Safe and Lean Location-based Construction Scheduling

Beidi Li [a], Carl Schultz [a], Jürgen Melzner [b], Olga Golovina [a], and Jochen Teizer [a]

[a] Department of Engineering, Aarhus University, Denmark
[b] Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt, Germany
E-mail: beidi.li@eng.au.dk; cschultz@eng.au.dk; teizer@eng.au.dk; juergen.melzner@fhws.de

Abstract –
Based on case studies within the construction industry, the application of location-based construction scheduling and utilizing software-based rule checking has delivered promising research results. We first explain a path on how the lessons from existing work practices can be used in digitalizing construction processes. This includes the objective of describing how the digital transformation of construction drawings and work break down structures lead to safe and lean work environments that crews – according to occupational laws and regulations – should face.

To achieve this objective, we present our ongoing work towards a unifying formal (logic-based) domain model that consists of: (1) a semantically rich ontology of construction schedules, work breakdown structures, and safety concepts; (2) rules for undertaking construction activities that avoid unsafe and wasteful situations.

This paper displays the case study of a newly built fire station for validation of the developed prototype. This experience illustrates that safe and lean allocation of work crews can be planned before construction starts. The outlook presents it's potential in future applications in the construction industry, e.g. resource allocation, as well as in research, e.g. automated work progress tracking by comparing actual vs. planned data.

Keywords –
Building Information Modeling (BIM); construction safety and health; fall protection; lean construction scheduling; prevention through design and planning; resource allocation; rule checking

1 Introduction

All project stakeholders that facilitate design, planning, construction and operation play a vital role in achieving project objectives for cost, schedule and quality. However, few recognize that design and planning can play a critical role for the safety, health and well-being of construction workers, maintenance staff or users during an entire project life-cycle. Although significant research has been undertaken in occupational construction safety, health and well-being, human-assisted software tools for detection and prevention of hazards embedded in construction schedules hardly exist in practice.

Example: Figure 1 illustrates a building information model (BIM) of a (real) building under construction. The roof panels were planned to be installed in sequence. One particularly ubiquitous hazard is that of a *fall* hazard: falls on construction sites account for approximately one third of all fatalities and numerous more severe accidents leading tragically to loss of life, serious and minor injuries [1].

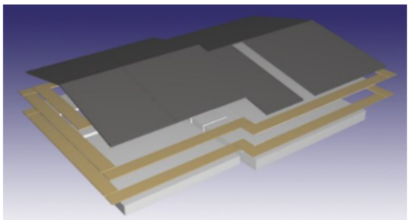


Figure 1. 4D BIM of roof panels under construction.

The edge of a working platform on which workers are occupied with a task that has a drop of more than approximately 2m is widely deemed to be categorized as a hazardous leading edge in many of the world's construction safety codes. In such situations, a fall protection system may prevent a worker from colliding with the ground, structure, or any other obstacle during a free fall and limit the impact force on the body of the worker during fall arrest.

Unfortunately, few projects utilizing BIM model fall protection (e.g., guardrails, safety nets, and covers for holes or openings on roofs) and personal protective equipment (PPE) (e.g., harnesses, lanyards, and temporary anchor points) are not part of the standard object libraries in commercial BIM software. Furthermore, a user-friendly software component to plan the use of PPE that is easy to use, fast, and perhaps can consider work progress, in brief here called a safety analysis system that assesses safety code compliance over the project schedule, does not exist.

Moreover, leading edges can be further distinguished based on the geometry of the work platform (i.e., in this example the steepness of the roof's slope, and the location of the edge relative to the slope), and tasks or next activities in a construction schedule.

As shown in Figure 2, in time step t_1 the first roof panel is installed, creating four leading edges (one edge on each side of the panel). In the very next time step t_2 the next roof panel is installed, causing the rightmost leading edges from t_1 to disappear (a transient edge). In contrast, the upper leading edge in t_1 will remain until the very last roof panel is installed on the far side of the roof (a persistent edge), and the leftmost edge and lower edge are permanent, and will still exist even when the building is completed (permanent edges).

Edges are further distinguished based on their position on the slope: the upper edge being at the top of the slope is dealt with differently than the lower edge at the bottom of the slope where preventative measures employ safety nets for catching workers, material and/or tools that may accidentally slide down and fall off the roof.

