Digital asset verification - State-of-the-art and feasibility study

Michael Unterkalmsteiner* and Alexandros Chirtoglou†
*Department of Software Engineering
Blekinge Institute of Technology
Karlskrona, Sweden
michael.unterkalmsteiner@bth.se
†HOCHTIEF ViCon
Essen, Germany
alexandros.chirtoglou@hochtief.de

I. Introduction

This report aims at answering three questions:

Q1 What is the state-of-art in automated compliance checking in the construction industry?
Q2 To what extent can Trafikverkets technical product requirements be automatically verified?
Q3 What are the common types of verification that are needed for compliance checking?

We answer *Q1* by reporting the results of a recent (2020) review that also developed a roadmap for the goal of wider industry acceptance of automated compliance checking (Section III). We then proceed with the analysis of a sample of Trafiverkets requirements, originating from TRVInfra and Eastlink (Section IV). The goal is to get a better understanding of whether the requirements are formulated in a verifiable manner (*Q2*). Furthermore, we conduct a bottom-up analysis, starting from requirements samples, to identify patterns of what information requirements contain that must be verified (*Q3*). Finally, we demonstrate how a particular type of verification could be implemented in a practical approach (Section V).

II. Background

Model checking consists of three separate steps: BIM validation, clash detection, and compliance checking [1]. In each discipline of the project, BIM validation is conducted to control the models congruence and accuracy in terms of geometric and alphanumeric values. Errors due to wrong modeling or positioning of elements can be detected in this step. When models from different disciplines are merged, clash detection identifies possible issues between models. Finally, in compliance checking, the models are analyzed to ensure that requirements from the client are fulfilled, and rules and regulations are not violated by the design. The focus of DCAT is on compliance checking, which is often also called code checking or rule checking [1].

Compliance checking is an essential activity in all life-cycle phases of a construction project [2]:

- Planning and design aspects of a project are governed by essentially two main sources of requirements: (a) clients specifications on what should be build and how it should be build, and (b) rules and regulations that contain normative requirements as well as resource and land use provisions. Typically, suppliers of digital assets perform compliance checks of design deliverables as part of their contract with the clients.
- During the construction phase, suppliers and subcontractors are subject to health and safety provisions, contractual obligations and other regulatory constraints.
- After the construction phase, asset operation and maintenance requirements need to be followed in order to achieve the facilities safety, reliability, functional and performance objectives.

Compliance checking is in principle based on two data sources:

1) A digital representation of the design data which is subject to the compliance check, i.e. a building model.
2) The requirements that define the needs of clients and regulations that determine the constraints within which the project can be executed and build.

Figure 1 illustrates how these two data sources are used in the manual and automated compliance checking process. In the manual process, compliance checking engineers need to interpret clients needs and regulations and verify that the building model complies to these requirements. This is a highly complex and resource intensive task [3], due to (1) an increasing rate of of updates to regulations and standards, (2) advancements in technology and equipment, and (3) an increase of data and its multidisciplinary nature [4]. In the interview study we conducted in Work Package 1, we found that compliance checking is a challenge in the interaction between suppliers and the client (Trafikverket). Since there are no granular trace links between building model and the relevant requirements, a complete, manual verification would require a large amount of resources.

In the automated compliance checking process, the information captured in natural language requirements needs to be translated into a form that is amenable to computational analysis, i.e. can be interpreted and processed by a

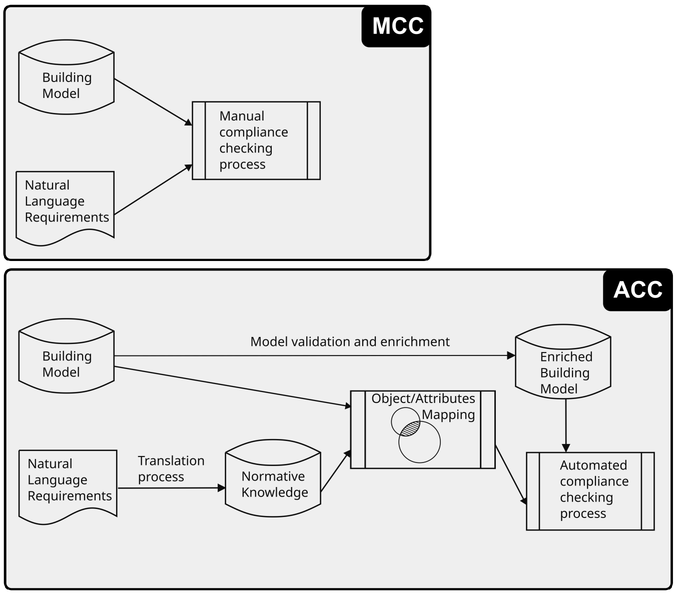

Fig. 1. Manual vs. automated compliance checking process (adapted from Amor and Dimyadi [2])

