself involves a process of turning the handle of the valve (ix), and when completed, the close action terminates the valve being open and initiates the valve being closed (x), which is the desired goal situation (xi).

D. DETAILED SURVEY OF MODELING IN DYNAMIC SPATIAL SYSTEMS

Based on the criteria catalog introduced in Sect. 3 we discuss logic-based modeling and reasoning approaches to autonomous agents in dynamic spatial systems.

Selection of Approaches. Our survey focuses on modeling concepts of logic-based commonsense and hybrid reasoning approaches in dynamic spatial systems. These reasoning approaches are especially valuable for controlling autonomous agents that operate as part of a dynamic spatial system and for analyzing a dynamic spatial system as a whole. With this focus, as already mentioned, we complement previous surveys on purely algebraic approaches in geographic information systems (e. g., [Worboys 2005], the family of (hybrid) process algebras (e. g., [Baeten 2005; Groote and Reniers 2001; Khadim 2008]), temporal (description) logics (e. g., [Artale and Franconi 2001; Emerson 1990; Konur 2013; Lutz et al. 2008]), model checking and simulation of hybrid systems (e. g., [Alur 2011; Casagrande and Piazza 2012; De Schutter et al. 2009]), numerical simulation (e. g., [Sulistio et al. 2004; Martinez et al. 2011]), and quantitative agent modeling (e. g., [Allan 2010; Heath et al. 2009; Nikolai and Madey 2009; Serenko and Detlor 2003]).

Interesting modeling concepts of logic-based and commonsense techniques for dynamic spatial systems can be found in each of the targeted reasoning fields:

- In fields concerned with checking *consistency* between state descriptions, for instance with a theory of dynamic spatial systems such as moving objects [Galton 1995; 2000], geographic information systems [Egenhofer and Al Taha 1992; Egenhofer and Mark 1995; Egenhofer and Wilmsen 2006; Egenhofer 2009; 2010], or in fields targeting situation awareness [Kokar et al. 2009; Matheus et al. 2003; Matheus et al. 2005a].
- In fields concerned with *planning* feasible steps in *projected* evolution towards a desired goal situation, for instance using generic methods such as the situation calculus [Bhatt 2012; Reiter 2001], event calculus [Shanahan 1997], fluent calculus [Thielscher 2005], or domain-dependent planning in robot control [Dylla and Moratz 2005; Dylla and Wallgrün 2007b; Miene et al. 2004].
- In fields concerned with analyzing a dynamic spatial system at design time by *simulating* its behavior in a qualitative manner [Apt and Brand 2005; Cui et al. 1992] or by *verifying* the correctness of its behavior using logic-based formal verification of hybrid systems [Chaochen et al. 1999; Hansen and Hung 2007; Platzer 2008; Platzer and Quesel 2008]

In order to give a comprehensive picture, we select approaches of each of these three targeted reasoning fields. We aim at a broad range and in-depth evaluation, and structure the survey in three corresponding major sections (Sect. D.1–D.3). When multiple similar approaches are available, we survey the most recent approach and merge older results into its discussion. Each approach section starts with a modeling excerpt of our sample scenario, and then discusses the criteria assessment per package of the catalog. A comparative summary of the evaluation results is given in Sect. 4.

D.1. Checking Consistency between States

D.1.1. Galton's Qualitative Theory of Movement. We begin our survey with the *qualitative theory of movement* [Galton 1995; 2000; Galton and Worboys 2005], since it is one of the earliest approaches on evolution in qualitative spatial reasoning. This theory has been developed as enhancement—termed *dominance space*—to the conceptual neighborhood theory in spatial relation calculi that had been considered as the prime means of modeling evolution constraints in spatio-temporal reasoning at that time. However, this theory of movement is rather narrowly focused in terms of spatial entities and spatial ordering, as we will see. Later approaches that build on the theoretical foundations laid here provide more comprehensive modeling and reasoning support and, thus, have been accepted for a wider range of applications. Recent advancements include discrete mereotopological relations for logic-based image analysis [Randell et al. 2013] and mining causal relations from movement [Bleisch et al. 2014]. Table XI summarizes the syntax of the qualitative theory of movement used in the subsequent modeling example.

Table XI: Syntax overview of the qualitative theory of movement

Operator	Description
$\boldsymbol{Holds}(s, i)$	State s holds throughout interval i
$\boldsymbol{Holds\text{-}at}(s, t)$	State s holds at instant t
$\boldsymbol{Occurs}(e, i)$	Event e occurs over interval i
$\boldsymbol{Occurs\text{-}at}(e, t)$	Event e occurs at instant t
$\boldsymbol{Occurs(\text{-}at)}(e, i) \triangleq \ldots$	Occurrence condition, if right-hand side holds true event e can occur
$\boldsymbol{Beg}(i), \boldsymbol{End}(i)$	Beginning and end of interval i: $i = [t, u] \leftrightarrow \boldsymbol{Beg}(i) = t \wedge \boldsymbol{End}(i) = u$
$\boldsymbol{pos}(o)$	Position of object o
$PP, \ldots BF, \ldots$	Relations of $\mathcal{RCC}$ and $\mathcal{OPRA}_m$
$f = a, f \neq a$	Propositions, fluent f equal/not equal to value a
$s \sqcap s'$	State conjunction: $\boldsymbol{Holds}(s \sqcap s', i) \leftrightarrow \boldsymbol{Holds}(s, i) \wedge \boldsymbol{Holds}(s', i)$
$-s$	State negation: $\boldsymbol{Holds}(-s, i) \leftrightarrow \neg\boldsymbol{Holds}(s, i)$

Modeling Example. Model 1 shows the initial situation of our running example with constant positions—see (unary & n-ary constants–1.1)—of the intersection and the lanes, as well as the fluent positions of the subject vehicle and the primary other vehicle (unary & n-ary fluents–1.2). The fluent positions are valid for time intervals i, i_1 and i_2, respectively. The constant positions are valid throughout the evolution of the vehicles ($i \supset \bigcup_{j=1}^{3} i_j$). As an example for such an evolution, a left-turn action with its so-called *occurrence condition* is listed in (qualification constraints–1.3). This occurrence condition represents an occurrent qualification constraint; the effect of the left-turn action is specified in formula (frame constraints–1.4). More precisely, the occurrence condition of the left-turn action is specified by conjunction of the states that must hold at the beginning $Beg(i_3)$ of the left-turn action interval: the subject vehicle must still be located on the left-turn lane, while the primary other vehicle has already moved to the outbound lane. As effect of the left-turn action (frame constraints–1.4), at time $End(i_3)$ the subject vehicle will be located on the outbound lane, just behind the primary other vehicle. Throughout the left-turn action, the subject vehicle is located on neither the left-turn lane, nor on the outbound lane.

Model 1 The sample traffic situation in terms of moving objects [Galton 2000]

$$\boldsymbol{Holds}(BF(pos(ltl), pos(x)) \sqcap LF(pos(ol), pos(x)) \sqcap \ldots, i) \tag{unary \& n-ary constants–1.1}$$

$$\boldsymbol{Holds}(PP(pos(sv), pos(ltl)), i_1) \wedge \boldsymbol{Holds}(PP(pos(pov), pos(il)), i_2) \tag{unary \& n-ary fluents–1.2}$$

$$\boldsymbol{Occurs}(\boldsymbol{LeftTurn}(sv, pos(ltl), pos(ol)), i_3) \triangleq \tag{qualification constraints–1.3}$$

$$\begin{aligned}
&\boldsymbol{Holds}\text{-}at(PP(pos(pov), pos(ol)), Beg(i_3))\\
&\wedge \boldsymbol{Holds}\text{-}at(PP(pos(sv), pos(ltl)), Beg(i_3))\\
&\wedge \boldsymbol{Holds\text{-}at}(PP(pos(sv), pos(ol)), End(i_3)) && \text{(frame constraints–1.4)}\\
&\wedge \boldsymbol{Holds\text{-}at}(BF(pos(sv), pos(pov)), End(i_3))\\
&\wedge \boldsymbol{Holds}(pos(sv) \neq pos(ltl), i_3) \wedge \boldsymbol{Holds}(pos(sv) \neq pos(ol), i_3)
\end{aligned}$$

Universe of Discourse. The theory of time given in this work builds upon the notion of *instants*, which are aggregated to *intervals* (UD.T). From a spatial viewpoint arbitrary connected topological *regions* [Galton 1995; 2000] are considered as spatial entities. In a later work on dynamic geo-networks [Galton and Worboys 2005], points are used as spatial entities (UD.S). Unfortunately, these different spatial entities are not integrated, i. e., regions and points cannot be used in conjunction but only separately. Moreover, the qualitative theory of movement is restricted to an informal discussion of a theory of entities, stating some examples, such as rigid vs. non-rigid objects, non-discrete (e. g., water), and non-concrete (e. g., shadows) entities (UD.E). As theory of properties, Galton uses *unary propositional fluents* for describing arbitrary propositions [Galton 1995]. *Functional* fluents denote concrete properties, such as speed, density and throughput [Galton and Worboys 2005], and relations between entities. These unary and n-ary fluents are defined to *hold* over either instants or intervals (UD.P).

Static View. Unary and n-ary properties can have multiple states, which hold over either instants or intervals [Galton 1995] (SV.ST). The theory of time is accompanied with a *successor* relationship on dense sets[7], defining for pairs of instants or intervals whether one precedes the other (SV.TO). The early work of Galton [1995] utilizes $\mathcal{RCC}$ [Randell et al. 1992] as spatial ordering between its spatial entities (i. e., topological regions). Subsequent work [Galton 2000; Galton and Worboys 2005] (SV.SO) discusses continuous motion in graphs, networks, or continuous spaces. These relations, however, are neither used to describe situations, nor combined to any other form of named situation concept (SV.S).

Dynamic View. Although in principle the successor relationship between temporal entities could result in a linear or a branching time structure, Galton [1995] is restricted to a *linear* time structure (DV.TS). *Motion* is considered as the prime means of spatial evolution of regions, which manifests itself in transitions between the relations of $\mathcal{RCC}$ [Galton 1995]. *Enter*, *connect* [Galton 1995], and *flow* on network links [Galton and Worboys 2005] are listed as additional occurrents between regions. None of these, however, is defined formally w.r.t. a quantitative reference frame (DV.CO). Relation transitions in the conceptual neighborhood graph of $\mathcal{RCC}$ are augmented with a so-called concept of a dominance space, which indicates whether a particular relation with respect to an occurrent must hold during an interval or may hold for only an instant (e. g., during a transition between DC and PO caused by motion, the relation EC may hold for only an instant) (DV.EX). In order to enable the construction of such dominance spaces on a formal basis, the notion of gradual change is defined in terms of the mathematical concept of a *continuous function*. Discontinuous occurrents, such as appearance and disappearance of objects, are only considered in the form of creation or removal of flow in a network [Galton and Worboys 2005], but not for regions or any other type of spatial entity (DV.DO).

Epistemic View. Concerning qualification constraints for n-ary states, the qualitative theory of movement relies on state qualification constraints in terms of exhaustiveness and disjointness (JEPD, see Sect. 2.4 page 15), symmetry, and inverseness characteristics defined by the spatial relation calculus $\mathcal{RCC}$ [Cohn et al. 1997]. Additional state qualification constraints (i. e., others than those defined in $\mathcal{RCC}$) were integrated in the theory rather late in terms of states enabling, disabling, or perpetuating other states [Galton and Worboys 2005], although they were exemplified in terms of a dominance space of a non-rigid object moving in relation to two fixed regions earlier [Galton 1995] (EV.QC). Sufficient conditions and effects can be defined for unary and

---

[7] Galton [1995] presents an additional axiom that turns dense time into discrete time, if preferred.

n-ary state changes with events that initiate or terminate states [Galton and Worboys 2005]. These events themselves must be caused to occur: As occurrent qualification constraints, states are described to enable or disable particular events [Galton and Worboys 2005], whereas the sufficient conditions are defined in terms of events causing other events [Galton and Worboys 2005] (e. g., *motion* events are considered as sufficient conditions for transitions between the relations of $\mathcal{RCC}$ [Galton 1995]) (EV.FC). Although the necessary modeling concepts are present in the theory, it lacks a concrete taxonomy of qualification and frame constraints, since a theory of entities as part of the universe of discourse is missing. Intra-property dependency and composition are inherited as ramification constraints from the composition table of $\mathcal{RCC}$ [Galton 1995], whereas inter-property dependencies can be defined in terms of states enabling events (being also the necessary condition for an event to happen) [Galton and Worboys 2005]. Inter-property composition is not considered (EV.RC).

Table XII: Summary of Theory of Movement (Galton et al.)

	Temporal Entities		**Spatial Entities**					**Physical Entities**		**Properties**
Univ. of Discourse	Instant	Interval	Point	Line	Region	Other	Kind	Description	Def.	Nature
	✓	✓	✓	–	✓	–	T	rigid, non-discrete, non-concrete	–	F

	State		**Temporal Order**		**Spatial Order**			**Situations**	
Static View	Arity	Time-dep.	Topological	Positional	Topological	Positional	Ref. frame	Implicit	Explicit
	1,+	✓	–	$\mathcal{PA}$	$\mathcal{RCC}$	✓[i]	I,E	–	–

	Time		**Continuous and Discontinuous Occurrents**						**Expressiveness**	
Dynamic View	Struc.	Dom.	Card.	Informal description	Formal	Ext.	(Dis)appear	Other	Temporal	Other
	L	$\mathbb{R}$	+	motion, enter, connect	–	–	–	–	∼	–

	Qualif. Constraints		**Frame Constraints**		**Ramification Constraints**			
Epist. View	State	Occurrent	State	Occurrent	Intra dep.	Intra comp.	Inter dep.	Inter comp.
	1,+[ii]	+[ii]	✓[iii]	✓[iii]	$\mathcal{RCC}$	$\mathcal{RCC}$	∼[iv]	–

Legend
Supported: yes (✓), definable (∼), no (–)
Cardinality: unary (1), n-ary (+)
Nature: constant (C), fluent (F)
Region kind: topological (T), geometrical (G)
Spatial: region connection calculus ($\mathcal{RCC}$)
intrinsic (I), extrinsic (E), or deictic (D)
Temporal: linear (L), branching (B), discrete ($\mathbb{N}$), dense ($\mathbb{R}$)
point algebra ($\mathcal{PA}$), interval algebra ($\mathcal{IA}$)

[i] graph distance [ii] $\mathcal{RCC}$ [iii] event [iv] state enables event

Overall, as summarized in Table XII, this theory of movement is rather narrowly focused in terms of spatial entities and spatial ordering. In the next section we therefore discuss qualitative reasoning approaches for geographic information systems.

D.1.2. Consistency in Qualitative Reasoning for Geographic Information Systems. In qualitative reasoning for geographic information systems we can find a broad discussion of the universe of discourse by means of many different relation calculi for describing a large variety of differently shaped physical entities [Egenhofer and Mark 1995; Hornsby and Egenhofer 1997; Egenhofer and Wilmsen 2006; Reis et al. 2008; Egenhofer 2009; 2010; Lewis et al. 2013]. Although research on evolution in qualitative reasoning for geographic information systems already dates back to the early 1990s (e. g., gradual change of topological relations [Egenhofer and Al Taha 1992]), current geographic information systems mostly implement consecutive snapshots of states [Worboys 2005]. Evolution between these consecutive snapshots is modeled in the form of conceptual neighborhood graphs of relation calculi. This means, that evolution is described with a single kind of occurrent: a transition between nodes in the graph. Different variants of these conceptual neighborhood graphs are built depending on various position, orientation, size, and shape deformations of spatial entities [Egenhofer 2010]. These variants describe evolution of relationships between objects under different assumptions about the evolution of these objects. In summary, qualitative reasoning approaches for geographic information systems provide a large variety of modeling concepts for representing states, which are accompanied by a rather basic kind of occurrent to describe evolution. Geographic information systems use existing logic-based approaches (e. g., the situation calculus) to model state propositions, which also provide more advanced modeling concepts for describing actions and their conditions. Thus, the modeling examples in the corresponding next sections refer to the spatial calculi of this section (e. g., to the 9-intersection calculus), instead of repeating a dedicated example in one of the subsequently discussed formalisms here.

Universe of Discourse. Although many different kinds of spatial entities were discussed—abstract regions without shape [Egenhofer and Al Taha 1992; Egenhofer and Wilmsen 2006], lines and topological regions [Egenhofer and Mark 1995], lines alone [Reis et al. 2008], compound topological objects [Egenhofer 2009], and spheres [Egenhofer 2010] (UD.S)—only one approach considers temporal instants as temporal entities [Hornsby and Egenhofer 1997] (UD.T). Spatial entities as abstractions from actual physical entities are considered more important than the physical entities themselves, since those are typically provided by applications utilizing a geographic information system. Hence, a theory of physical entities has only been discussed informally by Egenhofer [2010]: the validity of topological relations between regions depends on the size of physical entities (UD.E). For the same reasons, properties of physical entities are only considered by Hornsby and Egenhofer [1997]. These properties are constant and can be added to or removed from objects (UD.P).

Static View. None of the approaches above introduces a theory of temporal ordering, which can be attributed already to the absence of temporal entities (SV.TO). Consequently, geographic information systems built with those approaches do not support a history of entities with states being anchored in time (SV.ST). The 9-intersection calculus[8] is used as spatial ordering [Egenhofer and Herring 1991] (SV.SO). However, its relations are not used for modeling situations (SV.S).

Dynamic View. As already mentioned, only Hornsby and Egenhofer [1997] introduce a theory of temporal entities. They define a *linear* temporal structure (DV.TS). Regardless of a concrete temporal structure, occurrents are in almost all approaches [Egenhofer and Al Taha 1992; Egenhofer and Mark 1995; Egenhofer 2009; 2010; Reis et al. 2008] restricted to transitions between relations in conceptual neighborhood graphs. These transitions are refined with informal discussions of continuous occurrents in terms of the necessary topological transformations that cause relation transitions: (i) scaling of regions [Egenhofer and Al Taha 1992], (ii) translational motion of lines [Egenhofer and Mark 1995], (iii) splitting of regions [Egenhofer and Wilmsen 2006], and (iv) translational motion, scaling, rotation, and shape change of regions [Egenhofer 2010] (DV.CO). Discontinuous events are seldom considered. Only Hornsby and

---

[8] The 9-intersection calculus largely resembles $\mathcal{RCC}$ for extended topological regions, but can also be used to compare spatial entities of mixed dimensionality.

Egenhofer [1997] discuss create and destroy, as well as their refinements merge and split for regions (DV.DO). The expressiveness of continuous events, however, is limited to occurrence only (DV.EX).

Epistemic View. All approaches inherit the joint exhaustive and pairwise disjoint definition of relations and their symmetry and inverseness definitions from the underlying 9-intersection calculus for defining state qualification constraints for relational states. Occurrent qualification constraints in the form of necessary conditions for transitions between relations are specified in terms of conceptual neighborhood graphs [Egenhofer and Al Taha 1992; Egenhofer and Mark 1995; Egenhofer 2009; Reis et al. 2008]. Although Egenhofer [2010] informally discusses more detailed necessary conditions in terms of the properties of physical entities (e. g., a region must be smaller than another region), those are not formalized in the calculi (EV.QC). Neither are sufficient conditions and effects of occurrents formally included in the different calculi. Only discussions are provided, for instance that scaling and translational motion [Egenhofer and Al Taha 1992; Egenhofer 2010], as well as rotation and shape change of regions [Egenhofer 2010] are subsumed by transitions (EV.FC). Ramification constraints are defined formally in terms of *composition tables* of intra-property composition of the underlying 9-intersection calculus [Egenhofer and Al Taha 1992; Egenhofer and Mark 1995; Egenhofer 2009; Hornsby and Egenhofer 1997; Reis et al. 2008], and in terms of so-called *achievable splitting configurations* [Egenhofer and Wilmsen 2006]. Inter-property dependency is considered by Egenhofer and Wilmsen [2006], who discuss feasible relations in hierarchically composed CNGs, but not inter-property composition (EV.RC).