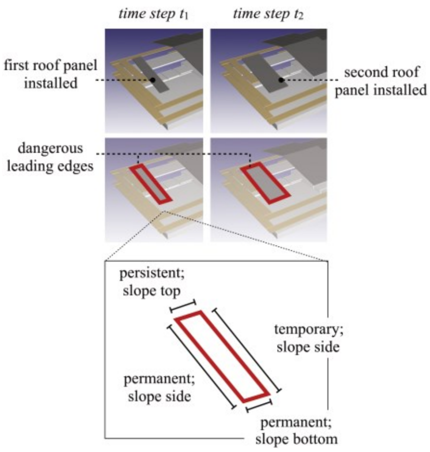


Figure 2. Part way through a construction plan. Roof panels are being installed sequentially in time steps t_1 and t_2. The dangerous leading edges that exist in each partially constructed state are distinguished based on the geometry of the roof slope, and the temporality of the leading edge based on the next construction task in the plan.

In general, measures to prevent fall hazards on roofs include installing fall protection and workers using PPE, or combinations of both. Depending on the type of work, the leading edge is always protected using a fall protection system. Workers without sufficient instruction or training and the right supervision often disregard wearing a personal fall arrest system [2].

Importantly, 4D building information models (4D BIM) are often *incomplete* in that the designer has omitted certain key pieces of information, e.g. the particular order in which roof panels will be installed. It makes it difficult to assess whether a safety code is complied with or not at a particular time step. This motivates the role of **default reasoning** in safety analysis, so that we might assert by default certain details that are missing from an incomplete BIM, and **hypothetical reasoning** that uses what-if scenarios to abduct missing information that would result in a particular situation, e.g. *"Suppose that the roof has to be completed in 5 workdays, can we fill in the blanks in the construction schedule in a way that meet with the availability of the crew and safety and health resources?"* Thus, we require:

- a semantically rich domain model of different features in a 4D BIM (such as leading edges and the refining semantic categories);
- a knowledge base of formal rules that can take a 4D BIM, analyze it, and augment it with these new concepts, injected as new, special kinds of *objects* in the model; and
- a reasoning engine that can take hypothetical statements about construction plans and missing details in the BIM and identify and mitigate resulting safety hazards using adequate methods that are consistent with safety codes and typical construction practices.

2 Related work

Work on roofs is highly dangerous and proper precautions are needed to control the risk. The main causes of death and injury are falling from roof edges or openings, through fragile roofs and through skylights. Many accidents could be avoided if the most suitable equipment was used and those doing the work were given adequate information, instruction, training and supervision. Roof work requires careful planning, particularly where work progresses along the roof. Sloping roofs require scaffolding to prevent people or materials falling from the edge. Another issue is that the small size and economic pressure of roofing companies often does not allow the execution of such best practices.

While Prevention through Design (PtD) concepts have been practiced for many years [3], most of the existing risk mitigation approaches are done manually, and are thus prone to error or not performed at the right time [4]. Several other key reasons contribute to such unacceptable practices: (a) the disconnect in a fragmented construction industry does not allow owners, architects, engineers, contractors and subcontractors to exchange their respective competent knowledge within the disciplines via an open, shareable platform, thus causing poor designs and unsafe execution and (b) the process of preventing hazards starts too late, often in the construction planning phase only, and involves safety and health experts who have to manage multiple projects at the same time.

More recently, research on Job Hazard Analysis (JHA) [5] and safety rule checking [6-8] that can automatically detect and resolve known hazards embedded in individual work activates have been introduced.

However, there is still a wide gap in standardization of safety concepts and software tools and a lack of strong requests from project owners and contractors to demand such solutions. An extensible, intuitive to apply, integrated suite of safety analysis software tools for construction is currently still missing. In part, the reasons are the complex, dynamic nature of construction projects and the multifaceted roles of its stakeholders. Aligning design intent with construction schedules and allocating resources (labor, material, equipment) is a demanding task. *Lean* in the field of construction safety refers to designing safe workplace, for example, using virtual construction models that attempt to bring construction process and safety product information to the job face [10]. Although research has shown that the creation of automated construction schedules is possible based on a priori knowledge about processes of activities and historical crew data, process related safety information has been left out so far for the majority of potential applications [11].

3 Methods

3.1 SafeConDM: an Ontology of Construction Safety

An example illustrates the German construction safety regulation for fall prevention (Figure 3) [12]. Similar to other countries, the code states that a guardrail must be installed at a leading edge if a worker could fall more than 2m, or a covering must be applied if the drop in a hole on the work platform is greater than 9 m^2.

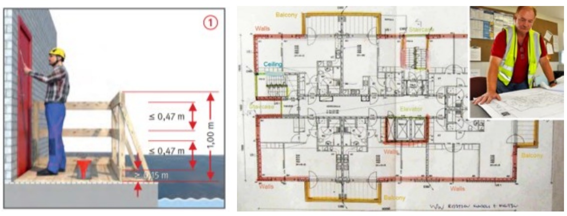


Figure 3. Examples: Safety regulation for fall prevention [12] (left) and manual hazard identification and mitigation (right).