computer [4]. It is of essence to ensure that this normative knowledge is updated when the natural language requirements change [2], otherwise the results of the automated process are not trustworthy. The building model is a digital presentation of the planned facility and, depending on its development level, has varying levels of detail of the objects that need to be constructed. When the building model is refined and validated, additional information is associated with the modeled objects, such as technical specification data or geological information. In order to enable the automated compliance checking process, the relevant objects and attributes from the building model and from the normative knowledge need to be selected. As indicated in Figure 1, not all objects/attributes in the building model and in the normative knowledge base are relevant at all life-cycle phases of the project. Different compliance check objectives exist depending on the level of development of the building model and whether the project is in the planning, construction or maintenance phase. Finally, the automated compliance checking process uses the object/attributes mapping to extract relevant information from the normative knowledge and the enriched building model and provides a verdict on the status of the model.

While there has been considerable research on enabling the automated compliance check process in the past 50 years [2], few of the proposed approaches have been adopted in practice [4], except for the Construction Real Estate Network (CORENET) project in Singapore [3].

III. State of the art

Beach et al. [3] have conducted a review of the approaches developed for ACC in the past 30 years. Table I provides an overview of these studies, indicating the the subject area in focus of compliance checking in the respective study, whether the approach supports digitisation (i.e. ability to transfer regulations into a form that allows for automated checks), and the supported input and output formats. The approaches that support digitisation are the most relevant to discuss as one of the technical contributions of DCAT is to support the mapping of client requirements and regulations to a classification system. Hence, we briefly summarize their contributions next.

Lee at al. [12] focus on the evaluation of spacial requirements in hospital design, for example accessibility and visibility conditions. They use the Solibri Model Checker (SMC) and extend the Building Environment Rule and Analysis (BERA) language as SMC does not support domain specific implementations of rules. The implemented rules can check for object existence, relations between objects and path distance. Requirements need to be mapped to a geometric model (the paper does not explain how that process work). From that model, a BERA object model is generated to which BERA rule sets can be applied to check the requirements.

Ciribini et al. [1] propose to use compliance checks of BIM models in bids for tender documents. Also their approach uses SMC, but the tender documents are annotated using the RASE [22] methodology. The key idea of RASE is to annotate normative text with the following elements:

- **R**equirement: the actual need expressed in the text, i.e. the property / aspect / characteristic that must be fulfilled by the developed product
- **A**pplicability: a statement that describes where the requirement applies, i.e. limiting the scope of the requirement
- **S**election: a statement that adds circumstances where the requirements applies, i.e. increases the scope of the requirement
- **E**xception: a statement that describes circumstances under which the requirement does not apply.

Annotating regulations with RASE provides important semantic information that can, for example, be used to formalize rules. Ciribini et al. [1] implemented rules concerning geometrical properties (thickness of layers, dimensions of panels of the facade) and their properties ((e.g. cold flexibility values of membranes, static air permeability, water tightness of curtain walls, thermal transmittance of doors, skylights, gates and curtain walls, thermal conductivity of insulation, thermal lag of wooden or concrete roofs, specific power of lightings, energy efficiency of extractors). Furthermore, rule-sets for checking presence of objects were developed (e.g. detectors in the garage). An additional benefit of creating the rule-sets is to identify equivalent and inconsistent tender requirements (e.g. same object named and described differently, different values ranges required for the same property). Ciribini et al. [1] suggest that clients could deliver rule sets with the tender documentation, allowing suppliers check their proposal before submission and allow the jury evaluating the bids a more efficient process. They note however also that not all requirements can be transformed into objective rule-sets. Hence, the jury

TABLE I
ACADEMIC RESEARCH ON AUTOMATED COMPLIANCE CHECKING (ADAPTED FROM BEACH ET AL. [3])