Table XIII: Summary of Qualitative Reasoning for Geographical Information Systems

Univ. of Discourse	**Temporal Entities**		**Spatial Entities**					**Physical Entities**		**Prop-erties**
	Instant	Interval	Point	Line	Region	Other	Kind	Description	Def.	Nature
	✓	–	–	✓	✓	✓	T	Size	–	C

Static View	**State**		**Temporal Order**		**Spatial Order**			**Situations**	
	Arity	Time-dep.	Topological	Positional	Topological	Positional	Ref. frame	Implicit	Explicit
	1,+	–	–	–	9$\mathcal{I}$	–	I	–	–

Dynamic View	**Time**		**Continuous and Discontinuous Occurrents**						**Expressiveness**	
	Struc.	Dom.	Card.	Informal	Formal	Ext.	(Dis)appear	Other	Temporal	Other
	L	–	+	move, scale, rotate, shape	–	✓	✓	split, merge	–	–

Epist. View	**Qualif. Constraints**		**Frame Constraints**		**Ramification Constraints**			
	State	Occurrent	State	Occurrent	Intra dep.	Intra comp.	Inter dep.	Inter comp.
	$+^{i}$	–	–	–	9$\mathcal{I}$	9$\mathcal{I}$	$\sim^{ii}$	–

Legend
Supported: yes (✓), definable (∼), no (–)
Cardinality: unary (1), n-ary (+)
Nature: constant (C), fluent (F)
Region kind: topological (T), geometrical (G)
Spatial: 9-intersection calculus (9$\mathcal{I}$)
intrinsic (I), extrinsic (E), or deictic (D)
Temporal: linear (L), branching (B), discrete ($\mathbb{N}$), dense ($\mathbb{R}$)

[i] 9$\mathcal{I}$ [ii] feasible relations

In summary (cf. Table XIII), qualitative reasoning for geographic information systems provide a wide variety of spatial relation calculi to model n-ary states and are thus especially suitable to describe a current situation and to check consistency between states. However, they do not provide modeling concepts for dynamic and epistemic knowledge besides conceptual neighborhood graphs and are thus less suitable for projection, planning, and simulation without additional modeling support.

Examples for such additional modeling support are the situation calculus, the event calculus, or the fluent calculus, as discussed in Sect. D.2.

D.1.3. The Ontology-based Situation Awareness System SAWA. The ontology-based situation awareness system SAWA [Kokar et al. 2009; Matheus et al. 2003; Matheus et al. 2005a] aims at increasing *situation awareness* [Endsley 2000] by highlighting relationships between entities and events. For this, SAWA provides a tool set for modeling domain knowledge in the form of ontologies and rules, for collecting information about entities and events, checking consistency of the collected information, and for monitoring and visualizing relations between these entities and events [Matheus et al. 2005b]. SAWA has been applied, for instance, in the domain of supply logistics [Matheus et al. 2005b]. It has to be noted, however, that the underlying data model of SAWA—the so-called situation theory ontology (STO)—focuses on describing states at snapshots (static view) rather than on concepts for describing evolution between these states (dynamic view), as summarized in Table XIV. Although SAWA does not yet allow modeling evolution, it still provides relevant concepts for reasoning about states (e. g., to detect inconsistencies). Hence, we provide a modeling example and an evaluation for the universe of discourse and the snapshot view criteria categories.

Table XIV: Syntax overview of the situation theory ontology STO

Operator	Description
$s \models \sigma$	Situation s supports σ
$\ll R, o_1, \ldots o_n, 0/1 \gg$	Information item (called *infon*), n-ary relation R holds true/false for objects $o_1 \ldots o_n$, we use specifically the following
$\ll p, s, v, 0/1 \gg$	Unary constant: property p of subject s has value v (always)
$\ll p, s, v, t/i, 0/1 \gg$	Unary fluent: property p of subject s has value v at instant t or throughout interval i
$\ll R, o_1, o_2, 0/1 \gg$	N-ary constant: relation R holds false/true between objects o_1 and o_2 (always)
$\ll R, o_1, o_2, t/i, 0/1 \gg$	N-ary constant: relation R holds false/true between objects o_1 and o_2 at instant t or throughout interval i
$(x_1, x_2, \ldots, x_n)$	Disjunction of $x_1 \ldots x_n$: $\bigvee_{i=1}^{n} x_i$

Modeling Example. Model 2 shows an excerpt from the initial situation as introduced in our sample scenario to demonstrate the modeling features of STO. First, the locations of the intersection and some of its lanes are defined (partial view: intersection x, left-turn lane ltl on 1st Avenue, inbound lane il on 2nd Street), cf. Model 2 (unary constant, partial view–2.1). Specifically, in STO the statement $s_{initial} \models \ll Location, x, l_x, 1 \gg$ means, that in the initial situation the location of x is l_x. Second, a priori knowledge about the intersection is described in terms of topology ($\mathcal{RCC}$) and orientation ($\mathcal{OPRA}_m$), cf. Model 2 (n-ary constant–2.2): the left-turn lane is externally connected (EC) and behind (BF) the intersection, while the inbound lane is externally connected to the right (RF), and both lanes are disconnected (DC), cf. Model 2 (n-ary constant–2.2). Next, the location of the subject vehicle sv is defined as a unary fluent along with its temporal validity, cf. Model 2 (unary fluent–2.3). Fourth,

---

Model 2 Excerpt of the sample initial traffic situation at the intersection, as observed by the subject vehicle and the traffic camera modeled in STO

$$
\begin{aligned}
s_{initial} &\models \ll location, x, l_x, 1 \gg && \text{(unary constant, partial view–2.1)}\\
s_{initial} &\models \ll location, ltl, l_{ltl}, 1 \gg \\
s_{initial} &\models \ll location, ol, l_{ol}, 1 \gg \\
s_{initial} &\models \ll (EC, BF), l_{ltl}, l_x, 1 \gg && \text{(n-ary constant–2.2)}\\
s_{initial} &\models \ll (EC, LF), l_{ol}, l_x, 1 \gg \\
s_{initial} &\models \ll DC, l_{ol}, l_{ltl}, 1 \gg \\
s_{initial} &\models \ll location, sv, l_{sv}, i_{sv}, 1 \gg && \text{(unary fluent–2.3)}\\
s_{initial} &\models \ll PP, l_{sv}, l_{ltl}, i_{sv-at-ltl}, 1 \gg && \text{(n-ary fluent–2.4)}
\end{aligned}
$$

---

the spatial state of the subject vehicle with respect to the left-turn lane is defined from a topological viewpoint using $\mathcal{RCC}$, cf. Model 2 (n-ary fluent–2.4): the subject vehicle is a proper part (PP) of the left-turn lane (i. e., it is situated on the left-turn lane).

Note, that this model does not exhaustively enumerate the relationships between all objects, since, naturally, sensors of the subject vehicle will measure only the relevant facts for turning left (e. g., the position of the subject vehicle and the orientation of the primary other vehicle). With a priori knowledge reported by the intersection and static reasoning using the composition tables of the utilized calculi, the missing information can be derived. For example, knowing that the left-turn lane is disconnected from the inbound lane ($\ll DC, l_{ltl}, l_{il}, 1 \gg$), the subject vehicle is located on the left-turn lane ($\ll PP, l_{sv}, l_{ltl}, 1 \gg$), and the primary other vehicle is located on the inbound lane ($\ll PP, l_{pov}, l_{il}, 1 \gg$, allows us to conclude that the subject vehicle and the primary other vehicle must be disconnected.

Universe of Discourse. Temporal instants and intervals are considered as temporal entities only by Matheus et al. [2003]. Most recently (e. g., [Kokar et al. 2009]), however, temporal entities are referred to with some unspecified abstract notion of time (UD.T). Neither a theory of spatial entities (UD.S), nor a concrete theory of physical entities (everything is simply an object, UD.E) are discussed in any of these approaches. The only possibility of modeling details about objects is given in terms of unary and n-ary fluents (UD.P). For example, in Model 2 (unary constant, partial view–2.1), $s_{initial} \models \ll Location, sv, l_{sv}, 1 \gg$ models that the subject vehicle sv is located at l_{sv} (unary fluent), whereas in Model 2 the formula (n-ary constant–2.2) $s_{initial} \models \ll DC, l_{ol}, l_{ltl}, 1 \gg$ models that the locations of the outbound lane ol and the left-turn lane ltl are disconnected (n-ary fluent).

Static View. Unsurprisingly, situations, such as $s_{initial}$ in Model 2, are considered as first-class citizens in the ontology in terms of *named* situations (UD.S). Situation types in STO are modeled as sets of situations, as in

$$
\begin{aligned}
\mathit{SubjectVehicleAtTrafficSignal} = \{s \mid s &\models \ll EC, l_{sv}, l_x, 1 \gg \\
\wedge\, s &\models \ll Location, sv, l_{sv}, 1 \gg \\
\wedge\, s &\models \ll Location, x, l_x, 1 \gg \}.
\end{aligned}
$$

Properties of objects (and in this approach, hence, the properties of situations) are modeled independently from their values; each value is anchored at a particular temporal entity [Matheus et al. 2003] (SV.ST). For example, in Model 2 (unary constant, partial view–2.1), $s_{initial} \models \ll Location, sv, l_{sv}, t_{sv}, 1 \gg$ models that the location l_{sv} is valid throughout t_{sv}. Due to the applied level of abstraction, however, neither concrete temporal relations (SV.TO) nor spatial relations (SV.SO) are supplied to enable temporal or spatial ordering of situations or other physical entities. These relations, just like in the example, must be provided by the modelers using STO. It is this lack of modeling support of change already on the snapshot view, which makes modeling of evolution impossible. Concerning epistemic knowledge, state qualification constraints (JEPD relations and CNG, EV.QC) and ramification constraints in the form of composition tables defining intra-property composition are supported (EV.RC).

Table XV summarizes the evaluation of the Situation Theory Ontology of the SAWA approach.

Table XV: Summary of the Situation Theory Ontology

Univ. of Discourse	**Temporal Entities**		**Spatial Entities**					**Physical Entities**		**Prop-erties**
	Instant	Interval	Point	Line	Region	Other	Kind	Description	Def.	Nature
	✓	✓	–	–	–	–	–	–	–	F
Static View	**State**		**Temporal Order**		**Spatial Order**			**Situations**		
	Arity	Time-dep.	Topolog-ical	Posi-tional	Topolog-ical	Posi-tional	Ref. frame	Implicit	Explicit	
	1,+	✓	∼	∼	∼	∼	I,E,D	–	∼	
Dynamic View	**Time**			**Continuous and Discontinuous Occurrents**					**Expressiveness**	
	Struc.	Dom.	Card.	Informal description	Formal	Ext.	(Dis)appear	Other	Temporal	Other
	–	$\mathbb{N}$	(+)[i]	–	–	✓	–	–	–	–
Epist. View	**Qualif. Constraints**		**Frame Constraints**		**Ramification Constraints**					
	State	Occurrent	State	Occurrent	Intra dep.	Intra comp.	Inter dep.	Inter comp.		
	–	–	–	–	–	–	–	–		
Legend	Supported: yes (✓), definable (∼), no (–) Cardinality: unary (1), n-ary (+) Nature: constant (C), fluent (F) Region kind: topological (T), geometrical (G) Spatial: intrinsic (I), extrinsic (E), or deictic (D) Temporal: linear (L), branching (B), discrete ($\mathbb{N}$), dense ($\mathbb{R}$)									

[i] definable

D.2. Logic-Based Planning and Projection

In this section, we discuss logic-based planning approaches, which deduce and thereby project possible future states from a current state description. We discuss the situation calculus, event calculus, and fluent calculus, as well as domain-dependent robot planners.

D.2.1. Situation and Fluent Calculus-Based Planning. The situation calculus is a "second-order language [...] for representing dynamically changing worlds" [Reiter 2001]. It supports quantification over variables of individuals in our universe of discourse (first-order), as well as over relations (second-order). The situation calculus is the basis for several action programming languages (e. g., Golog [Levesque et al. 1997], Con-Golog [De Giacomo et al. 2000], cc-Golog [Grosskreutz and Lakemeyer 2003] and IndiGolog [Giacomo et al. 2009]), and also implemented in planning algorithms, such as KPlanner [Levesque 2005] and FSAPlanner [Hu and Levesque 2009]. Lakemeyer and Levesque [2004] introduce a different logical variant with a formal semantics [Lakemeyer and Levesque 2005], which allows action formulas similar to traditional dynamic logic [Harel 1979; Pratt 1976]. An overview of various extensions of the situation calculus can be found in [Lakemeyer 2010; McCarthy 2002], for example with applications to robot control [Ferrein and Lakemeyer 2008]. Recent advances include a uniform formalization of induction in the situation calculus [Denecker and Ternovska 2007], reasoning about games [De Giacomo et al. 2010], fuzzy representations of fluents [Schiffer et al. 2012], learning situation models and narratives from noisy observations [Dubba et al. 2011], and efficient implementation (e. g., property persistance queries [Kelly and Pearce 2010] and projection queries [Ewin et al. 2014]).

The fluent calculus [Thielscher 1999] is a specification language for robots closely related to the situation calculus, but adds reasoning about knowledge to the original situation calculus (extensions to the situation calculus make such reasoning possible too). It is implemented in the constraint programming language FLUX [Thielscher 2005]. The fluent calculus was furthermore used to define an alternative semantics for the situation calculus programming language Golog [Schiffel and Thielscher 2006], which shows that the fluent calculus and the situation calculus share most of their concepts. It has also been the basis for the declarative strategy language ALPprolog [Drescher and Thielscher 2011], which can be parametrized with different action formalisms for dynamic systems (e. g., situation calculus or fluent calculus). The fluent calculus has been used, for example, for general game playing [Schiffel and Thielscher 2007], for planning using binary decision diagrams [Hölldobler and Störr 2000; Störr 2001], for automatic web service composition [Chifu et al. 2008; Chifu et al. 2009], and for modeling cooperation behavior of multiple agents [Gao et al. 2011].

Bhatt et al. [2005] and Bhatt and Loke [2008] introduced modeling extensions of the situation calculus specifically for dynamic spatial systems. These extensions are part of the constraint logic programming toolkit CLP(QS) [Bhatt et al. 2011; Schultz and Bhatt 2012]. Although there exists no formalization of these extensions for the fluent calculus, in principle they could be applied to the fluent calculus as well. Their approach focuses on the application of qualitative spatial calculi to provide prediction in terms of deductive planning and projection, as well as abductive explanation techniques, for instance to generate narratives in GIS [Bhatt and Wallgrün 2014]. Its epistemic knowledge in general, and ramification constraints in particular, exceed the approaches discussed above.

Table XVI summarizes the syntax of the situation calculus and the extensions introduced in [Bhatt et al. 2005; Bhatt and Loke 2008]. Formulas in the situation calculus are defined using the logical symbols $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$, whereas Table XVII summarizes the syntax of the fluent calculus[9].

Modeling Example. The situation calculus extension of Bhatt and Loke [2008] provides modeling concepts for situations and actions, as well as for their qualification constraints and their direct effects and non-effects. Model 3 denotes the initial scenario of our example as C_s and the goal scenario as C_f. These scenarios represent situations in terms of the situation calculus. They are characterized by so-called *dynamic physical properties* [Bhatt and Loke 2008] (e. g., ϕ_{rcc}). Dynamic physical properties can be used to describe unary fluents (e. g., *alive*[10]) as well as n-ary fluents in both propositional and functional manner, cf. (unary propositional constants–3.1) and (3.4). For functional fluents, so-called denotation sets are required, which list the feasible values of such a fluent. We assume that the relations of the spatial calculi introduced in Appendix B are defined as denotation sets for our functional n-ary fluents. In princi-

---

[9] Similar concepts to represent knowledge are also available in variants of the situation calculus (e. g., [Funge 1999]).

[10] We renamed the unary propositional fluent *exists* of Bhatt and Loke [2008] to *alive*, in order to avoid confusion with existential quantification.

Model 3 The sample scenario in the situation calculus extensions of Bhatt [2012]

$$
\begin{aligned}
C_s \equiv{}& Holds(alive(ltl), true, s) \wedge Holds(alive(x), true, s) \wedge \ldots \\
& \hfill \text{(unary propositional constants–3.1)} \\
& \wedge Holds(\phi_{rcc}(ltl, x), EC, s) \wedge Holds(\phi_{opra}(ltl, x), S, s) \\
& \hfill \text{(n-ary functional constants–3.2)} \\
& \wedge \ldots \\
& \wedge Holds(\phi_{rcc}(sv, ltl), PP, s) \wedge Holds(\phi_{rcc}(pov, il), PP, s) \\
& \hfill \text{(n-ary functional fluents–3.3)} \\
C_f \equiv{}& Holds(\phi_{rcc}(sv, ol), PP, s') \wedge Holds(\phi_{rcc}(pov, ol), PP, s') && (3.4) \\
& \wedge Holds(\phi_{rcc}(sv, pov), DC, s') \wedge Holds(\phi_{opra}(sv, pov), BF, s') \\
\vec{\Theta} ={}& [\ldots, \theta_{lt}] && (3.5) \\
Poss(\theta_{lt}, s) \equiv{}& Holds(\phi_{rcc}(sv, ltl), PP, s) \wedge Holds(\phi_{rcc}(pov, ol), PP, s) \\
& \hfill \text{(qualification constraints–3.6)} \\
Occurs(\theta_{lt}, s) \wedge Poss(\theta_{lt}, s) \supset{}& Holds(\phi_{rcc}(sv, ol), PP, Result(\theta_{lt}, s)) \\
& \hfill \text{(frame constraints–3.7)} \\
& \wedge \neg Holds(\phi_{rcc}(sv, ltl), PP, Result(\theta_{lt}, s)) \\
Result(\vec{\Theta}, s) \equiv_{def}{}& Result(\theta_{lt}, Result(..., s)) && (3.8) \\
C_s \models{}& \big(\exists s' : s' = Result(\Theta, s)\big) \supset C_f && (3.9)
\end{aligned}
$$

ple there exist variants of the situation calculus with concurrent actions (e. g., [Pinto and Reiter 1995; Reiter 1996; De Giacomo et al. 1997; Scherl 2003]) and concurrent sensing (e. g., [Zimmerbaum and Scherl 2000; Scherl 2003]). But actions in situation calculus-based planners are often treated sequentially (i. e., every action initiates a new situation). Hence, $\vec{\Theta}$ lists the actions of our sample scenario in the sequence of their occurrence, ending with the left-turn action θ_{lt}, cf. (3.5). The qualification constraint $Poss(\theta_{lt})$[11] defines the necessary conditions for the left-turn action to avoid

---

[11] Qualification constraints are called precondition axioms by Bhatt and Loke [2008].

Table XVI: Syntax overview of the situation calculus and its extensions

Operator	Description
$\equiv$	Semantic equivalence
$\equiv_{def}$	Equivalence by definition (shortcut operator definition)
$\supset$	Material implication
$\models$	Models
Holds$(p(sb), v, s)$	Boolean proposition that unary property p of subject sb has value v in situation s
Holds$(p(sb_1, sb_2, \ldots, sb_n), r, s)$	Boolean proposition that n-ary property p relates subjects $sb_1 \ldots sb_n$ by relation r in situation s.
Poss(θ, s)	Action θ is possible in situation s
Occurs(θ, s)	Boolean proposition that action θ occurs in situation s
Result(θ, s)	Result situation after executing action θ in situation s

Table XVII: Syntax overview of the fluent calculus

Operator	Description
$\equiv$	Equivalence
$\supset$	Logical entailment
$\circ$	Binary composition of states (the fluents of both arguments hold)
$State(s)$	State of the world in situation s
$KState(s, z)$	Knowledge about states: state z is a possible state in situation s, according to the knowledge of an agent
$Knows(p, s)$	Property p is known in situation s (p holds in all possible states)
$Holds(p(v), t)$	Property p holds value v at time instant t, which is an abbreviation for $\exists z : \big(State(p(v)) = p(v) \circ z\big)$
$Poss(\theta, s, t)$	Action θ is possible starting at time instant s and ending at t
$Do(\theta, s)$	Execute action θ at time instant s
$s < t$	Time instant s is before t

arbitrary execution of actions, cf. (qualification constraints–3.6). As a result of an occurrence of the left-turn action, the subject vehicle moves from the left-turn lane to the outbound lane, as modeled by the effect axiom in (frame constraints–3.7). The sequence of actions $\vec{\Theta}$ is executed by transitively applying actions to the situation arising as a result from the directly preceding action, finishing with θ_{lt}, cf. (3.8). Finally, for planning we require that the initial situation entails existence of a future situation s' that is characterized by the spatial properties C_f and can be reached by applying the actions $\vec{\Theta}$.