To detect and prevent, for example, a fall-from-height hazard and apply a protective guardrail system, in an ideal case, a designer would design-out the hazard (so it does not appear during construction or later in maintenance). In reality, a safety engineer manually identifies the hazardous locations on a paper-based drawing (e.g., colors in Figure 3 indicate types and locations where protective equipment needs to be installed) or substitutes unsafe construction methods with a safer method (e.g., instead of workers using ladders that can tilt, workers should apply a scissor lift platform).

We aim to develop software tools that can automatically assess such codes on a given BIM of a construction site. BIMs are an object-oriented formal representation of buildings, including classes such as door, wall, or slab. Furthermore, in the field of *construction planning*, 4D BIM is used to model how a BIM is planned to be erected in a series of discrete time steps, i.e. a 4D BIM is equivalent to a sequence of partially constructed BIMs that represent the building under construction.

Our approach has been developed based on previous research in ontological and logic-based approaches to Construction Safety including [5,6]. To illustrate this, we integrate our approach into a broader existing ontological framework for construction safety. Figure 4 illustrates the *construction safety ontology* by Zhang et al. [5] extended with new (abstract) classes: spatial artefact and hazard space [13]. The authors hereby distinguish the following three modelling layers:

1) *Construction Product Model:* building products and relations, such as doors, walls, stories, slabs;
2) *Construction Process Model:* the construction plan including resources (equipment, materials, labor);
3) *Construction Safety Model:* construction safety knowledge (potential hazards, regulations, mitigating steps).

We define pertinent spatial artefacts [14] that capture semantic information about regions of empty space based on construction site activities, and human perception and behavior (movement, visibility, falling spaces, activity, etc.). Similarly, we model hazards as spatial artefacts whose existence and (geometric) definition is often a simple expression involving topological relations and geometric operations between regions (intersection, union, offset etc.), i.e. the *algorithm* for hazard detection is often as simple as clash detection. Spatial artefacts are modeled on the same ontological level as any other object in the product model, i.e. they inherit from the abstract class Product.

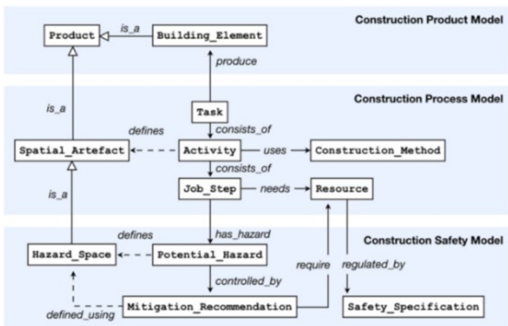


Figure 4. Construction Safety Ontology from [5] extended with spatial artefacts to create SafeConDM [13].

3.2 The shape of meaningful *empty spaces* in construction safety

Consider the region of empty space around an object such as a fuse box or valve; this region is meaningful because a person must be located in that region to perform a particular act (e.g., operate on the fuse box). The geometry of this *functional space* region depends on properties of the person (consider electrician, mechanic, etc.), the task, and the object. Agents (e.g., workers and vehicles) have a movement space which are the regions in which they can move (travel) within. Excavators and other heavy equipment have an *operational space* required for rotating and depositing dug up material. People and sensors have *range spaces* (which can be further refined into: *visibility space, hearing space, reach space*), and so on. These are examples of spatial artefacts, a modeling approach that was initially developed for human centered architectural analysis [14-18].

The idea is that human behavior and experience is formally modelled as regions of empty space in which those behaviors and perceptual experiences take place. By doing this, behavior and experience (such as where a person needs to stand to see a particular object, i.e. *visibility space*) can be *represented* as "objects" in a BIM, on the same ontological level as other building objects such as doors, walls, slabs, and so on. Moreover, they can be *reasoned about* in a similar way to other BIM objects that instead have a material extension, e.g. clash detection can now be used to reason about whether a worker is in danger of a heavy vehicle strike by finding the intersection between the worker's *movement space*, the o*perational space* of the vehicle, and the *blind spots* of the vehicle operator seated in the cab. Concretely, in a BIM such as IFC, spatial artefacts form an abstract class that is a subclass of *IfcSpace*.