In Model 4 we briefly show the modeling primitives of the fluent calculus, with a focus on those concepts that are not present in the original situation calculus (namely the explicit knowledge state of an agent and the corresponding knowledge update axioms). Note that numerous extensions of the situation calculus for encoding sensing and knowledge exist (e. g., [Kelly and Pearce 2007; Funge 1999], knowledge-producing actions [Moore 1977; Scherl and Levesque 1993]), which again reflects the large similarity between the situation and the fluent calculus. Here, we highlight some interesting modeling concepts: Funge [1999] distinguishes between knowing something and knowing who knows something. Kelly and Pearce [2007] introduce *observation* as the means to obtain knowledge, and they use the predicate *possible but unobservable* to denote actions that are possible in a situation but its effects cannot be observed by agents. As can be seen in formula (unary fluents–4.1), knowledge and actions are described implicitly from the viewpoint of the subject vehicle sv, since the fluent calculus cannot refer to agents in statements about knowledge explicitly. This means, we have to treat the knowledge of other agents as information in the environment when we want to reason about the knowledge of other agents [Martin et al. 2004]. The initial situation S_0 is described by the subject vehicle being located on the left-turn lane, the primary other vehicle on the inbound lane, while other fluents may hold too, cf. z in (unary fluents–4.1). In (4.2) we specify z in more detail: we state that the subject vehicle and the primary other vehicle are not located at any other position than specified by (unary fluents–4.1). We then define the qualification constraint for the left-turn action, which may only be executed if the subject vehicle is located on the left-turn lane and the primary other vehicle already passed the intersection and is located on the outbound lane, cf. (qualification constraints–4.3). Finally, the effects of the left-turn action are defined in terms of updating the knowledge state: after executing the

Model 4 The sample traffic situation in the fluent calculus [Thielscher 2005]

$$\forall z_0 : \mathit{KState}(S_0, z_0) \equiv \exists z : \Big(\big(z_0 = \mathit{At}(\mathit{ltl}) \circ \mathit{PovAt}(\mathit{il}) \circ z\big) \tag{unary fluents–4.1}$$

$$\wedge \big(\forall x \neq \mathit{ltl} : \neg \mathit{Holds}(\mathit{At}(x), z)\big) \wedge \big(\forall y \neq \mathit{il} : \neg \mathit{Holds}(\mathit{PovAt}(y), z)\big) \wedge \ldots\Big) \tag{4.2}$$

$$\mathit{Poss}(\mathit{LeftTurn}, z) \equiv \mathit{Holds}(\mathit{At}(\mathit{ltl}), z) \wedge \mathit{Holds}(\mathit{PovAt}(\mathit{ol}), z) \tag{qualification constraints–4.3}$$

$$\mathit{Knows}\big(\mathit{Poss}(\mathit{LeftTurn}, s)\big) \supset \Big(\mathit{KState}(\mathit{Do}(\mathit{LeftTurn}, s), z') \equiv \exists z : \big(\mathit{KState}(s, z) \wedge \mathit{Holds}(\mathit{At}(z), s) \wedge z = \mathit{ol}\big)\Big) \tag{frame constraints–4.4}$$

left-turn action, the subject vehicle knows that it is located on the outbound lane, cf. (frame constraints–4.4).

Universe of Discourse. From a spatial viewpoint, both the situation and the fluent calculus do not define any particular spatial layout (grid, Euclidian, or otherwise). Any spatial layout, as long as it can be represented using boolean propositions about states and updated using actions, can be included by modelers. In one example, a cleaning robot moves through a grid-like space [Thielscher 2005], whereas Bhatt and Loke [2008] consider topological *regions* in some arbitrary space of undefined dimensionality (UD.S).

Note that in principle these regions are not restricted to spatial entities and can, therefore, also be used to express temporal entities. However, facts about relations between spatial entities and between temporal entities are unrelated and require additional modeling effort: Finzi and Pirri [2005] present an approach for processes ranging over temporal intervals in the situation calculus. The fluent calculus supports instants as temporal entities (every situation represents a distinct instant). If needed, explicit temporal properties can refer to these instances to construct intervals manually. Therefore, here we only consider temporal *instants* (the situations) as inherited from the situation calculus to model temporal entities, while intervals can be defined (UD.T).

An interesting feature of this approach is the possibility of defining a theory of physical entities by specifying their *dynamic physical properties* [Bhatt and Loke 2008]. Similar extensions could be applied to the fluent calculus, which does not support physical entities and their properties out-of-the-box (UD.E). A dynamic physical property is unary, propositional, and fluent, such as in formula (unary propositional constants–3.1), which specifies that ltl is alive in s. Besides unary propositional properties, the situation calculus and the fluent calculus [Thielscher 2005] support fluent unary and n-ary functional properties (the situation calculus uses *denotation sets* [Bhatt and Loke 2008] to restrict the admissible values). Constants are implicitly modeled with fluents that do not change their value (UD.P).

Static View. The approach discusses situations in a *descriptive* manner by providing sequences of actions. Each action results in a new situation, which denotes also a new time instant. These sequences of actions are rooted in an initial situation [Bhatt and Loke 2008]. The fluent calculus, in contrast, comes with explicit state representation as a fundamental concept in order to spare computation from an initial situation [Thielscher 2005] (SV.S). This indicates the employed theory of temporal ordering, namely a *successor* relationship between actions, and in turn between temporal instants [Bhatt et al. 2005; Bhatt and Loke 2008]. Extensions for concurrent actions in the situation calculus (e. g., [Pinto and Reiter 1995; Reiter 1996; De Giacomo et al. 1997; Scherl 2003]), as well as temporal calculi manually axiomatized in the fluent calculus result in topological temporal ordering (SV.TO). A theory of spatial ordering is in former works [Bhatt et al. 2005] based on $\mathcal{RCC}$, whereas later [Bhatt and Loke 2008] it is relaxed to being modeled with arbitrary n-ary fluents. However, this has the downside of spatial ordering not being integrated in the modeling framework in a reusable fashion. The fluent calculus follows similar practice: it does not include qualitative spatio-temporal relation calculi for reasoning about relationships between entities. If needed, such calculi have to be defined manually using axioms about fluents (SV.SO). Both calculi can be used to describe

- intrinsic relations: e. g., the left-turn lane is connected to the intersection, as in (n-ary functional constants–3.2),
- extrinsic relations: e. g., the left-turn lane is south of the intersection, as in (n-ary functional constants–3.2), and
- deictic relations: e. g., the subject vehicle is left of the intersection as seen from the viewpoint of the primary other vehicle.

Although the approach considers these n-ary functional properties as free predicate symbols, the interpretation is provided by the employed calculus (e. g., the meaning of BF is defined in $\mathcal{OPRA}_m$). The notion of situations as time instants anchors the states of properties in time, which means that each change of state is considered a new situation, while the old state still holds in the preceding situation (SV.ST).

Dynamic View. Owing to the time semantics of the situation calculus, evolution occurs in a branching time structure (DV.TS). Such evolution is modeled as transitions between relations, which can be caused by arbitrary occurrents. Unfortunately, more detailed continuous occurrents are not reusably integrated [Bhatt and Loke 2008], although occurrents for regions have been discussed earlier [Bhatt et al. 2005]: one region may *split* into two, two regions may *approach* each other, *recede* from each other, or *coalesce* to one. In the fluent calculus, continuous occurrents can be modeled using *actions*. The calculus does not prescribe any such actions (not even transitions as in qualitative calculi), but Thielscher [2005] lists several examples of a cleaning robot, such as turn, clean, and go (DV.CO).

Continuous deformation of regions is defined in terms of a continuous function, which describes the minimum distance between two points contained in different regions [Bhatt et al. 2005]. Discontinuous occurrents in terms of appearance and disappearance of continuants are an important concept in the approach for modeling suddenly arising continuants that were not present in the initial situation [Bhatt and Loke 2008]. The fluent calculus and several knowledge extensions of the situation calculus (e. g., [Lakemeyer 1996; Lakemeyer and Levesque 1998; Scherl and Levesque 2003; Kelly and Pearce 2007]) take an even more explicit approach to discontinuity: discontinuity occurs in the knowledge of an agent, triggered by a discontinuous sensing action [Thielscher 2005]. This means, a sensing action may at any time update the knowledge of an agent with arbitrary facts. Appearance and disappearance of entities may be modeled with such sensing actions (DV.DO). Actions, however, are assumed to be instantaneous. Additional properties, such as action duration, are not definable within the calculus (DV.TO).

Epistemic View. Bhatt and Loke [2008] support qualification constraints for unary and n-ary states. Unary state qualification constraints (so-called *dynamic physical constraints*) define necessary conditions for n-ary states based on the unary states of entities [Bhatt and Loke 2008]. Qualification constraints for occurrents can be given in the form of so-called *action preconditions* (*Poss*, see Model 4), while those for states are modeled by *existence of state* axioms [Thielscher 2005] (EV.QC). For instance, if the dynamic physical property rigid (represented as a unary propositional fluent) holds on a particular object, it is entailed that a n-ary functional fluent contains cannot hold, as in $\forall o, o' : \boldsymbol{Holds}(\boldsymbol{rigid}(o), s) \supset \neg \boldsymbol{Holds}(\phi_{rcc}(o, o'), \boldsymbol{PP}, s)$. The opposite may of course be true (i. e., the rigid entity is contained in some other non-rigid entity) [Bhatt et al. 2005; Bhatt and Loke 2008]. Occurrent qualification constraints in terms of so-called *precondition axioms* are supported, as illustrated in Model 3 (qualification constraints–3.6). *Continuity constraints* are a special kind of occurrent qualification constraints used to encode the transitions of CNGs: e. g., a transition to disrelated is possible, as depicted in Figure 6 on page App–4, if two regions overlap: $\forall o, o' : Poss(trans(DR, o, o'), s) \equiv \boldsymbol{Holds}(\phi_{rcc}(o, o'), PO, s)$ [Bhatt and Loke 2008]. These continuity constraints can also be viewed as necessary conditions for n-ary states based on the n-ary states of entities (are thus n-ary state qualification constraints). Unfortunately, Bhatt explicitly decides not to include a reusable taxonomy of states and constraints into the approach, since this task is considered “enormous, if not infinite” [Bhatt and Loke 2008, p. 102].

Frame constraints are included in the situation calculus as *direct effect axioms* of unary and n-ary occurrents [Bhatt and Loke 2008]. For example, $\boldsymbol{Occurs}(\theta_{lt}, s) \supset \boldsymbol{Holds}(\phi_{rcc}(\boldsymbol{sv}, \boldsymbol{ol}), \boldsymbol{PP}, \boldsymbol{Result}(\theta_{lt}, s)) \wedge \neg \boldsymbol{Holds}(\phi_{rcc}(\boldsymbol{sv}, \boldsymbol{ltl}), \boldsymbol{PP}, \boldsymbol{Result}(\theta_{lt}, s))$ means that occurrent θ_{lt} causes the subject vehicle *sv* to be on the outbound lane *ol* but no longer on the left-turn lane *ltl*. This means, that every occurrent has an effect, which is a sufficient condition—a cause—for a state. Such a cause is usually a transition in a CNG (i. e., a relation transition event causes a new relation to hold), but could be any other event trigger just as well [Bhatt and Loke 2008]. For instance, in formula (frame constraints–3.7) a left-turn action θ_{lt} causes a new relation *PP* between the subject vehicle sv and the outbound lane ol. The sufficient conditions must be provided in terms of so-called *event occurrence axioms* [Bhatt and Loke 2008]. For example, $\boldsymbol{Holds}(\phi_{rcc}(sv, ltl), \boldsymbol{PP}, s) \supset \boldsymbol{Occurs}(\theta_{lt}, s)$ states that it is sufficient to be on the left-turn lane in order to make a left turn (which is too simple to be safe, of course). The fluent calculus provides *knowledge-update* and *action effect* axioms to handle the frame problem. These axioms either update the knowledge base when a sensing action obtains new information about the environment, or when a physical action changes some state (e. g., the move action of a robot moves it to a different place). Also, actions may have conditional effects (e. g., pressing a switch may turn something on or off, depending on the previous state). However, actions may not in turn directly cause other actions in the fluent calculus (EV.FC).

Concerning ramification constraints (EV.RC), intra-property dependencies are inherited from $\mathcal{RCC}$ and specified as n-ary state qualification constraints, such as exhaustiveness and disjointness of relations (JEPD). For example, the following formula states that no other relation of $\mathcal{RCC}$ can hold true if *DC* holds [Bhatt and Loke 2008].

$$\forall o, o' : \boldsymbol{Holds}(\phi_{rcc}(o, o'), DC, s) \supset \neg \boldsymbol{Holds}(\phi_{rcc}(o, o'), \gamma, s) \text{ where } \gamma \in \{\textit{PO,PP,PPi,EQ}\}$$

Composition tables with intra-property composition constraints must be supplied in terms of *state constraints*. Finally, inter-property dependencies and inter-property composition can be defined with so-called *axioms of interaction* [Bhatt and Loke 2008]. For example, $\forall o, o' : \boldsymbol{Holds}(\phi_{rcc}(o, o'), PP, s) \supset \boldsymbol{Holds}(\phi_{size}(o, o'), <, s)$ defines that every object o must be smaller than every other object o' it is a part of.

Sensing and knowledge representation are prominent in both the situation and fluent calculus: Scherl and Levesque [1993] focus on the frame problem of knowledge-producing actions (introduced first in [Moore 1977]); their approach results in memory, i. e., knowledge of one situation carries over to subsequent situations. Knowledge-producing actions are also termed sensing actions [Zimmerbaum and Scherl 2000], with Gabaldon and Lakemeyer [2007] introducing concepts to reason about noisy sensing. Lakemeyer [1996] as well as Lakemeyer and Levesque [1998] study how an agent can reason about what it knows about the world and how it can determine what and when it needs to sense new information (referred to as *only knowing*). Kelly and Pearce [2007] represent multi-agent knowledge in the situation calculus, with the premise that knowledge follows observation. To make reasoning about temporal properties in epistemic situation calculus [Scherl and Levesque 2003] decidable, De Giacomo et al. [2013] introduce a constant bound on the number of knowledge fluents of an agent. Scherl et al. [2009] introduce knowledge regression to let knowledge expire.

Table XVIII summarizes the modeling concepts of situation and fluent calculus-based planning.

Table XVIII: Summary of Planning in the Situation Calculus and Fluent Calculus

	Temporal Entities		**Spatial Entities**					**Physical Entities**		**Properties**
	Instant	Interval	Point	Line	Region	Other	Kind	Description	Def.	Nature
Universe of Discourse	✓	∼	∼	∼	✓	∼	T	dynamic physical properties[i], fluents[ii]	✓	F

	State		**Temporal Order**		**Spatial Order**			**Situations**	
	Arity	Time-dep.	Topological	Positional	Topological	Positional	Ref. frame	Implicit	Explicit
Static View	1,+	✓	∼[ii, iii]	$\mathcal{PA}$	$\mathcal{RCC}$	∼	I,E,D	action sequence	initial[i], ✓[ii]

	Time		**Continuous and Discontinuous Occurrents**						**Expressiveness**	
	Struc.	Dom.	Card.	Informal description	Formal	Ext.	(Dis)appear	Other	Temporal	Other
Dynamic View	B	ℕ	+	approach, recede, split, coalesce	–	✓	✓	sensing, knowledge update	–	–

	Qualif. Constraints		**Frame Constraints**		**Ramification Constraints**			
	State	Occurrent	State	Occurrent	Intra dep.	Intra comp.	Inter dep.	Intra comp.[iv]
Epist. View	1,+[v, vi]	1,+[iv, vii]	✓[viii, ix]	✓[x]	$\mathcal{RCC}$	∼[xi]	∼[xii]	∼

Legend
Supported: yes (✓), definable (∼), no (–)
Cardinality: unary (1), n-ary (+)
Nature: constant (C), fluent (F)
Region kind: topological (T), geometrical (G)
Spatial: region connection calculus ($\mathcal{RCC}$)
intrinsic (I), extrinsic (E), or deictic (D)
Temporal: point algebra ($\mathcal{PA}$), interval algebra ($\mathcal{IA}$)
linear (L), branching (B), discrete (ℕ), dense (ℝ)

[i] Situation calculus [ii] Fluent calculus [iii] concurrent actions [iv] occurrence axioms [v] $\mathcal{RCC}$ [vi] existence of state axioms [vii] action precondition axioms [viii] direct effect axioms [ix] knowledge update and action effects [x] occurrence axioms [xi] state constraints [xii] axioms of interaction

D.2.2. Event Calculus-based Planning. This section discusses the event calculus [Kowalski and Sergot 1986; Chen et al. 2001; Miller and Shanahan 1996; 2002; Shanahan 1996; 1999; Shanahan and Witkowski 2004] and approaches that are based thereupon. The event calculus represents duration of events and actions primarily using temporal intervals [Shanahan 1997] and allows concurrent events; it is strongly related to the situation calculus [Kowalski and Sadri 1994; 1997] and in restricted form to temporal action logics [Mueller 2006b]. The event calculus has been used, for example, to formalize workflows [Cicekli and Yildirim 2000], to monitor web service agreements [Mahbub and Spanoudakis 2007], to track epidemic spread [Chaudet 2006], and for robot navigation [Lago-Pereira and Barros 2004; Shanahan 1999].

The event calculus can be run as a logic program in Prolog, as well as in dedicated planning tools (e. g., in the abductive event calculus planner [Shanahan 2000] or the discrete event calculus reasoner [Mueller 2006a]). Table XIX summarizes the syntax of the event calculus; its formulas use the logical symbols $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$.

Table XIX: Syntax overview of the event calculus

Operator	Description
$\equiv$	Equivalence
$\leftarrow$	Implication (right to left)
$=$	Equality test
$\textbf{\textit{Initially}}(r(\vec{s}))$	Relation r holds true between subjects $\vec{s}$ from time 0
$\textbf{\textit{HoldsAt}}(r(\vec{s}), t)$	Relation r holds true between subjects $\vec{s}$ at time t
$\textbf{\textit{Happens}}(\theta, t_{start}, t_{end})$	Action θ takes place throughout the temporal interval $[t_{start}, t_{end}]$
$\textbf{\textit{Initiates}}(\theta, r(\vec{s}), t)$	Action θ initiates relation r between subjects $\vec{s}$ at time t
$\textbf{\textit{Terminates}}(\theta, r(\vec{s}), t)$	Action θ terminates relation r between subjects $\vec{s}$ at time t
$t < t'$	Time instant t is before instant t'

Modeling Example. Model 5 demonstrates the modeling features of the event calculus with an excerpt from the initial situation and the left-turn action of our sample scenario. In this model, situations evolve from an initial situation that characterizes the setting of the scenario, just as they would in the situation calculus, cf. (relational fluents–5.1). In such a situation, unary and n-ary fluents, such as $PP(sv, ltl)$, are used to describe the state of the world. Qualification constraints for actions are given by implication (e. g., generically $Action(...) \leftarrow Precondition$), cf. (qualification constraints–5.2). Also, a qualification constraint may encode checks about whether or not a physical entity is of a particular type. Actions may additionally be composed to form so-called *compound actions*, cf. (qualification constraints–5.2), which then have non-zero duration. The effects of actions are described in terms of the states they initiate or terminate, as in equation (frame constraints–5.3).

Model 5 The sample traffic situation in the event calculus [Shanahan 1997]

$$\textbf{\textit{Initially}}(PP(sv, ltl)) \wedge \textbf{\textit{Initially}}(PP(pov, il)) \wedge \ldots \qquad \text{(relational fluents–5.1)}$$

$$\begin{aligned}
\textbf{\textit{Happens}}(\textbf{\textit{LeftTurn}}(sv, ol), t1, t3) \leftarrow\ & sv = \textbf{\textit{Car}} \qquad \text{(qualification constraints–5.2)}\\
& \wedge \textbf{\textit{HoldsAt}}(PP(sv, ltl), t1)\\
& \wedge \textbf{\textit{HoldsAt}}(PP(pov, ol), t1)\\
& \wedge \textbf{\textit{Happens}}(\textbf{\textit{Move}}(sv, x), t2) \wedge \textbf{\textit{Happens}}(\textbf{\textit{Move}}(sv, ol), t3)\\
& \wedge t1 < t2 < t3
\end{aligned}$$

$$\begin{aligned}
&\textbf{\textit{Initiates}}(\textbf{\textit{LeftTurn}}(sv, ol), PP(sv, ol), t) \qquad \text{(frame constraints–5.3)}\\
&\textbf{\textit{Terminates}}(\textbf{\textit{LeftTurn}}(sv, ol), PP(sv, ltl), t)
\end{aligned}$$

Universe of Discourse. Instants are used as primary temporal entities. For compound actions, two such instants delimit a temporal interval to describe the action's duration (UD.T). Spatial entities are not formally described in the surveyed approaches. Nevertheless, the examples reveal that some sort of extended spatial entities, such as intervals or topological regions, are assumed to be used for modeling static infrastructure (e. g., rooms), whereas dynamic entities (e. g., a robot) seem to be rather points (UD.S). Although physical entities may be checked for being of a particular type, which could be useful for associating actions with particular entities, the main characterization relies on unary and n-ary properties (UD.E). The event calculus describes properties using fluents; constants are modeled as fluents whose values do not change (UD.P).

Static View. Unary as well as n-ary states can be modeled, with n-ary states being the predominant nature (SV.ST). The event calculus uses instants as well as intervals, and hence, positional and topological temporal relation calculi may be utilized (SV.TO). Although not explicitly stated, spatial relations, just as temporal relations [Shanahan 1997], can be ordered positionally and topologically using intrinsic, extrinsic, and deictic relations (SV.SO). Analogously to the situation calculus, the event calculus describes situations by executing actions. Thus, only the initial situation can be named explicitly (SV.S).

Dynamic View. The event calculus does not distinguish between continuous and discontinuous occurrents (DV.CO,DV.DO), but it allows one to specify the duration of occurrents in detail: while primitive actions are assumed to be instantaneous, compound ones may have non-zero duration (DV.EX).

Epistemic View. State frame constraints (i. e., direct effects of actions and events) are expressed in the event calculus using *Initiates* and *Terminates* statements (EV.FC). Qualification constraints for occurrents are defined using implication as logical connective, as in (qualification constraints–5.2) (EV.QC). Ramification constraints must be formalized manually as qualification or frame constraints (EV.RC).

Table XX summarizes the modeling concepts of event calculus-based planning.

D.2.3. Qualitative Planning for Robot Control. In this section we discuss the concepts of the qualitative planning approach for robot control of Ragni and Wölfl [2005]. This approach focuses on high-level qualitative motion planning of robots with different kinematic capabilities (e. g., omnidirectional drive). This planning approach is interesting, since it introduces a discretized spatial framework (essentially a grid) and formally defines neighborhood between positions with respect to this grid. Such a spatial framework enables us to relate the semantics of different qualitative spatial calculi to each other and lets us define the effects of events and actions in a uniform manner for all calculi at once. The effects in the spatial framework then propagate into the respective spatial calculi (usually as transitions in conceptual neighborhood graphs), which are guaranteed to be consistent. Without such a spatial framework we have to manually keep spatial states consistent (e. g., model constraints that define when and how a transition in $\mathcal{OPRA}_m$ would entail a simultaneous transition in $\mathcal{RCC}$). Also note that the idea of dominance spaces of Galton [1995] can be applied here when defining the semantics of a spatial calculus.

Modeling Example. Ragni and Wölfl [2005] formulate planning as a *transformation problem* $\langle C_s, C_f, C_{s/f} \rangle$: a so-called initial scenario C_s should be transformed into a final scenario C_f without violating a constraint network $C_{s/f}$. A scenario is a set of spatio-temporal relations, which, together, describe the spatial setting at hand. Ragni and Wölfl [2005] describe scenarios from an orientation viewpoint using cardinal directions ($\mathcal{CD}$ calculus [Goyal and Egenhofer 2001], e. g., north, east, south, west) only. In order to make this approach comparable to the others, in Model 6, we use spatial

Table XX: Summary of Event Calculus-based Planning

Univ. of Discourse	Temporal Entities		Spatial Entities					Physical Entities		Properties
	Instant	Interval	Point	Line	Region	Other	Kind	Description	Def.	Nature
	✓	✓	✓	✓	✓	–	T	fluents	✓	F

Static View	State		Temporal Order		Spatial Order			Situations	
	Arity	Time-dep.	Topological	Positional	Topological	Positional	Ref. frame	Implicit	Explicit
	1,+	✓	$\mathcal{IA}$	$\mathcal{PA}$	∼	∼	I,E,D	action sequence	initial

Dynamic View	Time		Continuous and Discontinuous Occurrents						Expressiveness	
	Struc.	Dom.	Card.	Informal description	Formal	Ext.	(Dis)appear	Other	Temporal	Other
	L	–	+	∼	–	✓	–	∼	✓	–

Epist. View	Qualif. Constraints		Frame Constraints		Ramification Constraints			
	State	Occurrent	State	Occurrent	Intra dep.	Intra comp.	Inter dep.	Inter comp.
	1,+	1,+	✓[i]	–	∼	∼	∼	∼

Legend

Supported: yes (✓), definable (∼), no (–)
Cardinality: unary (1), n-ary (+)
Nature: constant (C), fluent (F)
Region kind: topological (T), geometrical (G)
Spatial: region connection calculus ($\mathcal{RCC}$)
intrinsic (I), extrinsic (E), or deictic (D)
Temporal: point algebra ($\mathcal{PA}$), interval algebra ($\mathcal{IA}$)
linear (L), branching (B), discrete ($\mathbb{N}$), dense ($\mathbb{R}$)

[i] action effects

Model 6 The sample scenario in the planning encoding of Ragni and Wölfl [2005]

$$
\begin{aligned}
C_s = \big\{ & ltl\ EC\ x^{\mathrm{i}}, il\ EC\ x, ol\ EC\ x, && \text{(n-ary constant–6.1)}\\
& ltl\ BF\ x, il\ RF\ x, ol\ LF\ x, &&\\
& sv\ PP\ ltl, pov\ PP\ il, sv\ (DC, LR)^{\mathrm{ii}}\ pov\big\}^{\mathrm{iii}} && \text{(n-ary fluent–6.2)}\\
C_f = \big\{ & sv\ PP\ ol, pov\ PP\ ol, sv\ (DC, BF)\ pov\big\} && (6.3)\\
C_{s/f} = {} & C^i_{s/f}\ b\ C^j_{s/f} && \text{(safety constraint–6.4)}\\
C^i_{s/f} = {} & \big\{pov\ PP\ ol\big\} && (6.5)\\
C^j_{s/f} = {} & \big\{sv\ PP\ x\big\} && (6.6)
\end{aligned}
$$

[i] $o_1\ R\ o_2$: relation R holds true between objects o_1 and o_2 [ii] $(R_1, \ldots, R_n)$: conjunction of relations $R_1 \ldots R_n$, $\bigwedge_{i=1}^{n} R_i$ [iii] $\{R_1, \ldots R_n\}$: disjunction of relations $R_1 \ldots R_n$, $\bigvee_{i=1}^{n} R_i$

scenarios described in $\mathcal{RCC}\times\mathcal{OPRA}_m$ (i. e., a composition of the region connection calculus $\mathcal{RCC}$ with the oriented point relations algebra $\mathcal{OPRA}_m$) and constrain temporal evolution in $\mathcal{IA}$ (the interval algebra of Allen [1983]). For example, a constraint network $C_{s/f}$ defines that some scenario may occur *before* another scenario, as in equation (safety constraint–6.4).

In Model 6, the initial scenario C_s describes the initial setting of our running example (Figure 9) by listing pairwise relationships in $\mathcal{RCC}\times\mathcal{OPRA}_m$. These relationships describe the constant relational state of lanes at the intersection, cf. Model 6 (n-ary constant–6.1), as well as the fluent relational state of the vehicles with regard to the intersection. The goal is to find a sequence of actions (e. g., motion and turning) performed by the subject and the primary other vehicle in order to reach the desired final scenario C_f, cf. Model 6 (6.3). In this final scenario, the subject vehicle must be located on the outbound lane of 2nd Street, and the primary other vehicle must be located on the same street in front of the subject vehicle. Between the start and the final scenario, the constraint ($C^i_{s/f}$ has to be satisfied *before* (b) the constraint $C^j_{s/f}$ is satisfied), cf. Model 6 (safety constraint–6.4). Together, these constraints demand that the subject vehicle yields to the primary other vehicle, since the intersection acts as a mutex and the primary other vehicle must have passed the intersection already when $C^i_{s/f}$ holds.

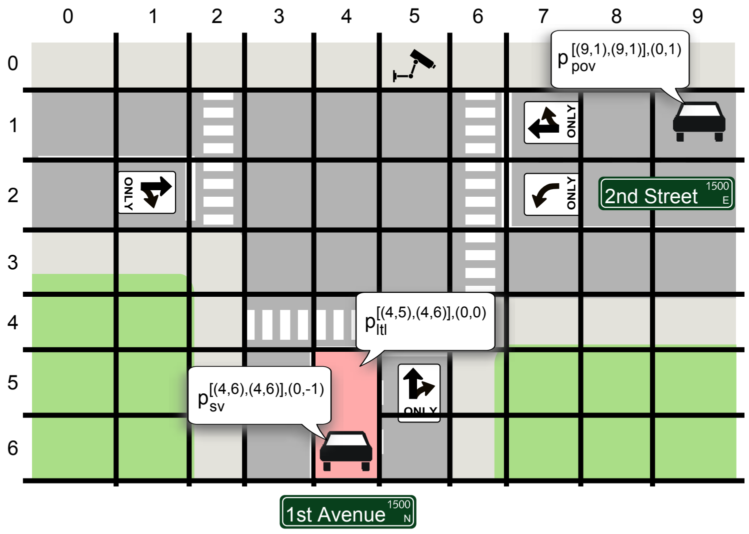


Fig. 14: Intersection scenario: grid encoding for planning

In order to specify possible actions (including their necessary conditions and effects), the first step in the planning approach of Ragni and Wölfl [2006] is to define an encoding of objects within the desired spatial setting ($\mathcal{RCC}\times\mathcal{OPRA}_m$ in our example). Note, that for this, in principle, the approach supports points, intervals, and regions for representing entities, and allows one to specify mathematical functions as actions. However, the examples given by Ragni and Wölfl [2006] use points and their motion in a two-dimensional grid for representing entities only.

Since in the sample scenario we use topological and positional relations in $\mathbb{N}\times\mathbb{N}$, in Figure 14 we therefore adapt the representation of entities from Ragni and Wölfl [2006]. Instead of point-positions we use positions in the form of rectangular spatial regions denoted by $p_x^{[(i,j),(k,l)],(h,v)}$, having $i \leq k \wedge j \leq l \wedge h, v \in \{-1, 0, 1\}$. Such a position means that

- object x occupies the spatial region, which starts at (i, j) and comprises all grid cells horizontally as well as vertically until (k, l) and
- object x is oriented in direction (h, v), which denotes the change in horizontal and vertical index caused by a motion action.

Position and orientation are thus coarse approximations of the real vehicle position and heading. The neighborhood relation N, which describes neighborhood between positions, is also adapted from Ragni and Wölfl [2006], see (6.7).

$$N\left(p_x^{[(i,j),(k,l)],(h,v)}, p_x^{[(m,n),(o,p)],(h',v')}\right) = \begin{cases} \top & \text{if } \big(m = i + h \wedge n = j + v \\ & \qquad \wedge\, o = k + h \wedge p = l + v \\ & \qquad \wedge\, (h,v) = (h',v')\big) \\ & \quad \vee \big([(i,j),(k,l)] = [(m,n),(o,p)] \\ & \qquad \wedge\, (h,v) \neq (h',v')\big) \\ \bot & \text{otherwise} \end{cases} \tag{6.7}$$

An entity may either change its position in the direction of its orientation, or it may change its orientation. Thus, the continuous occurrents that are possible within our example are translation and turning. Translation, as adapted from Ragni and Wölfl [2006], is possible under the necessary condition (6.8).

$$\begin{aligned} p_x^{[(i,j),(k,l)],(h,v)} \wedge N\left(p_x^{[(i,j),(k,l)],(h,v)}, p_x^{[(m,n),(o,p)],(h',v')}\right) \\ \wedge\, [(i,j),(k,l)] \neq [(m,n),(o,p)] \wedge (h,v) = (h',v') \end{aligned} \tag{6.8}$$

We define turning under the necessary condition (6.9).

$$\begin{aligned} p_x^{[(i,j),(k,l)],(h,v)} \wedge N\left(p_x^{[(i,j),(k,l)],(h,v)}, p_x^{[(m,n),(o,p)],(h',v')}\right) \\ \wedge\, [(i,j),(k,l)] = [(m,n),(o,p)] \wedge (h,v) \neq (h',v') \end{aligned} \tag{6.9}$$

Both, translation and turning, result in the effect $\neg p_x^{[(i,j),(k,l)],(h,v)} \wedge p_x^{[m,n),(o,p)],(h',v')}$. On the basis of object positions, as final step of Ragni and Wölfl [2006] the relations in $\mathcal{RCC}$ and $\mathcal{OPRA}_m$ must be encoded (due to reasons of brevity, we give an example for $\mathcal{RCC}$ only). We follow the approach of Egenhofer [1989] to encode topological relations in $\mathbb{N} \times \mathbb{N}$ by intersecting boundaries (∂) and interiors ($\imath$) of intervals. For example DC from $\mathcal{RCC}$ is defined by $\imath\,[(i,j),(k,l)] \cap \imath\,[(m,n),(o,p)] = \emptyset \wedge \imath\,[(i,j),(k,l)] \cap \partial\,[(m,n),(o,p)] = \emptyset \wedge \partial\,[(i,j),(k,l)] \cap \imath\,[(m,n),(o,p)] = \emptyset \wedge \partial\,[(i,j),(k,l)] \cap \partial\,[(m,n),(o,p)] = \emptyset$; the other relations are defined analogously.

Universe of Discourse. Ragni and Wölfl [2005] consider temporal instants as the only temporal entities. Later Ragni and Wölfl [2006; 2008] complement these instants with temporal intervals, as indicated in Model 6 (safety constraint–6.4) with the *before* relation of Allen [1983] (UD.T). Topological regions [Ragni and Wölfl 2005; 2008] and points [Ragni and Wölfl 2006] in two-dimensional space are used as spatial entities—as exemplified by the positions p_x above (UD.S). A theory of entities is introduced in an informal manner and on a rather abstract level in terms of *changeable* objects [Ragni and Wölfl 2005] and *size-persistent* objects (i. e., objects that cannot change their size) [Ragni and Wölfl 2008] (UD.E). Finally, properties are not explicitly distinguished into constant and fluent ones (cf. Model 6: both are modeled using the same concepts). Properties are considered constant if they are not subject to change by an action (UD.P).

Static View. The approach distinguishes between unary states (e. g., the positions p_x) and n-ary states (e. g., the relations between entities in Model 6). In order to anchor properties in time, Ragni and Wölfl [2005; 2006] use so-called temporal constraint networks $C_{s/f}$ (cf. Model 6 (safety constraint–6.4)), which define temporal relations between constraints on spatial relational properties. For example, in Model 6 the subject vehicle sv must wait for the primary other vehicle pov to pass (cf. $pov\ PP\ ol$ in $C^i_{s/f}$

(6.5)), *before* it may itself enter the intersection x (cf. $sv\ PP\ x$ in $C^j_{s/f}$ (6.6)) (SV.ST). Ordering between temporal entities (SV.TO) is achieved in these temporal constraint networks in terms of

— the point algebra [Ragni and Wölfl 2005; 2008] as a positional relation calculus
— the interval algebra of Allen [1983] as a topological relation calculus [Ragni and Wölfl 2005; 2006].

The approach applies intrinsic topological relations ($\mathcal{RCC}$ in [Ragni and Wölfl 2005; 2008]) and extrinsic positional relations (cardinal directions $\mathcal{CD}$ [Ragni and Wölfl 2006; 2008]) as spatial ordering. Our example has illustrated, that further intrinsic positional relations ($\mathcal{OPRA}_m$) are possible too (SV.SO). For example, spatial relations of $\mathcal{RCC} \times \mathcal{OPRA}_m$ are used to describe the initial scenario C_s in Model 6 (n-ary constant–6.1). These relations can only be used to describe two types of named situations: an *initial* situation C_s (cf. Model 6 (n-ary constant–6.1)) for describing the start configuration of the planning algorithm, and a *final* situation C_f (cf. Model 6 (6.3)) denoting the planning goal [Ragni and Wölfl 2006] (SV.S).

Dynamic View. Ragni and Wölfl [2005] apply a linear and discrete temporal structure (DV.TS). Discontinuous occurrents are explicitly excluded from the ontology (assuming the "same set of objects in two scenarios" [Ragni and Wölfl 2005, p. 72]). Concerning continuous occurrents, only motion of points is specified in one example [Ragni and Wölfl 2005], but the approach may be extended with others (e. g., turning, as in our example) (DV.CO).

Epistemic View. State qualification constraints (JEPD, symmetry, inverseness) are inherited from the applied qualitative spatial calculi, and may additionally be specified in the form of constraint networks (e. g., $C^i_{s/f}$ in Model 6 (6.5)). These constraint networks configure the planning algorithm [Ragni and Wölfl 2006]. The necessary conditions for position change events are defined as neighborhood between points in a grid-structure of a two-dimensional space, cf. (6.7) (EV.QC). Translational motion is a sufficient condition for a position change event to occur [Ragni and Wölfl 2006], and defines the effects in detail, cf. (6.8). Frame constraints (EV.FC) can be provided on n-ary properties in terms of:

— Restriction of the number of objects that are allowed to change, and their allowed change operation (e. g., only one side of a single region is allowed to change, resulting in various different CNGs);
— Applicability of operators [Ragni and Wölfl 2006] (defined in terms of transitions in the conceptual neighborhood graph of $\mathcal{RCC}$);
— Changes in CNGs according to size-persistency [Ragni and Wölfl 2008].

Intra-property dependencies are included as state qualification constraints. Intra-property composition is inherited from the composition tables of the applied qualitative spatial calculi [Ragni and Wölfl 2005; 2008]. Although inter-property dependencies are not considered, inter-property composition is achieved with manually defined composition tables over multiple calculi, for instance with a combined $\mathcal{RCC}\times\mathcal{CD}$ composition table [Ragni and Wölfl 2008] (EV.RC).

D.3. Logic-Based Simulation and Verification

In this section, we discuss approaches for analyzing a dynamic spatial system at design time. These include logic-based approaches to qualitative simulation, as well as logic-based verification techniques.