Example. Consider the previously discussed natural language code about a specific fall hazard: *"A platform that has a leading edge to a drop of more than 2m must be secured by a guardrail."*

We define a new spatial artefact called *Fall Space*, parametrically defined as: the region in which a person will fall by at least height `DANGEROUS_DISTANCE`, i.e. in the German code example the parameter is set to 2m. The dangerous platform edges can now be precisely, formally defined as: "where a *Movement space* horizontally meets (touches) a *Fall Space*".

There are a number of desirable properties of using spatial artefacts from a knowledge engineering perspective: the formalization is (a) very faithful to the original natural language code (semantics only, without any additional clauses for speeding up rule checking which would clutter the formalization and obscure the intended meaning), (b) easy to understand and verify by other planners (transparent); (c) directly applies to different contexts without changing the declarative statement that formalizes the code (portable), i.e. the geometry of Fall Space is customized according to the project and context, whereas the concept dangerous edge as defined above does not need to change. Importantly, this provides a uniform approach for modelling a large range of human-centered concepts (movement, visibility, performing tasks etc.) that can seamlessly be integrated within a BIM, and are effective "building blocks" for formalizing a broad range of hazards in a clear and transparent way.

In Tables 1 and 2 we list the spatial artefacts that we use to define fall hazards. We develop two new classes of spatial artefacts: *Falling spaces* and *Hazard spaces*. We ground the geometry of the spatial artefacts in our models based on the specific context of construction. We encode rules about hazards as the spatial definition of specific (subclasses of) hazard spaces.

Table 1. Construction site spatial artefacts

Spatial Artefact	**Description**
Movement space	Regions in which an agent (e.g., construction worker, manager, and visitor) can travel.
Movement corridor	Specific pathways along which a group of agents (e.g. crowds) is moving.
Functional space	Region in which an agent must be located to perform a given function or use a given object.
Work area	Area where an agent is occupied with a given task (e.g. electrician working on a fuse box).
Range space	Regions carrying information about how an object can be detected by an agent.
Visibility space	Region in which there is an unobstructed line of sight to a given object.
Blind spot	Region to which a vehicle operator has obstructed line of sight.
Fall space	Region in which an object or agent will fall by a dangerous distance.
Broad fall space	Region through which an agent can (easily) fall.
Narrow fall space	Region that is too narrowly shaped for an agent to (easily) fall through, but through which equipment and material could fall, or in which an agent's ankle could get stuck or sprain.
Operational space	Region that an object or vehicle may occupy to perform a given function.

3.3 Reasoning about hazards in construction

4D BIM introduces time to model a (possibly incomplete) construction plan. We encode this in Answer Set Programming (ASP) using two new predicates. Each element can optionally be assigned to a symbolic time step *construct/2*. The set of time steps form a partial order through an intransitive relation *next/2*: given time steps t_i, t_j then the interpretation of *next*(t_i, t_j) is that t_j occurs directly after t_j such that there does not exist time step t_k where *next*(t_i, t_k) and *next*(t_k, t_j). The temporal relation *before/2* between time steps is the transitive closure of *next/2*. Finally, 4D BIMs express temporal dependencies between elements: *dependency/2* between two BIM elements *A*, *B* means that element *A* must be constructed before *B*.

The following code snippet describes a series of slabs s_1, s_2, s_3 optionally assigned a time step, and their temporal dependencies. ASP derives the partial order *before/2* and the total order *next/2* of time steps.

Table 2. Construction safety hazards defined as spatial artefacts.

Subclass of hazard spaces	Hazard category	Description	Spatio-temporal definition
Fall hazard space	Slips, trips and falls	A person in these regions is at risk of falling from a dangerous height.	Intersection of movement spaces and broad fall spaces.
Sprain ankle hazard space	Slips, trips and falls	A person in these regions is at risk of twisting their ankle by walking into a small hole.	Intersection of movement spaces and narrow fall spaces.
Falling object hazard space	Struck by	A person in these regions may get hit by a falling object (e.g. tools, materials).	Subset of movement spaces that is directly below a narrow fall space (e.g. their vertical projection overlap).
Falling material corridor	Struck by	A person in these regions is at risk of being hit by an object falling from an active work area directly above.	Subset of movement spaces that is directly below the intersection of narrow fall space and work area.
Travelling vehicle strike hazard space	Struck by	A person in these regions is at risk of being hit by a moving vehicle.	Intersection of vehicle movement space (or corridor) and worker movement space (or corridor).
Operating vehicle strike hazard space	Struck by	A person located in these regions is at risk of being struck by a heavy vehicle in operation.	Intersection of vehicle operational space and worker movement space. The presence of blind spots further increases the risk.
Electrocution hazard space	Electrocution	A person located in these regions is at risk of electrocution.	Functional space of fuse box.

```
construct(s1, 1).
construct(s2, 3).
construct(s3, 4).
timestep(1..5).
before(T1, T2):-
   timestep(T1), timestep(T2),
   T1 < T2.
-next(T1, T3):-
   before(T1, T2), before(T2, T3).
next(T1, T2):-
   before(T1, T2), not -next(T1, T2).
dependency(S1, S2) :-
   construct(S1, T1), construct(S2, T2),
   before(T1, T2).
```