With respect to simulation, qualitative simulation techniques [Apt and Brand 2005; Cui et al. 1992] are most closely related to the focus of this article. We exclude numer-

Table XXI: Summary of Qualitative Planning for Robot Control

Univ. of Discourse	**Temporal Entities**		**Spatial Entities**					**Physical Entities**		**Properties**
	Instant	Interval	Point	Line	Region	Other	Kind	Description	Def.	Nature
	✓	✓	✓	–	✓	–	T	changeable, size-persistent	–	F

Static View	**State**		**Temporal Order**		**Spatial Order**			**Situations**	
	Arity	Time-dep.	Topological	Positional	Topological	Positional	Ref. frame	Implicit	Explicit
	1	✓	$\mathcal{IA}$	$\mathcal{PA}$	$\mathcal{RCC}$	$\mathcal{CD}$	I,E	–	initial, final

Dynamic View	**Time**		**Continuous and Discontinuous Occurrents**						**Expressiveness**	
	Struc.	Dom.	Card.	Informal desc.	Formal	Ext.	(Dis)appear	Other	Temporal	Other
	L	$\mathbb{N}$	(+)[i]	motion	–	✓	–	–	–	–

Epist. View	**Qualif. Constraints**		**Frame Constraints**		**Ramification Constraints**			
	State	Occurrent	State	Occurrent	Intra dep.	Intra comp.	Inter dep.	Inter comp.
	1,+[ii]	+[ii]	✓[iii]	✓[iv]	$\mathcal{RCC}$,$\mathcal{CD}$	$\mathcal{RCC}$,$\mathcal{CD}$	–	✓

Legend

Supported: yes (✓), definable (∼), no (–)
Cardinality: unary (1), n-ary (+)
Nature: constant (C), fluent (F)
Region kind: topological (T), geometrical (G)
Spatial: region connection calculus ($\mathcal{RCC}$)
cardinal directions ($\mathcal{CD}$)
intrinsic (I), extrinsic (E), or deictic (D)
Temporal: point algebra ($\mathcal{PA}$), interval algebra ($\mathcal{IA}$)
linear (L), branching (B), discrete ($\mathbb{N}$), dense ($\mathbb{R}$)

[i] definable [ii] $\mathcal{RCC}$ and $\mathcal{CD}$ [iii] motion [iv] operators (e. g., move)

ical simulation techniques, since they are covered in previous surveys (e. g., [Sulistio et al. 2004; Martinez et al. 2011]).

With respect to verification, *hybrid systems* are most closely related to dynamic spatial systems. Hybrid systems [Alur et al. 1995; Alur 2011; Branicky 1996; Davoren and Nerode 2000; Henzinger 1996; Maler et al. 1991; Platzer 2010b; 2012c] consider systems with joint discrete and continuous dynamics. In our context of dynamic spatial systems, we follow the most common type of hybrid systems: continuous physical evolution (e. g., a car moves on the road) and discrete control actions (e. g., a car can brake, accelerate, or stop). At the same time, hybrid techniques illustrate how the approaches discussed so far can be refined into more detailed system descriptions. A number of different verification approaches are being pursued for hybrid systems. We limit our attention to approaches that fit the logic-based focus of this survey[12]: the duration calculus [Chaochen et al. 1991; Hansen and Hung 2007] with its various extensions [Chaochen et al. 1993; Schäfer 2006], and differential dynamic logic [Platzer 2008; 2010b; 2012b].

D.3.1. Qualitative Envisionment-based Simulation. Envisionment-based simulation exhaustively enumerates possible future states of a dynamic system, which is described using qualitative spatial constraint calculi [Apt and Brand 2005; Cui et al. 1992]. Qualitative envisionment-based simulation is the basis for the simulation tools QSIM [Kuipers 1994] and DecSim [Clancy and Kuipers 1997], which decomposes a model into components to reduce the number of simulation states in comparison to QSIM.

---

[12] This excludes *model checking* tools, such as HyTech [Henzinger et al. 1997], PHaver [Frehse 2008], and SpaceEx [Frehse et al. 2011] since those use numerical approximations of hybrid systems instead of using logic-based analysis.

Although it is applicable only when the number of possible states is finite and small enough, qualitative envisionment-based simulation can be used when precise numerical data, which are the basis for numerical simulation, are missing [De Jong 2004]. In this survey, we focus on the approach of [Apt and Brand 2005; 2006]. For further qualitative envisionment-based simulation techniques we refer to the survey of De Jong [2004]. Apt and Brand [2005] base on the results of Cui et al. [1992], whom we refer to whenever concepts were already present in the original work. Table XXII summarizes the syntax.

Table XXII: Syntax overview of envisionment-based simulation.

Operator	Description
$\mathcal{Q}$	Set of qualitative relations
$Q\,[o,o'] = r$	Atomic formula: relations r hold true between objects o and o', $r \in \mathcal{Q}$
$Q\,[o,o'] \in \{r_1,\ldots,r_n\}$	Disjunction of relations $r_1 \ldots r_n$
$Q\,[o,o'] \in r_1 \times S$	Shortcut for $Q\,[o,o'] \in \{(r_1,s_1),\ldots(r_1,s_n)\}$
$\Box$, $\Diamond$, $\bigcirc$, U	Temporal operators (from now on, eventually, next, until)

Modeling Example. Apt and Brand [2005] interpret qualitative simulation as a constraint-satisfaction problem. They describe each simulation state by a so-called *qualitative array* Q, which denotes the qualitative relation between objects A and B with a variable $Q\,[A,B]$. In Model 7, we operate on qualitative relations Q obtained by composition of $\mathcal{RCC}$ and $\mathcal{OPRA}_m$ to define the initial and goal situation of our sample scenario, cf. (7.1). Note, that Apt and Brand [2005] utilized $\mathcal{RCC}$ and $\mathcal{CD}$ to demonstrate their approach; we replaced $\mathcal{CD}$ with $\mathcal{OPRA}_m$ to enable comparison with the other approaches evaluated in this article. Simulation states may be embedded in a temporal logic [Pnueli 1977] with operators $\Diamond a$ (eventually state a must be reached), $a\mathsf{U}b$ (a until state b is reached), $\bigcirc a$ (the next state is a), and $\Box a$ (from now on state a

Model 7 The sample scenario modeled for envisionment-based simulation

$$\mathcal{Q} = \mathcal{RCC} \times \mathcal{OPRA}_m \tag{7.1}$$

$$\begin{aligned}
C_s \equiv\; & Q\,[ltl,x] = (EC,BF) \wedge Q\,[il,x] = (EC,RF) && \text{(relational constant–7.2)}\\
& \wedge Q\,[ol,x] = (EC,LF) \wedge Q\,[ltl,il] \in DC \times \mathcal{OPRA}_m\\
& \wedge Q\,[ltl,ol] \in DC \times \mathcal{OPRA}_m \wedge Q\,[il,ol] \in DC \times \mathcal{OPRA}_m\\
& \wedge Q\,[sv,ltl] \in PP \times \mathcal{OPRA}_m \wedge Q\,[pov,il] \in PP \times \mathcal{OPRA}_m\\
& && \text{(relational fluent–7.3)}\\
& \wedge Q\,[sv,pov] \in DC \times \mathcal{OPRA}_m
\end{aligned}$$

$$C_f \equiv \Big(Q\,[sv,ltl] \in PP \times \mathcal{OPRA}_m \,\mathsf{U}\, \big(Q\,[pov,ol] \in PP \times \mathcal{OPRA}_m\big)\Big) \tag{7.4}$$

$$\wedge \Diamond\Big(Q\,[sv,x] \in PP \times \mathcal{OPRA}_m \tag{7.5}$$

$$\wedge \Diamond\big(Q\,[sv,ol] \in PP \times \mathcal{OPRA}_m \wedge Q\,[sv,pov] = (DC,BF)\big)\Big) \tag{7.6}$$

$$C_s \rightarrow \bigcirc C_f \tag{7.7}$$

holds). The initial situation C_s describes relational constants (relational constant–7.2), such as the positions of the lanes relative to the intersection, as well as relational fluents (relational fluent–7.3). The goal situation C_f expresses that the subject vehicle sv must remain on the left-turn lane ltl *until* the primary other vehicle pov has passed the intersection x, cf. (7.4). Eventually, the subject vehicle may enter the intersection (7.5) and turn onto the outbound lane ol, cf. (7.6). Additionally, Model 7 (7.7) defines a so-called inter-state constraint of the form $C_s \rightarrow \bigcirc C_f$ between the initial and the goal situation, which splits time into two distinct intervals $[0, t]$ and $[t+1, t_{max}]$ with C_s holding during the former and C_f holding during the latter.

Universe of Discourse. Apt and Brand [2005] use instants (called moments [Cui et al. 1992]) and intervals (called periods [Cui et al. 1992]) as basic temporal entities (UD.T). Although originally Cui et al. [1992] use only topological regions as spatial entities, Apt and Brand [2005] allow one to supply a theory of topological spatial entities as configuration in terms of constraints to the simulation algorithm (UD.S). Different classes of physical entities are not explicitly distinguished (UD.E). However, their properties can be defined with unary constants and fluents [Cui et al. 1992] and their relations to each other with n-ary constants and fluents, as demonstrated in equation (relational constant–7.2) and equation (relational fluent–7.3) [Apt and Brand 2005; Cui et al. 1992] (UD.P).

Static View. The simulation algorithm of Cui et al. [1992] enumerates situations from an initial situation. These situations, as already mentioned above, are represented in terms of arrays of qualitative relations [Apt and Brand 2005], (SV.S). Cui et al. [1992] base upon the interval algebra of Allen [1983] as temporal ordering. However, from these complex topological relationships, only the meets relationship is considered [Cui et al. 1992], which reduces temporal ordering to a successor relationship between temporal entities. Later, Apt and Brand [2005] close this gap and provide a full temporal logic comprising quantification operators over temporal formulas (e. g., eventually $\Diamond$, until U, and next $\bigcirc$ as demonstrated in Model 7) (SV.TO). Regions are topologically ordered using $\mathcal{RCC}$ [Cui et al. 1992] and orientation-wise using $\mathcal{CD}$ [Apt and Brand 2005; 2006] to define a spatial ordering of entities. Other orderings may be used as well, as demonstrated in Model 7 (SV.SO). Due to the implementation of the simulation algorithm, only relation calculi with temporalization in the form of CNGs are supported. Unary and n-ary states are considered to hold in a so-called state description [Cui et al. 1992], which is anchored in time (SV.ST).

Dynamic View. The described simulation algorithms base upon CNGs to exhaustively enumerate all reachable states [Cui et al. 1992]. Hence, the temporal structure is discrete and branching (DV.TS). In this branching temporal structure, continuous occurrents can be defined by supplying transitions between relations (e. g., from a CNG) as inter-state constraints [Apt and Brand 2005; Cui et al. 1992]. Besides transitions, other more concrete continuous occurrents (e. g., motion), which could be used as sufficient conditions for these relation transition occurrents are not supported (DV.CO). Discontinuous occurrents are part of the simulation algorithm of Cui et al., which supports so-called *add* and *delete* rules that introduce new or destroy existing entities during simulation [Cui et al. 1992] (DV.DO). Neither continuous nor discontinuous occurrents can be described in more detail concerning their duration or probability of occurrence, since those would result in an infinite state space (DV.EX).

Epistemic View. Qualification constraints are defined in this approach using so-called intra-state constraints [Cui et al. 1992]. Inter-state constraints [Cui et al. 1992] define the necessary conditions (i. e., occurrent qualificiation constraints) of a particular occurrent, such as a relation transition (EV.QC). Sufficient conditions are not

Table XXIII: Summary of Qualitative Envisionment-based Simulation

	Temporal Entities		**Spatial Entities**					**Physical Entities**		**Properties**
	Instant	Interval	Point	Line	Region	Other	Kind	Description	Def.	Nature
Univ. of Discourse	✓	✓	–	–	✓	–	T	–	–	C,F

	State		**Temporal Order**		**Spatial Order**			**Situations**	
	Arity	Time-dep.	Topological	Positional	Topological	Positional	Ref. frame	Implicit	Explicit
Static View	1,+	✓	$\mathcal{IA}$	–	$\mathcal{RCC}$	∼	I	transition sequence	initial

	Time			**Continuous and Discontinuous Occurrents**					**Expressiveness**	
	Struc.	Dom.	Card.	Informal description	Formal	Ext.	(Dis)appear	Other	Temporal	Other
Dynamic View	B	ℕ	+	–	–	✓	✓	–	–	–

	Qualif. Constraints		**Frame Constraints**		**Ramification Constraints**			
	State	Occurrent	State	Occurrent	Intra dep.	Intra comp.	Inter dep.	Inter comp.
Epist. View	1,+[i]	1,+[ii]	–	–	✓[i]	✓[ii]	–	–

Legend
Supported: yes (✓), definable (∼), no (–)
Cardinality: unary (1), n-ary (+)
Nature: constant (C), fluent (F)
Region kind: topological (T), geometrical (G)
Spatial: region connection calculus ($\mathcal{RCC}$)
intrinsic (I), extrinsic (E), or deictic (D)
Temporal: point algebra ($\mathcal{PA}$), interval algebra ($\mathcal{IA}$)
linear (L), branching (B), discrete (ℕ), dense (ℝ)

[i] intra-state constraints [ii] inter-state constraints

definable (EV.FC). Only intra-calculus composition axioms are supported as indirect effects of occurrents [Apt and Brand 2005; Cui et al. 1992]. These, however, are not restricted to composition tables, but instead allowed to comprise additional free axioms (e. g., proper-part between two objects implies equality between two other, unrelated objects) (EV.RC).

In summary (cf. Table XXIII), qualitative envisionment-based simulation is especially helpful when data is insufficient for numerical simulation [De Jong 2004].

D.3.2. Verification with the Duration Calculus. Duration calculus (DC [Chaochen et al. 1991; Hansen and Hung 2007]) is an interval logic that was used for case studies in embedded systems. Several extensions exist to describe repetitive behavior (DC* [Guelev and Van Hung 2005]), hybrid systems (HDC [Chaochen et al. 1993]), and shapes (Shape Calculus [Schäfer 2006]). Duration calculus specifications can be verified using model checking techniques when rewritten as automata [Fränzle and Hansen 2008; Meyer et al. 2008], transformed to more implementation-like languages [Olderog and Schenke 1995], and used to synthesize controllers [Fränzle 1996]. Table XXIV summarizes the syntax of the duration calculus.

Modeling Example. To illustrate the duration calculus, we model an excerpt of the sample scenario up to the point where the primary other vehicle reaches the intersection, cf. Model 8. This model loosely follows the cat and mouse model of Chaochen et al. [1993]. Schäfer [2006] and Hansen and Hung [2007] describe additional modeling case studies of railroad crossings, car platooning, and communication protocols. For the sake of simplicity, we assume that the subject vehicle and the primary vehicle are located on the same straight line with some distance to the intersection.

We model our sample traffic scenario in the duration calculus as an initial situation followed by driving, both over non-empty temporal intervals, cf. (8.1). Thus, the

Table XXIV: Syntax overview of the duration calculus

Symbol	Description
$\llbracket p \rrbracket$	Predicate p
$^\frown$	Temporal chop operator
$\mathit{Continuous}(p)$	Assertion: p is continuous
$\rightleftharpoons$	Equivalence
$b.f$, $e.f$	Value of function f at the beginning/end of a non-empty interval
$\dot{v} = a$	Differential equation

Model 8 Excerpt of the sample traffic situation in duration calculus

$$\llbracket \mathit{Initial} \rrbracket ^\frown \llbracket \mathit{Drive} \rrbracket \tag{8.1}$$

$$P_{sv}, P_{pov} : \mathit{Time} \to \mathbb{R} \tag{8.2}$$

$$\mathit{ProperPart}_{sv,ltl} : \mathit{Time} \to \{0,1\} \tag{8.3}$$

$$\mathit{Initial} \rightleftharpoons \mathit{ProperPart}_{sv,ltl} \wedge b.P_{pov} = SP_{pov} \wedge b.P_{sv} = SP_{sv} \tag{8.4}$$

$$\mathit{Drive} \rightleftharpoons \mathit{Drive}_{sv} \wedge \mathit{Drive}_{pov} \tag{8.5}$$

$$\mathit{Drive}_{pov} \rightleftharpoons b.P_{pov} = SP_{pov} \wedge \llbracket \dot{P}_{pov} = V_{pov} \rrbracket \wedge \llbracket \mathit{Continuous}(P_{pov}) \rrbracket \tag{8.6}$$

$$\mathit{Drive}_{sv} \rightleftharpoons \begin{cases} & b.P_{sv} = SP_{sv} \wedge SP_{sv} < P_x \wedge \llbracket \mathit{Continuous}(P_{sv}) \rrbracket \\ \wedge & P_{pov} < P_x \to \llbracket \dot{P}_{sv} = 0 \rrbracket \end{cases} \tag{8.7}$$

overall scenario lasts over an interval that is *chopped* ($^\frown$) into two subintervals. During the first subinterval, *Initial* holds throughout, on the second subinterval *Drive* is true throughout. Unary properties can be defined using real-valued time-dependent state variables, cf. the positions of the subject vehicle and the primary other vehicle in (8.2). N-ary properties are defined as boolean time-dependent state variables, cf. $\mathit{ProperPart}_{sv,ltl}$ in (8.3). $\mathit{ProperPart}_{sv,ltl}$ indicates for every time instant whether or not the subject vehicle is a proper part of the left-turn lane. Unary and n-ary properties can be linked. For example, to express that the subject vehicle is a proper part of the left-turn lane if and only if its position is between the left-turn lane's lower and upper bound, we use $ProperPart_{sv,ltl} \leftrightarrow \underline{P}_{ltl} < P_{sv} < \overline{P}_{ltl}$.

Model 8 refines in (8.4) the definition of *Initial*: initially, (i) the subject vehicle is located on the left-turn lane, (ii) the positions of the subject vehicle and of the primary other vehicle at time point b are equal to some starting positions SP, and (iii) the primary other vehicle is already closer to the intersection than the subject vehicle. *Drive* in (8.5) details the behavior of the subject vehicle and the primary other vehicle:

- The primary other vehicle drives with velocity V_{pov}. Its position P_{pov} changes as a continuous function of time at the starting position SP_{pov}, cf. (8.6).
- The subject vehicle's behavior, defined in (8.7) is slightly more complicated to demonstrate qualification constraints. Unless the primary other vehicle already passed the intersection, the subject vehicle must be stopped (qualification constraint $P_{pov} < P_x \to \dots$). This only makes sense when the starting position of the subject vehicle is in front of the intersection. Note, that the model says nothing about the behavior after the primary other vehicle pov passed the intersection; the subject vehicle may or may not drive at this point.

Universe of Discourse. The duration calculus uses instants as well as intervals as temporal entities (UD.T) [Hansen and Hung 2007]. Instants are used to assign state to properties at particular time instants, whereas intervals are used to denote the *duration* of a state. The duration calculus allows one to model spatial entities using *time-dependent variables*; an example is given in (8.2), which defines the positions of vehicles as points. Further concrete geometric spatial entities may be defined manually using multiple time-dependent variables (UD.S). Since the duration calculus is a first-order logic it has no concepts to represent higher-order physical entities [Hansen and Hung 2007]. However, the shape calculus supports compound regions and shapes for abstract spatial entities (UD.E) . The duration calculus keeps track of properties with so-called *state variables* [Hansen and Hung 2007]. These state variables are fluent, but when left unchanged they are then called *rigid* [Chaochen et al. 1999] and can capture constant state too (UD.P).

Static View. Function symbols and relation symbols of arity ≥ 0 assign values to state variables. Boolean combinations (*state expressions*) of state variables are used to model combined states in a system. State variables and expressions can be defined over temporal instants (e. g., $pos_{sv} : \mathbb{T}ime \to \mathbb{R}^2$), whereas atomic formulas (e. g., $\int_b^e Drive(t)dt \leq e - b$) are true, if $Drive$ is true throughout the interval $[b, e]$. Atomic formulas can be connected using the usual logical connectives and quantifiers of predicate logic. Thus, state is either of *unary* or *n-ary* nature (SV.ST) and valid at a particular point in time or over a temporal interval [Hansen and Hung 2007]. Relation symbols can be introduced to define positional temporal ordering with the usual relation symbols on $\mathbb{R}$ [Hansen and Hung 2007]. Additionally, an encoding of interval algebra ($\mathcal{IA}$) is given [Chaochen et al. 1993], which enables topological temporal relations (SV.TO). Likewise, an intrinsic positional spatial ordering can be defined easily on $\mathbb{R}$. Boolean relation symbols for further topological, extrinsic or deictic spatial relations may be defined manually (SV.SO). Just as physical entities, situations cannot be represented since the duration calculus is a first-order language (SV.S).

Dynamic View. The duration calculus uses a linear temporal structure. Unlike many other approaches available at the time of its proposal, the duration calculus uses dense time (DV.TS). It provides *modal operators* of the form $\phi^\frown\psi$, which divides an interval into two intervals[13], $\Diamond\phi$ (for some sub-interval ϕ is true), and $\Box\phi$ (for all sub-intervals ϕ is true). Continuous occurrents (DV.CO: focusing unary ones, e. g., (8.6) defines motion of the primary other vehicle) can be defined with state expressions containing linear differential equations. Discontinuous occurrents (DV.DO) are supported by including a qualification constraint that specifies a time instant that must be passed before a particular entity becomes active [Hansen and Hung 2007]. The duration calculus, as already suggested by its name, focuses on the duration of occurrents (DV.EX).

Epistemic View. Qualification constraints for states and for occurrents are given in state expressions [Hansen and Hung 2007], cf. (8.7) for an example of qualification constraints for motion of the subject vehicle (EV.QC). Since the focus of the duration calculus is on analyzing all possible model behaviors, sufficient conditions for occurrents are usually not listed. The effects of occurrents can be defined with linear differential equations using the hybrid duration calculus [Chaochen et al. 1993], otherwise the effects are of discrete nature. No other effects besides explicitly specified occurrents can

Table XXV: Summary of the Duration Calculus

Univ. of Discourse	Temporal Entities		Spatial Entities					Physical Entities		Properties
	Instant	Interval	Point	Line	Region	Other	Kind	Description	Def.	Nature
	✓	✓	✓	∼	∼	∼	T,G	–	–	C,F

Static View	State		Temporal Order		Spatial Order			Situations	
	Arity	Time-dep.	Topological	Positional	Topological	Positional	Ref. frame	Implicit	Explicit
	1,+	✓	$\mathcal{IA}$	✓	✓[i]	✓	I,E,D	–	–

Dynamic View	Time		Continuous and Discontinuous Occurrents						Expressiveness	
	Struc.	Dom.	Card.	Informal description	Formal	Ext.	(Dis)appear	Other	Temporal	Other
	L	ℝ	1	–	✓[ii]	✓	–	–	✓	–

Epist. View	Qualif. Constraints		Frame Constraints		Ramification Constraints			
	State	Occurrent	State	Occurrent	Intra dep.	Intra comp.	Inter dep.	Inter comp.
	1,+	1,+	–	✓[ii]	∼	∼	∼	∼

Legend
Supported: yes (✓), definable (∼), no (–)
Cardinality: unary (1), n-ary (+)
Nature: constant (C), fluent (F)
Region kind: topological (T), geometrical (G)
Spatial: intrinsic (I), extrinsic (E), or deictic (D)
Temporal: point algebra ($\mathcal{PA}$), interval algebra ($\mathcal{IA}$)
linear (L), branching (B), discrete (ℕ), dense (ℝ)

[i] in the Shape Calculus extension [Schäfer 2006] [ii] linear differential equations

occur (EV.FC). Finally, ramification constraints can be defined manually to interlink state variables (EV.RC). Table XXV summarizes the duration calculus.

D.3.3. Verification with Differential Dynamic Logic. Differential dynamic logic (d$\mathcal{L}$ [Platzer 2008; 2010b; 2012b; 2015]) supports specifying hybrid systems and correctness theorems about them. It has a more precise model of dynamics and control, but less abstract qualitative expressions than the qualitative approaches for consistency checking, planning, and simulation discussed above. Thus, d$\mathcal{L}$ is a first step to link the abstract approaches of the previous sections to real systems and implementations. Recent advancements include support for stochastic hybrid systems [Platzer 2011], distributed hybrid systems [Platzer 2012a], hybrid games [Platzer 2013], invariant generation [Ghorbal and Platzer 2014; Ghorbal et al. 2014; 2015], reasoning about refactoring [Mitsch et al. 2014a], and monitoring real systems for model compliance [Mitsch and Platzer 2014]. d$\mathcal{L}$ was used, for example, to verify properties about road traffic [Loos et al. 2011; Mitsch et al. 2012], air traffic maneuvers [Platzer and Clarke 2009; Loos et al. 2013; Jeannin et al. 2015], medical robotics [Kouskoulas et al. 2013], and robotic obstacle avoidance [Mitsch et al. 2013; Mitsch et al. 2014b]. The syntax of d$\mathcal{L}$ formulas is specified by the following EBNF grammar (where $\sim \in \{<, \leq, =, \neq, \geq, >\}$ and θ_1, θ_2 are arithmetic expressions in $+, -, \cdot, /$ over the reals, and hybrid programs α follow the hybrid program representation of Table XXVI):

$$\phi ::= \theta_1 \sim \theta_2 \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \rightarrow \psi \mid \forall x \phi \mid \exists x \phi \mid [\alpha]\,\phi \mid \langle\alpha\rangle\phi$$

Modeling example. In Model 9, we have a notion of space that is $\mathbb{R}^2$ and consider the subject vehicle and the primary other vehicle to be points in this space. Their dynamics are phrased as differential equations, in which acceleration is the control variable. The property that we want to prove is that the subject vehicle and the primary other

---

[13] In the Shape Calculus [Schäfer 2006] extension the operator $\frown$ chops a polyhedron in space-time.

Table XXVI: Hybrid program representations of hybrid systems

Statement	Description
$\alpha;\ \beta$	Sequential composition, first run α, then β
$\alpha \cup \beta$	Nondeterministic choice, following either α or β
α^*	Nondeterministic repetition, repeats α $n \geq 0$ times
$x := \theta$	Assign value of term θ to variable x (discrete jump)
$x := *$	Assign arbitrary real number to variable x
$\big(x_1' = \theta_1, \ldots,$ $x_n' = \theta_n \ \&\ F\big)$	Evolve x_i along differential equation system $x_i' = \theta_i$ restricted to maximum evolution domain F

vehicle will not crash. In Model 9, we study whether the subject vehicle makes a safe left-turn (i. e., it will wait for the primary other vehicle to pass, and then follow with adaptive cruise control). For this, the subject vehicle may only choose its acceleration freely, if it remains capable of braking to a complete stand still at the intersection, cf. Model 9 (qual. constraint–9.2). Otherwise, it may choose to stand still if it already does so (i. e., wait for the primary other vehicle to pass) as in Model 9 (9.3); braking, as a safety measure, is always allowed, cf. Model 9 (9.4). Finally, the subject vehicle may follow adaptive cruise control if the subject vehicle already passed the intersection, cf. Model 9 (9.5). We refer to [Loos et al. 2011] for details on the adaptive cruise control model, which we just abbreviate here. Since the primary other vehicle has right-of-way, it may choose its acceleration freely, cf. Model 9 (9.6). Besides the behavior of both, the subject vehicle and the primary other vehicle, Model 9 specifies the dynamics of the intersection system, cf. Model 9 (frame constraints–9.7): both vehicles follow ideal-world physics and change their positions and velocities according to their current accelerations, while time t is measured with a variable of constant slope 1.

Universe of Discourse. Time is typically measured in d$\mathcal{L}$ with real-valued variables of constant slope 1 (i. e., clocks—cf. t in Model 9 (frame constraints–9.7)), which results in a dense linear time structure [Loos et al. 2011; Mitsch et al. 2012] (UD.T). The lifespan of an object is usually not considered in differential dynamic logic, but only plays a role in its extension quantified differential dynamic logic [Platzer 2012a]. Concerning spatial entities, points are the most prominently used ones [Loos et al. 2011; Mitsch et al. 2012]. However, using multiple variables in d$\mathcal{L}$, concrete geomet-

Model 9 Excerpt of the sample traffic situation in differential dynamic logic

$$vsl \equiv (ctrl_{sv}; ctrl_{pov}; dyn)^* \tag{9.1}$$

$$ctrl_{sv} \equiv\ ?x_{sv} + \frac{v_{sv}^2}{2B} + \left(\frac{A}{B}+1\right)\left(\frac{A}{2}\varepsilon^2 + v_{sv}\varepsilon\right) < x_x; \qquad \text{(qual. constraint–9.2)}$$

$$a_{sv} := *;\ ?(0 \leq a_{sv} \leq A)$$

$$\cup\ ?(v_{sv} = 0); a_{sv} := 0 \tag{9.3}$$

$$\cup\ a_{sv} := -B \tag{9.4}$$

$$\cup\ ?(x_{pov} > x_x); a_{sv} := \text{adaptive cruise control [Loos et al. 2011]} \tag{9.5}$$

$$ctrl_{pov} \equiv a_{pov} := *; ?(-B \leq a_{pov} \leq A) \tag{9.6}$$

$$dyn \equiv (t := 0;\ x_{sv}' = v_{sv},\ v_{sv}' = a_{sv} \qquad \text{(frame constraints–9.7)}$$

$$x_{pov}' = v_{pov},\ v_{pov}' = a_{pov}, t' = 1 \ \&\ t \leq \varepsilon \wedge v_{sv} >= 0)$$

ric spatial entities, such as rectangular regions, circles, ellipsoids, and algebraic varieties can be described. Topological regions are definable using sorts in quantified differential dynamic logic for boundaries and interiors of sets (UD.S). While the representation of complicated regions is straightforward, the verification complexity is lower for simpler regions, so that linear and quadratic regions are used most in practice. Different classes of physical entities can be represented with sorts in quantified differential dynamic logic [Platzer 2012a]; even without sorts, their structure can be described in detail using unary constants[14] (e. g., x_x does not change in Model 9) and fluents (e. g., evolution of the position of the subject vehicle x_{sv} detailed in Model 9 (frame constraints–9.7)). N-ary properties can be derived from unary ones using the usual relational operators on $\mathbb{R}$ (UD.P). The behavior of objects can be modeled using control structures, such as conditional expressions, property value assignments, non-deterministic choices, and non-deterministic repetition (UD.E). For example, the behavior of the subject vehicle is described in Model 9 (qual. constraint–9.2)–(9.5) as a non-deterministic choice ($\cup$) between four alternatives. These alternatives are specified using random value assignments (e. g., $a_{sv} := *$ assigns an arbitrary value to a_{sv}) and state assertions (e. g., $?(0 \leq a_{sv} \leq A)$ restricts the acceleration a_{sv} to nonnegative values up to A).

Static View. Polynomial arithmetic expressions define temporal orderings, since temporal entities are represented as clock variables (SV.TO). Analogously, quantitative spatial orderings via polynomial arithmetic expressions compare spatial entities from a static viewpoint (SV.SO). Situations are considered implicitly in terms of starting conditions, switching conditions, and invariants, i. e., logical formulas that must hold during all evolution steps (i. e., one may specify the characteristics of a *safe* situation), but not as named first-class entities (SV.S). Finally, property history can be modeled by explicit variables that remember the values of previous iterations (SV.ST).

Dynamic View. As discussed above, variables with constant slope 1 (clocks) measure *linear* time. Branching time can be emulated manually by nested modal operators $[\alpha]$ and $\langle\alpha\rangle$. The proof calculus for d$\mathcal{L}$ (which is implemented in the verification tool KeYmaera [Fulton et al. 2015; Platzer and Quesel 2008]) enumerates symbolic representatives of all possible evolutions in a *branching* structure (DV.TS). Continuous occurrents are specified by discrete assignment to variables (i. e., control decisions or actions, if you will), while the effects of these actions are specified using linear and non-linear differential equations, see below (DV.CO). Extensions for differential-algebraic equations and differential inequalities exist [Platzer 2010a]. Discontinuous occurrents (e. g., occurrence, disappearance of objects) were shown in a recent proof of adaptive cruise control systems [Loos et al. 2011] in quantified differential dynamic logic (DV.DO). Expressiveness of occurrents is currently merely seen from the viewpoint of time (DV.EX).

Epistemic View. Qualification constraints for changing state and taking actions are either defined on unary fluents, such as $?(v_{sv} = 0$ in Model 9 (9.3) for entering the stand-still phase of the subject vehicle), or on n-ary fluents, such as the distance between the intersection and the subject vehicle $?x_x - x_{sv} > \ldots$, as in (qual. constraint–9.2). Additionally, one may specify the evolution domain of a differential equation in order to exclude invalid states during continuous evolution (e. g., in (frame constraints–9.7) $v_{sv} \geq 0$ prevents the subject vehicle from moving backwards when the brakes are hit during stand-still). Since d$\mathcal{L}$ is a first-order logic, higher ob-

---

[14] Note, that constant properties are seldom included in models, however, because they do not contribute to the evolution of the system.

jects are not considered (EV.QC). Hence, qualification constraints need to be encoded entirely in terms of constraints on their respective state variables. Concerning frame constraints, the effects of continuous occurrents can be modeled with differential equations (cf. evolution of vehicles in Model 9 (frame constraints–9.7)); thus, more detailed evolution descriptions are possible than with strictly qualitative approaches (EV.FC). Ramification constraints for relational operators on $\mathbb{R}$ are integrated into the proof calculus. These include inter-property dependency and composition, such as transitivity, symmetry, and inverseness of the operators. Further ramification constraints (e. g., for manually represented qualitative relation calculi) must be added to the model explicitly and mapped to ramification constraints for relational operators (EV.RC). For example, to represent transitivity of the equals operator of $\mathcal{IA}$, we can define a new external function with two parameters (one for an interval's begin, one to denote its end) and create a proof taclet that replaces equality in $\mathcal{IA}$ with equality of the interval boundaries in $\mathbb{R}$. Differential dynamic logic is summarized in Table XXVII.

Table XXVII: Summary of Differential Dynamic Logic

Univ. of Discourse	**Temporal Entities**		**Spatial Entities**					**Physical Entities**		**Properties**
	Instant	Interval	Point	Line	Region	Other	Kind	Description	Def.	Nature
	✓	~	✓	~	~	~	T,G	~[i]	✓	C,F

Static View	**State**		**Temporal Order**		**Spatial Order**			**Situations**	
	Arity	Time-dep.	Topological	Positional	Topological	Positional	Ref. frame	Implicit	Explicit
	1	✓	~	✓	~	✓	I	–	–

Dynamic View	**Time**		**Continuous and Discontinuous Occurrents**						**Expressiveness**	
	Struc.	Dom.	Card.	Informal description	Formal	Ext.	(Dis)appear	Other	Temporal	Other
	L,B[ii]	$\mathbb{R}$	1	–	✓[iii]	✓	✓[iv]	✓	✓[v]	–

Epist. View	**Qualif. Constraints**		**Frame Constraints**		**Ramification Constraints**			
	State	Occurrent	State	Occurrent	Intra dep.	Intra comp.	Inter dep.	Inter comp.
	1[vi]	1[vii]	✓[viii]	✓[vii]	~	~	~	~

Legend

Supported: yes (✓), definable (~), no (–)
Cardinality: unary (1), n-ary (+)
Nature: constant (C), fluent (F)
Region kind: topological (T), geometrical (G)
Spatial: intrinsic (I), extrinsic (E), or deictic (D)
Temporal: linear (L), branching (B), discrete ($\mathbb{N}$), dense ($\mathbb{R}$)

[i] Sorts in quantified differential dynamic logic [Platzer 2012a] [ii] by nested [] and ⟨⟩ [iii] linear and non-linear differential algebraic equations [iv] quantified differential dynamic logic [Platzer 2012a] [v] temporal dynamic logic [Platzer 2010b] [vi] evolution domain [vii] state test/check [viii] differential equations

REFERENCES

D. J. Abel and B. C. Ooi (Eds.). 1993. *Advances in Spatial Databases, Third International Symposium, SSD'93, Singapore, June 23-25, 1993, Proceedings*. LNCS, Vol. 692. Springer.

L. C. Aiello, J. Doyle, and S. C. Shapiro (Eds.). 1996. *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96), Cambridge, Massachusetts, USA, November 5-8, 1996*. Morgan Kaufmann.

R. J. Allan. 2010. *Survey of Agent Based Modelling and Simulation Tools*. Technical Report DL-TR-2010-007. Computational Science and Engineering Department, STFC Daresbury Laboratory.

J. F. Allen. 1983. Maintaining Knowledge about Temporal Intervals. *Commun. ACM* 26, 11 (1983), 832–843.

R. Alur. 2011. Formal verification of hybrid systems. In *Proc. of the 11th Int. Conf. on Embedded Software*, S. Chakraborty, A. Jerraya, S. K. Baruah, and S. Fischmeister (Eds.). ACM, 273–278.

R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. 1995. The algorithmic analysis of hybrid systems. *Theor. Comput. Sci.* 138, 1 (1995), 3–34.

K. R. Apt and S. Brand. 2005. Constraint-Based Qualitative Simulation. In *Proc. of the 12th Int. Symp. on Temporal Representation and Reasoning*. IEEE, 26–34.

K. R. Apt and S. Brand. 2006. Infinite Qualitative Simulations by Means of Constraint Programming. In *CP (LNCS)*, Frédéric Benhamou (Ed.), Vol. 4204. Springer, 29–43.

A. Artale and E. Franconi. 2001. A survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence* 30, 1-4 (2001), 171–210.

J. C. M. Baeten. 2005. A brief history of process algebra. *Theor. Comp. Sci.* 335, 2-3 (2005), 131–146.

M. Bhatt. 2012. *Reasoning about Space, Actions and Change*. IGI Global, Chapter 9, 284–320.

M. Bhatt, J. H. Lee, and C. Schultz. 2011. CLP(QS): A Declarative Spatial Reasoning Framework. In *COSIT (LNCS)*, M. J. Egenhofer, N. A. Giudice, R. Moratz, and M. F. Worboys (Eds.), Vol. 6899. Springer, 210–230.

M. Bhatt and S. Loke. 2008. Modelling Dynamic Spatial Systems in the Situation Calculus. *Spatial Cognition and Computation* 8, 1 (2008), 86–130.

M. Bhatt, W. Rahayu, and G. Sterling. 2005. Qualitative Simulation: Towards a Situation Calculus based Unifying Semantics for Space, Time and Actions. In *Conf. on Spatial Information Theory*. Springer.

M. Bhatt and J. O. Wallgrün. 2014. Geospatial Narratives and Their Spatio-Temporal Dynamics: Commonsense Reasoning for High-Level Analyses in Geographic Information Systems. *ISPRS Int. J. Geo-Information* 3, 1 (2014), 166–205.

R. Billen and E. Clementini. 2004. A Model for Ternary Projective Relations between Regions. In *EDBT (LNCS)*, E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Böhm, and E. Ferrari (Eds.), Vol. 2992. Springer, 310–328.

R. Billen and E. Clementini. 2006. Projective Relations in a 3D Environment. In *GIScience (LNCS)*, M. Raubal, H. J. Miller, A. U. Frank, and M. F. Goodchild (Eds.), Vol. 4197. Springer, 18–32.

S. Bleisch, M. Duckham, A. Galton, P. Laube, and J. Lyon. 2014. Mining candidate causal relationships in movement patterns. *International Journal of Geographical Information Science* 28, 2 (2014), 363–382.

P. Bogaert, N. Van de Weghe, A. G. Cohn, F. Witlox, and P. De Maeyer. 2006. The Qualitative Trajectory Calculus on Networks. In *Spatial Cognition (LNCS)*, T. Barkowsky, M. Knauff, G. Ligozat, and D. R. Montello (Eds.), Vol. 4387. Springer, 20–38.