3.4 Evaluating consistency

Given a declarative formal encoding of construction safety codes, we can evaluate the consistency of such statements on a given BIM using off-the-shelf solvers that have spatial reasoning support; we opt for using a logic programming paradigm where the knowledge base consists of Horn clause rules of the form: $h \leftarrow b_1,\ldots, b_n$, where proposition h is true (the rule head) if propositions $b_1,\ldots, b_n$ are all true (the rule body). Horn clauses strike a balance between being sufficiently expressive to capture logical IF-THEN relationships between symbolic terms, while still being computable (unlike full first-order logic). We specifically use ASP, a logic- programming paradigm developed within the artificial intelligence community, that supports both deduction and other forms of non-monotonic reasoning (including default reasoning and hypothetical reasoning) and is computationally efficient. Similar to Prolog, ASP has a knowledge base of facts and rules of the form: "*Head :- Body.*" meaning that if the Body is true, then the Head must also be true. Rules with no Head are ASP integrity constraints, written as "*:- Body*." meaning that the Body must not be true (i.e. as a logical expression: Body implies False). Head and Body expressions consist of literals, representing propositions that can be either True or False, and ASP reasoning engines are specifically designed to rapidly find combinations of deduced facts that are consistent with all given domain rules (referred to as models or answer sets).

We have extended the base language of ASP beyond propositions so that a set of consistent facts must also be spatially consistent, e.g. a 2D point can never be both inside, and outside, of a given polygon [19]. We use our extension of ASP that also supports spatial reasoning, called ASPMT(QS) [19], by encoding a building information model and derived artefacts as ASP facts, encoding the inference of hazards and responses as ASP rules and constraints, and implementing safety checking via ASP's answer set search. We have implemented ASPMT(QS) based on clingo [20], a complete ASP system composed of a grounder (gringo) and a solver (clasp).

Example. The following ASPMT(QS) rule states that, for all movement spaces that meet flush (touch horizontally) with a fall space in the same time step, then deduce a fall hazard space object. In this example, fall hazard spaces are modeled as the intersection of movement spaces and fall spaces, offset by a threshold of 200mm.

```
fall_hazard_space(H, Time) :-
   timestep(Time),
   movement_space(M, Time),
   fall_space(F, Time),
   topology(externally_connected, M, F),
   H_ = intersection(M,F),
   H = buffer(H_, 200).
```

We then derive the semantic refinements of fall hazard spaces based on their temporal duration in the construction plan. Firstly, a part of a fall hazard space (H) at timestep T is defined to be *permanent* (Hp) if it is still there in the final timestep (Te), i.e. it is the intersection between fall hazard space H at time T, and a fall hazard space He in the final timestep Te:

```
permanent_fall_hazard_space(Hp, T) :-
   timestep(T),
   timestep(Te), not next(Te, _),
   fall_hazard_space(H, T),
   fall_hazard_space(He, Te),
   Hp = intersection(H, He).
```

Conversely, a part of a fall hazard space (H) at time step T is temporary (Ht) if it does *not* exist in the final time step (Te), which is computed by subtracting the final fall hazard spaces (He) from H:

```
temporary_fall_hazard_space(Ht, T) :-
   timestep(T),
   timestep(Te), not next(Te, _),
   fall_hazard_space(H, T),
   fall_hazard_space(He, Te),
   Ht = difference(H, He).
```

A temporary fall space is *transient* at T if it disappears in the very next time step Ti:

```
transient_fall_hazard_space(Ht, T) :-
   timestep(T), next(T, Ti),
   temporary_fall_hazard_space(H, T),
   temporary_fall_hazard_space(Hi, Ti),
   Ht = difference(H, Hi).
```

A temporary fall space is *persistent* at T if it still exists in the very next timestep Ti:

```
persistent_fall_hazard_space(Hp, T) :-
   timestep(T), next(T, Ti),
   temporary_fall_hazard_space(H, T),
   temporary_fall_hazard_space(H_, Ti),
   Hp = intersection(H, Hi).
```

Similarly, movement spaces are created as the volume 2m directly on top of slabs, subtracted by walls, columns and other movement obstacles. Fall spaces are the volume of space between the top surface of each object, and the next surface directly above (or the "sky") with the lower 2m subtracted. For this first prototype we simplified the calculation of movement spaces as the top surface of slabs subtracted by movement obstacles (columns and walls with voids where windows and doors will be placed), and we simplified fall spaces by taking a 2D bounding box of the site on each building floor and subtracting the slabs on that floor.