M. S. Branicky. 1996. General hybrid dynamical systems: Modeling, analysis, and control. In *Hybrid Systems III*, R. Alur, T. A. Henzinger, and E. D. Sontag (Eds.). LNCS, Vol. 1066. Springer, 186–200.

M. Carroll. 1993. Deictic and Intrinsic Orientation in Spatial Descriptions: A Comparison Between English and German. In *Cognition and Culture A Cross-Cultural Approach to Cognitive Psychology*, J. Altarriba (Ed.). Advances in Psychology, Vol. 103. North-Holland, 23–44.

A. Casagrande and C. Piazza. 2012. Model Checking on Hybrid Automata. In *15th Euromicro Conf. on Digital System Design*. 493–500.

Z. Chaochen, D. Guelev, and Z. Naijun. 1999. A Higher-Order Duration Calculus. In *Millenial Perspectives in Computer Science. Proc. of the 1999 Oxford-Microsoft Symposium in Honour of Professor Sir Anthony Hoare, Palgrave*. Springer, 13–15.

Z. Chaochen, C. A. R. Hoare, and A. P. Ravn. 1991. A Calculus of Durations. *Inf. Process. Lett.* 40, 5 (1991), 269–276.

Z. Chaochen, A. P. Ravn, and M. R. Hansen. 1993. An extended duration calculus for hybrid real-time systems. In *Hybrid Systems*, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel (Eds.). Springer, 36–59.

H. Chaudet. 2006. Extending the event calculus for tracking epidemic spread. *Artificial Intelligence in Medicine* 38, 2 (2006), 137–156.

L. Chen, K. Bechkoum, and G. Clapworthy. 2001. A logical approach to high-level agent control. In *Proc. of the 5th Int. Conf. on Autonomous Agents*. ACM, 1–8.

V. R. Chifu, I. Salomie, and E. S. Chifu. 2008. Automatic Web Service Composition Using OWL-S and Fluent Calculus. In *2008 Int. Conf. on Computational Intelligence for Modelling Control Automation*. IEEE, 271–276.

V. R. Chifu, I. Salomie, I. Harsa, and M. Gherga. 2009. Semantic Web Service Composition Method Based on Fluent Calculus. In *11th Int. Symp on Symbolic and Numeric Algorithms for Scientific Computing*. IEEE, 325–332.

N. K. Cicekli and Y. Yildirim. 2000. Formalizing Workflows Using the Event Calculus. In *Proc. of the 11th Int. Conf. on Database and Expert Systems Applications*. Springer, 222–231.

D. J. Clancy and B. Kuipers. 1997. Model Decomposition and Simulation: A Component Based Qualitative Simulation Algorithm. In *AAAI/IAAI*, B. Kuipers and B. L. Webber (Eds.). AAAI Press / The MIT Press, 118–124.

E. Clementini. 2013. Directional relations and frames of reference. *GeoInformatica* 17, 2 (2013), 235–255.

E. Clementini and P. Di Felice. 1995. A Comparison of Methods for Representing Topological Relationships. *Inf. Sci. Appl.* 3, 3 (May 1995), 149–178.

E. Clementini and P. Di Felice. 1997. Approximate topological relations. *Int. J. Approx. Reasoning* 16, 2 (1997), 173–204.

E. Clementini, P. Di Felice, and D. Hernández. 1997. Qualitative Representation of Positional Information. *Artif. Intell.* 95, 2 (1997), 317–356.

A. G. Cohn, B. Bennett, J. M. Gooday, and N. Gotts. 1997. Qualitative Spatial Representation and Reasoning with the Region Connection Calculus. *GeoInformatica* 1, 3 (1997), 275–316.

Z. Cui, A. G. Cohn, and D. A. Randell. 1992. Qualitative Simulation Based on a Logical Formalism of Space and Time. In *Proc. of the 10th National Conf. on Artificial Intelligence*. AAAI Press, 679–684.

J. M. Davoren and A. Nerode. 2000. Logics for hybrid systems. *Proc. IEEE* 88, 7 (2000), 985–1010.

G. De Giacomo, Y. Lespérance, and H. J. Levesque. 1997. Reasoning about Concurrent Execution Prioritized Interrupts, and Exogenous Actions in the Situation Calculus. In *IJCAI*. Morgan Kaufmann, 1221–1226.

G. De Giacomo, Y. Lespérance, and H. J. Levesque. 2000. ConGolog, a concurrent programming language based on the situation calculus. *Artif. Intell.* 121, 1-2 (2000), 109–169.

G. De Giacomo, Y. Lespérance, and A. R. Pearce. 2010. Situation Calculus Based Programs for Representing and Reasoning about Game Structures. In *KR*, F. Lin, U. Sattler, and M. Truszczynski (Eds.). AAAI Press.

H. De Jong. 2004. Qualitative simulation and related approaches for the analysis of dynamic systems. *Knowl. Eng. Rev.* 19, 2 (June 2004), 93–132.

B. De Schutter, W. P. M. H. Heemels, J. Lunze, and C. Prieur. 2009. Survey of modeling, analysis, and control of hybrid systems. In *Handbook of Hybrid Systems Control – Theory, Tools, Applications*, J. Lunze and F. Lamnabhi-Lagarrigue (Eds.). Cambridge University Press, Chapter 2, 31–55.

N. Van de Weghe, P. Bogaert, A. G. Cohn, M. Delafontaine, L. De Temmerman, T. Neutens, P. De Maeyer, and F. Witlox. 2007. How to Handle Incomplete Knowledge Concerning Moving Objects. In *BMI (CEUR Workshop Proceedings)*, B. Gottfried (Ed.), Vol. 296. CEUR-WS.org, 91–101.

N. Van de Weghe, B. Kuijpers, P. Bogaert, and P. De Maeyer. 2005a. A Qualitative Trajectory Calculus and the Composition of Its Relations. In *GeoS (LNCS)*, M. A. Rodríguez, I. F. Cruz, M. J. Egenhofer, and S. Levashkin (Eds.), Vol. 3799. Springer, 60–76.

N. Van de Weghe, G. De Tré, B. Kuijpers, and P. De Maeyer. 2005b. The Double-Cross and the Generalization Concept as a Basis for Representing and Comparing Shapes of Polylines. In *OTM Workshops (LNCS)*, R. Meersman, Z. Tari, P. Herrero, G. Méndez, L. Cavedon, D. Martin, A. Hinze, G. Buchanan, M. S. Pérez, V. Robles, J. Humble, A. Albani, J. L. G. Dietz, H. Panetto, M. Scannapieco, T. A. Halpin, P. Spyns, J. M. Zaha, E. Zimányi, E. Stefanakis, T. S. Dillon, L. Feng, M. Jarrar, J. Lehmann, A. de Moor, E. Duval, and L. Aroyo (Eds.), Vol. 3762. Springer, 1087–1096.

M. Delafontaine, N. Van de Weghe, P. Bogaert, and P. De Maeyer. 2008. Qualitative relations between moving objects in a network changing its topological relations. *Inf. Sci.* 178, 8 (2008), 1997–2006.

M. Denecker and E. Ternovska. 2007. Inductive situation calculus. *Artif. Intell.* 171, 5-6 (2007), 332–360.

C. Drescher and M. Thielscher. 2011. ALPprolog - A new logic programming method for dynamic domains. *Theory and Practice of Logic Programming* 11, 4–5 (2011), 451–468.

K. S. R. Dubba, M. Bhatt, F. Dylla, D. C. Hogg, and A. G. Cohn. 2011. Interleaved Inductive-Abductive Reasoning for Learning Complex Event Models. In *ILP (LNCS)*, S. Muggleton, A. Tamaddoni-Nezhad, and F. A. Lisi (Eds.), Vol. 7207. Springer, 113–129.

D. Dubois, C. A. Welty, and M.-A. Williams (Eds.). 2004. *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004), Whistler, Canada, June 2-5, 2004*. AAAI Press.

F. Dylla and R. Moratz. 2005. Exploiting Qualitative Spatial Neighborhoods in the Situation Calculus. In *Proc of the 4th Int. Conf. on Spatial Cognition: Reasoning, Action, Interaction*. Springer, 304–322.

F. Dylla, T. Mossakowski, T. Schneider, and D. Wolter. 2013. Algebraic Properties of Qualitative Spatio-temporal Calculi, See Tenbrink et al. [2013], 516–536.

F. Dylla and J. O. Wallgrün. 2007a. On Generalizing Orientation Information in OPRA$_m$. In *Proc. of the 29th Annual German Conf. on Artificial Intelligence*. Springer, 274–288.

F. Dylla and J. O. Wallgrün. 2007b. Qualitative Spatial Reasoning with Conceptual Neighborhoods for Agent Control. *Intelligent Robotics Systems* 48, 1 (2007), 55–78.

M. Egenhofer. 2009. A Reference System for Topological Relations between Compound Spatial Objects. In *Proc. of the 3rd Int. Workshop on Semantic and Conceptual Issues in GIS*. Springer, 307–316.

M. Egenhofer. 2010. The Family of Conceptual Neighborhood Graphs for Region-Region Relations. In *Proc. of the 6th Int. Conf. on Geographic Information Science*. Springer, 42–55.

M. Egenhofer and K. K. Al Taha. 1992. Reasoning about Gradual Changes of Topological Relationships. In *From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning*. Springer, 196–219.

M. Egenhofer and J. R. Herring. 1991. *Categorizing Binary Topological Relationships between Regions, Lines and Points in Geographic Databases*. Technical Report 7. National Center for Geographic Information and Analysis.

M. Egenhofer and D. Mark. 1995. Modeling Conceptual Neighborhoods of Topological Line-Region Relations. *J. of Geographical Information Systems* 9, 5 (1995), 555–565.

M. Egenhofer and D. Wilmsen. 2006. Changes in Topological Relations when Splitting and Merging Regions. In *Proc. of the 12th Int. Symposium on Spatial Data Handling*. Springer, 339–352.

M. J. Egenhofer. 1989. A Formal Definition of Binary Topological Relationships. In *FODO (LNCS)*, W. Litwin and H.-J. Schek (Eds.), Vol. 367. Springer, 457–472.

M. J. Egenhofer. 2005. Spherical Topological Relations. *J. Data Semantics III* 3534 (2005), 25–49.

M. J. Egenhofer, E. Clementini, and P. Di Felice. 1994. Topological Relations Between Regions with Holes. *International Journal of Geographical Information Systems* 8, 2 (1994), 129–142.

M. J. Egenhofer and R. D. Franzosa. 1991. Point Set Topological Relations. *International Journal of Geographical Information Systems* 5, 2 (1991), 161–174.

M. J. Egenhofer and J. Sharma. 1993. Topological Relations Between Regions in R^2 and Z^2, See Abel and Ooi [1993], 316–336.

E. A. Emerson. 1990. Handbook of theoretical computer science. Vol. B. MIT Press, Chapter Temporal and modal logic, 995–1072.

M. R. Endsley. 2000. *Theoretical Underpinnings of Situation Awareness: A Critical Review*. Lawrence Erlbaum Associates, 3–33.

C. J. Ewin, A. R. Pearce, and S. Vassos. 2014. Transforming Situation Calculus Action Theories for Optimised Reasoning. In *KR*, C. Baral, G. De Giacomo, and T. Eiter (Eds.). AAAI Press.

A. Ferrein and G. Lakemeyer. 2008. Logic-based robot control in highly dynamic domains. *Robotics and Autonomous Systems* 56, 11 (2008), 980–991.

A. U. Frank. 1996. Qualitative Spatial Reasoning: Cardinal Directions as an Example. *J. of Geographical Information Science* 10, 3 (1996), 269–290.

M. Fränzle. 1996. Synthesizing controllers from Duration Calculus. In *Formal Techniques in Real-Time and Fault-Tolerant Systems*, B. Jonsson and J. Parrow (Eds.). LNCS, Vol. 1135. Springer, 168–187.

M. Fränzle and M. R. Hansen. 2008. Efficient Model Checking for Duration Calculus Based on Branching-Time Approximations. In *Proc. of the 2008 6th IEEE Int. Conf. on Software Engineering and Formal Methods*. IEEE, 63–72.

G. Frehse. 2008. PHAVer: algorithmic verification of hybrid systems past HyTech. *Int. J. Softw. Tools Technol. Transf.* 10, 3 (May 2008), 263–279.

G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. 2011. SpaceEx: Scalable Verification of Hybrid Systems. In *CAV (LNCS)*, G. Gopalakrishnan and S. Qadeer (Eds.), Vol. 6806. Springer, 379–395.

C. Freksa. 1991. Conceptual Neighborhood and Its Role in Temporal and Spatial Reasoning. In *Proc. of the Imacs Int. Workshop on Decision Support Systems and Qualitative Reasoning*, M. G. Singh and L. Trave-Massuyes (Eds.). Elsevier, 181–187.

C. Freksa. 1992a. Temporal Reasoning Based on Semi-Intervals. *Artif. Intell.* 54, 1 (1992), 199–227.

C. Freksa. 1992b. Using orientation information for qualitative spatial reasoning. In *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, A.U. Frank, I. Campari, and U. Formentini (Eds.). LNCS, Vol. 639. Springer, 162–178.

N. Fulton, S. Mitsch, J.-D. Quesel, M. Völp, and A. Platzer. 2015. KeYmaera X: An Axiomatic Tactical Theorem Prover for Hybrid Systems. In *CADE (LNCS)*, A. Felty and A. Middeldorp (Eds.). Springer.

J. Funge. 1999. Representing Knowledge within the Situation Calculus Using Interval-Valued Epistemic Fluents. *Reliable Computing* 5, 1 (1999), 35–61.

A. Galton. 1995. Towards a Qualitative Theory of Movement. In *Proc. of the Int. Conf. on Spatial Information Theory: A Theoretical Basis for GIS*. Springer, 377–396.

A. Galton. 2000. Continuous Motion in Discrete Space. In *Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 26–37.

A. Galton and M. F. Worboys. 2005. Processes and Events in Dynamic Geo-Networks. In *Proc. of the 1st Int. Conf. on Geospatial Semantics*. Springer, 45–99.

J. Gao, Y. Zhao, and T. Yang. 2011. Research on cooperation behaviors of multiple intelligent virtual humans based on fluent calculus. In *2011 Int. Conf. on Computer Science and Service System*. IEEE, 1600–1603.

K. Ghorbal and A. Platzer. 2014. Characterizing Algebraic Invariants by Differential Radical Invariants. In *TACAS*, E. Ábrahám and K. Havelund (Eds.), Vol. 8413. Springer, 279–294.

K. Ghorbal, A. Sogokon, and A. Platzer. 2014. Invariance of Conjunctions of Polynomial Equalities for Algebraic Differential Equations. In *SAS (LNCS)*, M. Müller-Olm and H. Seidl (Eds.), Vol. 8723. Springer, 151–167.

K. Ghorbal, A. Sogokon, and A. Platzer. 2015. A Hierarchy of Proof Rules for Checking Differential Invariance of Algebraic Sets. In *VMCAI (LNCS)*, D. D'Souza, A. Lal, and K. Guldstrand Larsen (Eds.), Vol. 8931. Springer, 431–448.

G. Giacomo, Y. Lespérance, H. J. Levesque, and S. Sardina. 2009. IndiGolog: A High-Level Programming Language for Embedded Reasoning Agents. In *Multi-Agent Programming:*, A. El Fallah Seghrouchni, J. Dix, M. Dastani, and R. H. Bordini (Eds.). Springer, 31–72.

R. Goyal and M. Egenhofer. 2001. Similarity of Cardinal Directions. In *Advances in Spatial and Temporal Databases*, C. Jensen, M. Schneider, B. Seeger, and V. Tsotras (Eds.). LNCS, Vol. 2121. Springer, 36–55.

J. F. Groote and M. A. Reniers. 2001. *Algebraic Process Verification*. Elsevier, Chapter 17, 1151–1208.

H. Grosskreutz and G. Lakemeyer. 2003. ccGolog – A Logical Language Dealing with Continuous Change. *Logic Journal of the IGPL* 11, 2 (2003), 179–221.

D. P. Guelev and D. Van Hung. 2005. On the completeness and decidability of duration calculus with iteration. *Theor. Comput. Sci.* 337, 1-3 (June 2005), 278–304.

M. R. Hansen and D. Hung. 2007. A Theory of Duration Calculus with Application. In *Domain Modeling and the Duration Calculus*, C. W. George, Z. Liu, and J. Woodcock (Eds.). Springer, 119–176.

D. Harel. 1979. *First-Order Dynamic Logic*. LNCS, Vol. 68. Springer.

B. Heath, R. Hill, and F. Ciarallo. 2009. A Survey of Agent-Based Modeling Practices (January 1998 to July 2008). *J. of Artificial Societies and Social Simulation* 12, 4 (2009), 35.

T. A. Henzinger. 1996. The Theory of Hybrid Automata. In *Proc. of the 11th Symposium on Logic in Computer Science*. IEEE, 278–292.

T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. 1997. HyTech: A model checker for hybrid systems. In *Computer Aided Verification*, O. Grumberg (Ed.). LNCS, Vol. 1254. Springer, 460–463.

D. Hernández, E. Clementini, and P. Di Felice. 1995. Qualitative Distances. In *COSIT*. 45–57.

S. Hölldobler and H.-P. Störr. 2000. Solving the Entailment Problem in the Fluent Calculus Using Binary Decision Diagrams. In *Proc. of the 1st Int. Conf. on Computational Logic*. Springer, 747–761.

K. Hornsby and M. Egenhofer. 1997. Qualitative Representation of Change. In *Proc. of the 2nd Int. Conf. on Spatial Information Theory: A Theoretical Basis for GIS*. Springer, 15–33.

Y. Hu and H. J. Levesque. 2009. Planning with Loops: Some New Results. In *Proc. of the ICAPS Workshop on Generalized Planning: Macros, Loops, Domain Control*, C. Fritz, S. McIlraith, S. Srivastava, and S. Zilberstein (Eds.). University of California at Berkeley, 35–42.

J.-B. Jeannin, K. Ghorbal, Y. Kouskoulas, R. Gardner, A. Schmidt, and E. Zawadzki A. Platzer. 2015. A Formally Verified Hybrid System for the Next-generation Airborne Collision Avoidance System. In *TACAS (LNCS)*, C. Baier and C. Tinelli (Eds.), Vol. 9035. Springer, 21–36.

L. Pack Kaelbling and A. Saffiotti (Eds.). 2005. *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*. Professional Book Center.

R. F. Kelly and A. R. Pearce. 2007. Knowledge and observations in the situation calculus. In *AAMAS*, E. H. Durfee, M. Yokoo, M. N. Huhns, and O. Shehory (Eds.). IFAAMAS, 124.

R. F. Kelly and A. R. Pearce. 2010. Property persistence in the situation calculus. *Artif. Intell.* 174, 12-13 (2010), 865–888.

U. Khadim. 2008. *Process Algebras for Hybrid Systems: Comparison and Development*. Ph.D. Dissertation. Eindhoven University of Technology.

M. M. Kokar, C. J. Matheus, and K. Baclawski. 2009. Ontology-based Situation Awareness. *Information Fusion* 10, 1 (2009), 83–98.

S. Konur. 2013. A survey on temporal logics for specifying and verifying real-time systems. *Frontiers of Computer Science* (2013), 1–34.

Y. Kouskoulas, D. Renshaw, A. Platzer, and P. Kazanzides. 2013. Certifying the Safe Design of a Virtual Fixture Control Algorithm for a Surgical Robot. In *Hybrid Systems: Computation and Control, HSCC'13, Philadelphia, PA, USA, April 8-13, 2013*, C. Belta and F. Ivancic (Eds.). ACM.

R. A. Kowalski and F. Sadri. 1994. The Situation Calculus and Event Calculus Compared. In *ILPS*, Maurice Bruynooghe (Ed.). MIT Press, 539–553.