Note: The final time step Te is identified as the time step that does **not** have any *next* time step, i.e. "not next(Te,_)". In ASP underscore refers to an "anonymous" variable that does not need to be named. In the above rules, we assimilate object IDs to their geometric representation in ASP's internal geometry database. Moreover, topology relations and Boolean operations on polygons (intersection, difference, buffer) are evaluated using external Python libraries (Polygon, PyClipper) and non-linear real arithmetic solver *z3*.

4 Case study and results

A fire station is a structure or other area for storing firefighting apparatus such as fire engines and related vehicles, personal protective equipment, fire hoses and other specialized equipment, extended the nature of fire emergencies. Fire stations around the world also provide an important role in training volunteering or professional fire fighters or search and rescue personnel on site and educating the public regarding fire and safety (Figure 5). Most fire or Emergency Medical System (EMS) stations are municipally owned and usually require public bidding. In rural areas, many firefighters contribute labor time, increasing the potential risk on such projects.

Construction budgets and schedules respectively vary by the project. Due to the large number of such buildings, preference is given to systems design and functionality (which can be repeated once available).

The fire station model in this case study consists of a building with several levels. We focus on the roof aggregate consisting of 24 panels and 1 chimney opening. The chimney is added after the roof panels are installed. Supposing the roof is installed in the anti-clockwise order from the lower right corner, we assign each panel with a time step ranging from 1 to 24 (Figure 5). ASP then identifies fall hazard spaces (permanent, transient and persistent) at each time step. Moreover, permanent and persistent fall hazards are mitigated depending on the location of fall hazard spaces with respect to the slope, e.g. leading edges on the bottom of the slope require safety nets to capture falling objects, leading edges on the top or the side of the slope require guardrails, small openings on the slope require a cover (Figure 5). Table 3 shows fall hazard spaces and mitigation measures at 3 representative time steps.

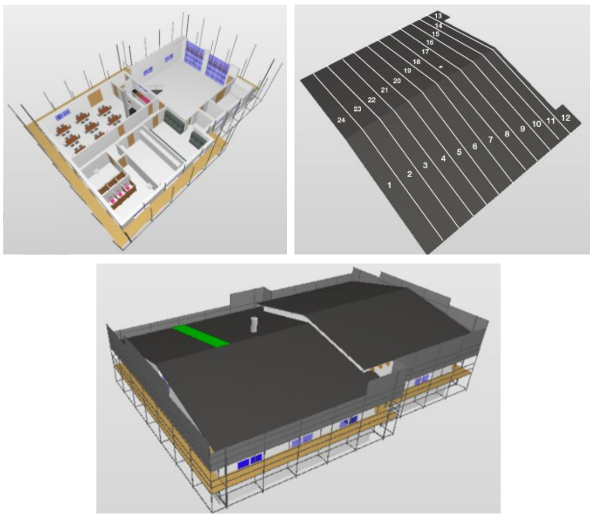


Figure 5. Main functions of a fire station on first floor (top left), order of roof panel installation (top right),3D view of the fire station model incl. protective safety nets on scaffolding with a single highlighted roof panel.

Table 3. Identified fall hazard spaces and proposed mitigation measures at t = 5, 19, and 24. Permanent (red), transient (green), and persistent (purple) fall hazards; Non-transient fall hazards are mitigated using a combination of safety nets (orange), guardrails (blue), and coverings (pink).

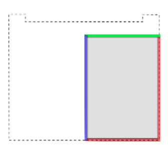
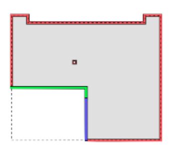
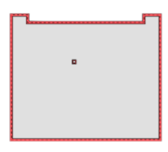
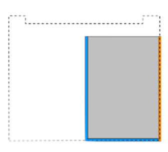
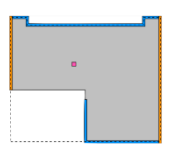
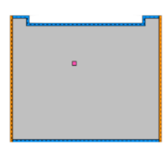
	t = 5	t = 19	t = 24
Fall hazard spaces			
Miti-gation mea-sure			

Figure 6 shows the quantity of safety protection materials needed at each time step. While these quantities were automatically generated, it shows a nearly linear demand for safety nets. Midway the project (t = 13) the peak of demand for guardrail is reached (panels on the other side of the roof allow removal of earlier installations). The demand for coverings becomes a non-null constant at t = 17 when the roof panel with one sole opening (for the later installation of the chimney) is installed and requires protection.

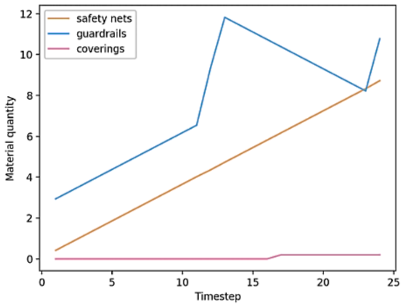


Figure 6. Requirement of safety protection equipment as roof panels are installed successively.

Providing a project manager or a safety engineer with such visual and quantitative information can impact decision making. Noteworthy examples are: (a) understand the location of fall risks associated to specific tasks in a construction schedule and (b) ordering the right quantities of protective equipment when needed. In a further step, if done continuously throughout a project, responsible personnel can seek forecasts of potential demands of (fall) protection resources and align with proper construction methods. On this particular project, due to the roof covering the entire and relatively small building area, resource leveling would not offer much potential savings [21]. However, depending on the amount of time needed to install one roof panel, an alternative fall arrest system (e.g., lanyard and energy absorber instead of guardrails, safety nets, and a hole cover) could be employed to protect the workforce who is installing the roof panels [22].

In this paper, we natively integrate an internal geometry database within ASPMT(QS) to manage large amounts of complex geometric data. To do this, we generate the polyhedral mesh representation of BIM objects via our modified version of IfcConvert (IfcConvert+). ASPMT(QS) then retrieves object geometries and deduces spatial artefacts using previously defined rules. Table 4 presents model statistics and ASPMT(QS) runtime. Compared to previous research [6,9] the runtime has significantly improved to a level where practitioners could apply it.

Table 4. Model statistics and runtime.

Number of BIM objects		1273
Average number of vertices		35
Number of spatial artefacts	Movement spaces	24
	Fall spaces	30
	Fall hazard spaces	66 (of which: 24 permanent, 22 transient, 20 persistent)
Time to generate 3D meshes from IFC via IfcConvert+		340 seconds (1218 meshes; total of 870k triangles)
Time to generate all spatial artefacts		0.181 seconds

5 Conclusion

We have presented a work in progress domain model of safety concepts on a construction site, and an approach for reasoning about safety compliance and mitigation strategies that integrates with 4D BIM construction schedules using Answer Set Programming (ASP) extended to natively support spatial reasoning. A key feature of our approach is the role *spatial artefacts* for representing and reasoning about semantically rich regions of worker perception, behavior and activities. We demonstrate this modeling approach with *fall hazards*, spatial regions where a worker is at risk of falling from a dangerous height. We refine fall hazard spaces according to their position in relation to building elements (e.g. with respect to a sloped roof) and temporal persistence (according to the 4D BIM construction schedule). These refinements are critical for reasoning about alternative mitigation plans that make tradeoffs against cost and construction progress (*lean* construction), particularly when 4D BIMs are incomplete, such as the installation of guard rails or safety nets. Our empirical evaluation on a real 4D BIM shows that our approach runs fast enough to be practical to use on large 4D BIMs.

In our future work we are expanding the scope of hazards to include a wide range visuo-locomotive features that are critical to safety in construction. In order to demonstrate further competitiveness over existing approaches, future testing may focus on highly complex 4D BIMs. Monitoring as-planned vs. as-built situations may yield further insights in how technology [23,24] or combinations thereof can assist future decision making in construction safety and health planning and mitigation.

References

[1] OSHA (Occupational Safety and Health Administration) (2015). Commonly used statistics, https://www.osha.gov/data/commonstats.

[2] Choi, S. (2006). "Fall Protection Equipment Effects on Productivity and Safety in Residential Roofing Construction." *Construction Research,* 7 (1/2) 149-157, doi.org/10.1142/S1609945106000578.

[3] Toole, M., Gambatese, J.A., Abowitz, D.A. (2017). Owners' role in facilitating prevention through design. *Professional Issues in Engineering Education and Practice*, 143 (1) 04016012, doi.org/10.1061/(ASCE)EI.1943-5541.0000295.

[4] Teizer, J. (2016). Right-time vs real-time pro-active construction safety and health system architecture. *Construction Innovation,* 16 (3), 253-280, http://dx.doi.org/10.1108/CI-10-2015-0049.