R. A. Kowalski and F. Sadri. 1997. Reconciling the Event Calculus With the Situation Calculus. *J. Log. Program.* 31, 1-3 (1997), 39–58.

R. A. Kowalski and M. J. Sergot. 1986. A Logic-based Calculus of Events. *New Generation Comput.* 4, 1 (1986), 67–95.

B. Kuipers. 1994. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press.

S. Lago-Pereira and L. N. Barros. 2004. High-Level Robot Programming: An Abductive Approach Using Event Calculus. In *Advances in Artificial Intelligence*, A. L. C. Bazzan and S. Labidi (Eds.). LNCS, Vol. 3171. Springer, 73–82.

G. Lakemeyer. 1996. Only Knowing in the Situation Calculus, See Aiello et al. [1996], 14–25.

G. Lakemeyer. 2010. The Situation Calculus: A Case for Modal Logic. *Journal of Logic, Language and Information* 19, 4 (2010), 431–450.

G. Lakemeyer and H. J. Levesque. 1998. AOL: A logic of Acting, Sensing, Knowing, and Only Knowing. In *KR*, Anthony G. Cohn, Lenhart K. Schubert, and Stuart C. Shapiro (Eds.). Morgan Kaufmann, 316–329.

G. Lakemeyer and H. J. Levesque. 2005. Semantics for a useful fragment of the situation calculus, See Kaelbling and Saffiotti [2005], 490–496.

J. H. Lee, J. Renz, and D. Wolter. 2013. StarVars - Effective Reasoning about Relative Directions, See Rossi [2013].

H. J. Levesque. 2005. Planning with loops. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence*. Morgan Kaufmann, 509–515.

H. J. Levesque, R. Reiter, Y. Lesperance, F. Lin, and R. Scherl. 1997. GOLOG: A logic programming language for dynamic domains. *J. of Logic Programming* 31, 1-3 (April 1997), 59–83.

J. A. Lewis, M. P. Dube, and M. J. Egenhofer. 2013. The Topology of Spatial Scenes in $\mathbb{R}2$, See Tenbrink et al. [2013], 495–515.

W. Liu, X. Zhang, S. Li, and M. Ying. 2010. Reasoning about cardinal directions between extended objects. *Artif. Intell.* 174, 12-13 (2010), 951–983.

S. M. Loos, A. Platzer, and L. Nistor. 2011. Adaptive Cruise Control: Hybrid, Distributed, and Now Formally Verified. In *FM (LNCS)*, M. Butler and W. Schulte (Eds.), Vol. 6664. Springer, 42–56.

S. M. Loos, D. Renshaw, and A. Platzer. 2013. Formal Verification of Distributed Aircraft Controllers. In *Hybrid Systems: Computation and Control (part of CPS Week 2013), HSCC'13, Philadelphia, PA, USA, April 8-13, 2013*, C. Belta and F. Ivancic (Eds.). ACM.

C. Lutz, F. Wolter, and M. Zakharyaschev. 2008. Temporal Description Logics: A Survey. In *Proc. of the 15th Int. Symposium on Temporal Representation and Reasoning*. IEEE, 3–14.

K. Mahbub and G. Spanoudakis. 2007. Monitoring WS-Agreements: An Event Calculus-Based Approach. In *Test and Analysis of Web Services*, L. Baresi and E. D. Nitto (Eds.). Springer, 265–306.

O. Maler, Z. Manna, and A. Pnueli. 1991. From Timed to Hybrid Systems. In *Real-Time: Theory in Practice, REX Workshop, Mook, The Netherlands, June 3-7, 1991, Proceedings (LNCS)*, J. W. de Bakker, C. Huizing, W. P. de Roever, and G. Rozenberg (Eds.), Vol. 600. Springer, 447–484.

Y. Martin, I. Narasamdya, and M. Thielscher. 2004. Knowledge of Other Agents and Communicative Actions in the Fluent Calculus, See Dubois et al. [2004], 623–633.

F. J. Martinez, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni. 2011. A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). *Wirel. Commun. Mob. Comput.* 11, 7 (2011), 813–828.

C. J. Matheus, M. M. Kokar, and K. Baclawski. 2003. A Core Ontology for Situation Awareness. In *Proc. of the 6th Int. Conf. on Information Fusion*. IEEE, 545–552.

C. J. Matheus, M. M. Kokar, K. Baclawski, and J. A. Letkowski. 2005a. An Application of Semantic Web Technologies to Situation Awareness. In *4th Int. Semantic Web Conference*. Springer, 944–958.

C. J. Matheus, M. M. Kokar, K. Baclawski, J. A. Letkowski, C. Call, M. L. Hinman, J. J. Salerno, and D. Boulware. 2005b. SAWA: An Assistant for Higher-Level Fusion and Situation Awareness. In *Proc. of the SPIE Conf. on Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, B. V. Dasarathy (Ed.). International Society for Optical Engineering, 75–85.

J. McCarthy. 2002. Actions and Other Events in Situation Calculus. In *KR*, D. Fensel, F. Giunchiglia, D. L. McGuinness, and M.-A. Williams (Eds.). Morgan Kaufmann, 615–628.

R. Meyer, J. Faber, J. Hoenicke, and A. Rybalchenko. 2008. Model checking Duration Calculus: a practical approach. *Form. Asp. Comput.* 20, 4-5 (June 2008), 481–505.

A. Miene, U. Visser, and O. Herzog. 2004. Recognition and Prediction of Motion Situations Based on a Qualitative Motion Description. In *RoboCup 2003: Robot Soccer World Cup VII*. Springer, 77–88.

R. Miller and M. Shanahan. 1996. Reasoning about Discontinuities in the Event Calculus. In *Proc. of the 5th Int. Conf. on Principles of Knowledge Rep. and Reasoning*. Morgan Kaufmann, 63–74.

R. Miller and M. Shanahan. 2002. Some Alternative Formulations of the Event Calculus. In *Computational Logic: Logic Programming and Beyond*, A. C. Kakas and F. Sadri (Eds.). LNCS, Vol. 2408. Springer, 452–490.

J. Misener, M. Barnes, C.-Y. Chan, D. Cody, S. Dickey, R. Goodsell, T. Gordon, Z. W. Kim, T. Kuhn, T. Lian, D. Nelson, C. Nowakowski, K. Nubukawa, A. Sharafsaleh, S. Shladover, J. Spring, J. VanderWerf, W.-B. Zhang, L. Zhang, and K. Zhou. 2010. *Cooperative Intersection Collision Avoidance System (CICAS): Signalized Left Turn Assist and Traffic Signal Adaptation*. Technical Report UCB-ITS-PRR-2010-20. University of California, Berkeley.

S. Mitsch, K. Ghorbal, and A. Platzer. 2013. On Provably Safe Obstacle Avoidance for Autonomous Robotic Ground Vehicles. In *Robotics: Science and Systems*, P. Newman, D. Fox, and D. Hsu (Eds.). TU Berlin.

S. Mitsch, S. M. Loos, and A. Platzer. 2012. Towards formal verification of freeway traffic control. In *ICCPS*, C. Lu (Ed.). IEEE, 171–180.

S. Mitsch and A. Platzer. 2014. ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models. In *Runtime Verification - 5th International Conference, RV 2014, Toronto, ON, Canada, September 22-25, 2014. Proceedings (LNCS)*, B. Bonakdarpour and S. A. Smolka (Eds.), Vol. 8734. Springer, 199–214.

S. Mitsch, J.-D. Quesel, and A. Platzer. 2014a. Refactoring, Refinement, and Reasoning - A Logical Characterization for Hybrid Systems. In *FM 2014: Formal Methods - 19th International Symposium, Singapore, May 12-16, 2014. Proceedings (LNCS)*, C. B. Jones, P. Pihlajasaari, and J. Sun (Eds.), Vol. 8442. Springer, 481–496.

S. Mitsch, J.-D. Quesel, and A. Platzer. 2014b. From Safety to Guilty and from Liveness to Niceness. In *5th Workshop on Formal Methods for Robotics and Automation*.

R. C. Moore. 1977. Reasoning About Knowledge and Action. In *IJCAI*, R. Reddy (Ed.). William Kaufmann, 223–227.

R. Moratz and M. Ragni. 2008. Qualitative spatial reasoning about relative point position. *J. Vis. Lang. Comput.* 19, 1 (2008), 75–98.

R. Moratz, J. Renz, and D. Wolter. 2000. Qualitative Spatial Reasoning about Line Segments. In *ECAI*, Werner Horn (Ed.). IOS Press, 234–238.

R. Moratz and J. O. Wallgrün. 2012. Spatial reasoning with augmented points: Extending cardinal directions with local distances. *J. Spatial Information Science* 5, 1 (2012), 1–30.

T. Mossakowski and R. Moratz. 2012. Qualitative reasoning about relative direction of oriented points. *Artif. Intell.* 180-181 (2012), 34–45.

E. Mueller. 2006a. Event calculus and temporal action logics compared. *AI* 170, 11 (Aug. 2006), 1017–1029.

E. T. Mueller. 2006b. Event calculus and temporal action logics compared. *Artif. Intell.* 170, 11 (2006), 1017–1029.

K. A. Nedas and M. J. Egenhofer. 2004. Splitting Ratios: Metric Details of Topological Line-Line Relations. In *FLAIRS Conference*, Valerie Barr and Zdravko Markov (Eds.). AAAI Press, 795–800.

C. Nikolai and G. Madey. 2009. Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *Artificial Societies and Social Simulation* 12, 2 (2009), 2–39.

E.-R. Olderog and M. Schenke. 1995. Design of Real-Time Systems: Interface between Duration Calculus and Program Specifications. In *Structures in Concurrency Theory*, J. Desel (Ed.). Springer, 32–54.

J. Pinto and R. Reiter. 1995. Reasoning About Time in the Situation Calculus. *Ann. Math. Artif. Intell.* 14, 2-4 (1995), 251–268.

A. Platzer. 2008. Differential Dynamic Logic for Hybrid Systems. *J. Autom. Reasoning* 41, 2 (2008), 143–189.

A. Platzer. 2010a. Differential-algebraic Dynamic Logic for Differential-algebraic Programs. *J. Log. Comput.* 20, 1 (2010), 309–352.

A. Platzer. 2010b. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer.

A. Platzer. 2011. Stochastic Differential Dynamic Logic for Stochastic Hybrid Programs. In *CADE (LNCS)*, N. Bjørner and V. Sofronie-Stokkermans (Eds.), Vol. 6803. Springer, 431–445.

A. Platzer. 2012a. A Complete Axiomatization of Quantified Differential Dynamic Logic for Distributed Hybrid Systems. *Logical Methods in Computer Science* 8, 4 (2012), 1–44.

A. Platzer. 2012b. The Complete Proof Theory of Hybrid Systems. In *Proc. of the 27th Annual ACM/IEEE Symposium on Logic in Computer Science*, N. Dershowitz (Ed.). IEEE, 541–550.

A. Platzer. 2012c. Logics of Dynamical Systems. In *Proc. of the 27th Symposium on Logic in Computer Science*, N. Dershowitz (Ed.). IEEE, 13–24.

A. Platzer. 2013. *A Complete Axiomatization for Differential Game Logic for Hybrid Games*. Technical Report CMU-CS-13-100R. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. Extended in revised version from July 2013.

A. Platzer. 2015. A Uniform Substitution Calculus for Differential Dynamic Logic. In *CADE (LNCS)*, A. Felty and A. Middeldorp (Eds.). Springer.

A. Platzer and E. M. Clarke. 2009. Formal Verification of Curved Flight Collision Avoidance Maneuvers: A Case Study. In *Proc. of the 16th Int. Symp. on Formal Methods (LNCS)*, A. Cavalcanti and D. Dams (Eds.), Vol. 5850. Springer, 547–562.

A. Platzer and J.-D. Quesel. 2008. KeYmaera: A Hybrid Theorem Prover for Hybrid Systems.. In *Proc. of the 4th Int. Joint Conf. on Automated Reasoning*. Springer, 171–178.

A. Pnueli. 1977. The Temporal Logic of Programs. In *Proc. of the 18th Annual Symposium on Foundations of Computer Science*. IEEE, 46–57.

V. R. Pratt. 1976. Semantical Considerations on Floyd-Hoare Logic. In *17th Annual Symposium on Foundations of Computer Science, Houston, Texas, USA, 25-27 October 1976*. IEEE Computer Society, 109–121.

M. Ragni and S. Wölfl. 2005. Temporalizing Spatial Calculi: On Generalized Neighborhood Graphs. In *Proc. of the 28th Annual German Conf. on Artificial Intelligence*. Springer, 64–78.

M. Ragni and S. Wölfl. 2006. Temporalizing Cardinal Directions: From Constraint Satisfaction to Planning. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Rep. and Reasoning*. AAAI, 472–480.

M. Ragni and S. Wölfl. 2008. Reasoning about Topological and Positional Information in Dynamic Settings. In *Proc. of the 21st Int. Florida Artificial Intelligence Research Society Conf.* AAAI, 606–611.

D. A. Randell, Z. Cui, and A. G. Cohn. 1992. A Spatial Logic based on Regions and Connection. In *Proc. of the 3rd Int. Conf. on Knowledge Rep. and Reasoning*. Morgan Kaufmann, 165–176.

D. A. Randell, G. Landini, and A. Galton. 2013. Discrete Mereotopology for Spatial Reasoning in Automated Histological Image Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 3 (2013), 568–581.

D. A. Randell and M. Witkowski. 2004. Tracking Regions using Conceptual Neighbourhoods. In *Proc. of the Workshop on Spatial and Temporal Reasoning*. IOS Press, 63–71.

R. Reis, M. Egenhofer, and J. Matos. 2008. Conceptual Neighborhoods of Topological Relations between Lines. In *13th Int. Symp. on Spatial Data Handling*, A. Ruas and C. Gold (Eds.). Springer, 557–574.

R. Reiter. 1996. Natural Actions, Concurrency and Continuous Time in the Situation Calculus, See Aiello et al. [1996], 2–13.

R. Reiter. 2001. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press.

J. Renz and D. Mitra. 2004. Qualitative Direction Calculi with Arbitrary Granularity. In *PRICAI 2004: Trends in Artificial Intelligence*, C. Zhang, H. W. Guesgen, and W.-K. Yeap (Eds.). LNCS, Vol. 3157. Springer, 65–74.

J. Renz and B. Nebel. 2007. Qualitative Spatial Reasoning Using Constraint Calculi. In *Handbook of Spatial Logics*, M. Aiello, I. Pratt-Hartmann, and J. van Benthem (Eds.). Springer, 161–215.

F. Rossi (Ed.). 2013. *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. IJCAI/AAAI.

A. Schäfer. 2006. *Specification and Verification of Mobile Real-Time Systems*. Ph.D. Dissertation. Universität Oldenburg.

R. B. Scherl. 2003. Reasoning about the Interaction of Knowledge, Time and Concurrent Actions in the Situation Calculus. In *IJCAI*, G. Gottlob and T. Walsh (Eds.). Morgan Kaufmann, 1091–1098.

R. B. Scherl and H. J. Levesque. 1993. The Frame Problem and Knowledge-Producing Actions. In *AAAI*, R. Fikes and W. G. Lehnert (Eds.). AAAI Press / The MIT Press, 689–695.

R. B. Scherl and H. J. Levesque. 2003. Knowledge, action, and the frame problem. *Artif. Intell.* 144, 1-2 (2003), 1–39.

S. Schiffel and M. Thielscher. 2006. Reconciling Situation Calculus and Fluent Calculus. In *AAAI*. AAAI Press, 287–292.

S. Schiffel and M. Thielscher. 2007. Fluxplayer: A Successful General Game Player. In *Proc. of the 22nd AAAI Conf. on Artificial Intelligence*. AAAI Press, 1191–1196.

S. Schiffer, A. Ferrein, and G. Lakemeyer. 2012. Reasoning with Qualitative Positional Information for Domestic Domains in the Situation Calculus. *Journal of Intelligent and Robotic Systems* 66, 1-2 (2012), 273–300.

C. Schlieder. 1995. Reasoning About Ordering. In *COSIT*. 341–349.

C. Schultz and M. Bhatt. 2012. Towards a Declarative Spatial Reasoning System. In *ECAI (Frontiers in Artificial Intelligence and Applications)*, L. De Raedt, C. Bessière, D. Dubois, P. Doherty, P. Frasconi, F. Heintz, and P. J. F. Lucas (Eds.), Vol. 242. IOS Press, 925–926.

A. Scivos and B. Nebel. 2001. Double-Crossing: Decidability and Computational Complexity of a Qualitative Calculus for Navigation. In *COSIT (LNCS)*, D. R. Montello (Ed.), Vol. 2205. Springer, 431–446.

A. Serenko and B. Detlor. 2003. Agent toolkit satisfaction and use in higher education. *J. of Computing in Higher Education* 15 (2003), 65–88. Issue 1.

M. Shanahan. 1996. Robotics and the Common Sense Informatic Situation. In *Proc. of the 12th European Conf. on Artificial Intelligence*. John Wiley and Sons, 684–688.

M. Shanahan. 1997. Event Calculus Planning Revisited. In *Proc. of the 4th European Conf. on Planning (LNCS)*, S. Steel and R. Alami (Eds.), Vol. 1348. Springer, 390–402.

M. Shanahan. 1999. The Ramification Problem in the Event Calculus. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence*. Morgan Kaufmann, 140–146.

M. Shanahan. 2000. An abductive event calculus planner. *J. of Logic Programming* 44, 1-3 (2000), 207–240.

M. Shanahan and M. Witkowski. 2004. Event Calculus Planning Through Satisfiability. *J. Log. and Comput.* 14, 5 (Oct. 2004), 731–745.

S. Skiadopoulos and M. Koubarakis. 2001. Composing Cardinal Direction Relations. In *Advances in Spatial and Temporal Databases*, C. S. Jensen, M. Schneider, B. Seeger, and V. J. Tsotras (Eds.). LNCS, Vol. 2121. Springer, 299–317.

H.-P. Störr. 2001. Planning in the Fluent Calculus Using Binary Decision Diagrams. *AI Magazine* 22, 3 (2001), 103–106.

A. Sulistio, C. S. Yeo, and R. Buyya. 2004. A taxonomy of computer-based simulations and its mapping to parallel and distributed systems simulation tools. *Softw. Pract. Exper.* 34, 7 (2004), 653–673.

T. Tenbrink, J. G. Stell, A. Galton, and Z. Wood (Eds.). 2013. *Spatial Information Theory - 11th International Conference, COSIT 2013, Scarborough, UK, September 2-6, 2013. Proceedings*. LNCS, Vol. 8116. Springer.

M. Thielscher. 1999. From Situation Calculus to Fluent Calculus: State Update Axioms as a Solution to the Inferential Frame Problem. *Artificial Intelligence* 111, 1-2 (1999), 277–299.

M. Thielscher. 2005. FLUX: A logic programming method for reasoning agents. *Theory Pract. Log. Program.* 5, 4-5 (July 2005), 533–565.

M. Vasardani and M. J. Egenhofer. 2009. Comparing Relations with a Multi-holed Region. In *COSIT (LNCS)*, K. S. Hornsby, C. Claramunt, M. Denis, and G. Ligozat (Eds.), Vol. 5756. Springer, 159–176.

M. F. Worboys. 2005. Event-oriented Approaches to Geographic Phenomena. *Geographical Information Science* 19, 1 (2005), 1–28.

S. Zimmerbaum and R. B. Scherl. 2000. Sensing Actions, Time, and Concurrency in the Situation Calculus. In *ATAL (LNCS)*, C. Castelfranchi and Y. Lespérance (Eds.), Vol. 1986. Springer, 31–45.

K. Zimmermann and C. Freksa. 1996. Qualitative Spatial Reasoning Using Orientation, Distance, and Path Knowledge. *Appl. Intell.* 6, 1 (1996), 49–58.