[5] Zhang, S., Boukamp, F., and Teizer, J. (2015). "Ontology-Based Semantic Modeling of Construction Safety Knowledge: Towards Automated Safety Planning for Job Hazard Analysis (JHA)", *Automation in Construction*, Elsevier, 52, 29-41, http://www.itcon.org/2015/19.

[6] Zhang, S., Teizer, J., Lee, J.-K., Eastman, C., Venugopal, M. (2013). "Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules", *Automation in Construction*, 29, 183-195, http://dx.doi.org/10.1016/j.autcon.2012.05.006.

[7] Melzner, J, Zhang, S., Teizer, J., Bargstädt, H.-J. (2013). "A case study on automated safety compliance checking to assist fall protection design and planning in building information models", *Construction Management and Economics*, 31(6), 661-674, https://doi.org/10.1080/01446193.2013.780662.

[8] Sulankivi, K., Teizer J., Kiviniemi, M., Eastman, C.M., Zhang, S., Kim, K. (2012). "Framework for Integrating Safety into Building Information Modeling." *Proc. of CIB W099*, Singapore, 93-100.

[9] Schwabe, K., Teizer, J., König, M. (2019). "Applying rule-based model-checking to construction site layout planning tasks", *Automation in Construction*, 97, 205-219, https://doi.org/10.1016/j.autcon.2018.10.012.

[10] Zhou, W., Whyte, J., Sacks, R. (2012). "Construction safety and digital design: A review." *Automation in Construction*, 22, 102-111, https://doi.org/10.1016/j.autcon.2011.07.005.

[11] Teizer, J., Melzner, J., Wolf, M., Golovina, O., König, M. (2017). "Automatisierte 4D-Bauablaufvisualisierung und Ist-Datenerfassung zur Planung und Steuerung von Bauprozessen", Bauingenieur, Springer, ISSN 0005-6650, 129-135.

[12] BG Bau (2019) Absturzsicherungen auf Baustellen. https://www.bgbau.de/fileadmin/Medien-Objekte/Medien/Bausteine/b_100/b_100.pdf.

[13] Schultz, C., Li, B., Teizer, J., (2020). Towards a Unifying Domain Model of Construction Safety: SafeConDM. *EG-ICE* (accepted for publication).

[14] Bhatt, M., Schultz, C., Huang, M. (2012). "The shape of empty space: Human-centred cognitive foundations in computing for spatial design." *IEEE Symposium on Visual Languages and Human-Centric Computing*, 33–40.

[15] Schultz, S., Bhatt, M. (2012). "Multimodal spatial data access for architecture design assistance." *AI for Engineering Design, Analysis and Manufacturing*, 26 (2) 177–203.

[16] Schultz, C., Bhatt, M. (2012) "Towards a Declarative Spatial Reasoning System." *20th European Conference on Artificial Intelligence*.

[17] Bhatt, M., Schultz, C, Huang, M. (2012). "The Shape of Empty Space: Human-Centred Cognitive Foundations in Computing for Spatial Design." IEEE Symposium on Visual Languages and Human-Centric Computing. Innsbruck, 33–40.

[18] Bhatt M., Freksa C. (2015). "Spatial Computing for Design -an Artificial Intelligence Perspective." *Studying Visual and Spatial Reasoning for Design Creativity*. doi.org/10.1007/978-94-017-9297-4_7.

[19] Walega, P., Schultz, C., Bhatt, M. (2017). "Non-monotonic spatial reasoning with Answer Set Programming Modulo Theories." *Theory and Practice of Logic Programming*. 17 (2) 205–225.

[20] Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., and Wanko, P. (2016). "Theory solving made easy with clingo 5." *Technical Communications of the 32nd Intl. Conference on Logic Programming*, 52, 2:1-2:15.

[21] Cheng, T., Yang, J., Teizer, J, Vela, P.A. (2010). "Automated Construction Resource Location Tracking to Support the Analysis of Lean Principles." 18th IGLC, Haifa, Israel, 643-653.

[22] OSH-Wiki (2020). Fall arrest systems. https://oshwiki.eu/wiki/Fall_arrest_systems#Types_of_fall_arrest_systems_and_main_components.

[23] Zhang, S., Teizer, J., Pradhananga, N., Eastman, C. (2015). "Workforce location tracking to model, visualize and analyze workspace requirements in building information models for construction safety planning", *Automation in Construction*, 60, 74-86, http://dx.doi.org/10.1016/j.autcon.2015.09.009.

[24] Getuli, V., Capone, P., Bruttini, A., Isaac, S. (2020). "BIM-based immersive Virtual Reality for construction workspace planning: A safety-oriented approach." *Automation in Construction,* 114, https://doi.org/10.1016/j.autcon.2020.103160.