Name	Subject of compliance checking	Digitisation	Input	Output
e-PlanCheck [5] (1990s)	Regulations related to Singapore building design	No	IFC	Report
DesignCheck [6] (1997)	Disabled access regulations	No	IFC	Interactive report
Tan [7] (2010)	Building envelope design	No	Object model	Report
Zhang [8] (2011)	Site safety	No	Tekla API	Report
Melzner [9] (2013)	Site safety	No	IFC	Report
LiCA [10] (2013)	Water distribution systems	No	IFC	Report + visualization
Cheng and Das [11] (2014)	Energy simulation	No	GBXML	Report
Lee [12] (2015)	Not specified	Yes - DSL	IFC	Report
Ciribini [1] (2015)	Tenders	Yes - RASE	IFC	IFC + JSON report
Macit [13] (2015)	Izmir municipality housing and zoning code	No	Not specified	Not specified
RegBIM [14] (2015)	UK building regulations	Yes - RASE	IFC	Report
Zhang [15] (2016)	International building code	Yes - via NLP	ICF	Report
Dimiyadi [16] (2016)	New Zealand building code	Yes - LegalRuleML	IFCOwl	Report
Zhong [17] (2018)	Environmental monitoring	No	IFCOwl	Report
Bus [18] (2018)	French fire safety, accessibility regulations	No	IFCOwl	Report
Zhang [19] (2019)	Multiple use cases (Norway, US, S. Korea)	No	IFCOwl	BCF
Nawari [20] (2019)	Florida building code	Yes	IFCXML	Report
Nawari [4] (2020)	Construction regulations	Yes	IFC	N/A
Messaoudi [21] (2020)	Permitting for State of Florida	No	IFC	Report

cannot be replaced completely as some subjective judgment is still required.

Beach et al. [14] use also RASE to annotate regulations, extending the annotation of each element with metadata:

- the abstract and specific objects from a regulatory, domain-specific ontology
- properties, i.e. type, width, height of an object
- comparison values, i.e. equal, smaller, larger
- values
- units

This process is important to get a detailed understanding of the relationships between the objects that are covered in the regulations. Furthermore, a data-format ontology is needed to describe the objects in the BIM model. Beach et al [14] use IFC-OWL for that purpose and map elements from this ontology to the regulation ontology. With this mapping in place, it is possible to define logical rules that can be evaluated and decided whether (a) a rule is applicable for a specific modeled object, and (b) whether the requirements are fulfilled by the model.

Dimyadi et al. [16] propose to use a language-based compliance checking approach. Their idea is to reuse methods from the legal domain, such as LegalDocML and LegalRuleML, to encode regulatory knowledge that can then be used to execute compliance checks. A key aspect in their proposal is to use OmniClass as a mean to harmonize the potential different names of the same objects used in regulations. A rule in a regulation can then be expressed as a condition and an action. Designer can then capture their tacit knowledge in compliant design procedure (CDP) workflows which specify which objects are checked against which requirements. The advantage of these CDP workflows is that they can be reused for different design alternatives or in different projects. Dimyadi et al. [16] propose also an alternative approach to represent regulatory knowledge with semantic web technologies. The advantage of that approach is the possibility to use existing inference engines. While this might be appealing, the authors also point out that there exist still many decisions that are best handled by the tacit knowledge of a human. In other words, a system that supports compliance checking by providing reusable workflows and minimizing manual information lookup, while keeping human decision-making and intuition in the loop, is preferable.

Zhang and El-Gohari [15] propose a method, based on natural language processing (NLP), to extract information from regulatory requirements. Their principle idea is that text in regulations can be analyzed by hand-crafted, rule-based extraction methods that identify information relevant for compliance checking (for example, requirements that define geometrical constraints on objects, or the existence of objects). The following information elements are extracted:

- *Subject* is a ”thing” (e.g. building object, space) that is subject to a particular regulation.
- *Compliance checking attribute* is a characteristic of ”subject” by which is compliance is assessed (e.g. the width of a door).
- *Deontic operator indicator* describes whether a regulation represents an obligation, permission or prohibition.
- *Quantitative relation* describes the relation between a compliance checking attribute (subject characteristic) and a quantity. For example, in the requirement ”The door width shall be increased by 3 cm by each 100 square meter room size”, ”increase” is a quantitative relation.
- *Comparative relation* is used to compare quantity values, such as greater than, or less than or equal.
- *Quantity value* is a number, or a range of numbers, defining the quantified requirement.
- *Quantity unit* refers to a unit for the quantity value.
- *Subject restriction* and *quantity restriction* place a constraint on a subject and quantity value respectively.

TABLE II
ROOT CAUSES FOR NOT ADOPTING ACC IN PROFESSIONAL PRACTICE (ADAPTED FROM BEACH ET AL. [3])

Root cause	Frequency
Lack of precise, digitisable regulations	21
Lack of standardized data models for regulatory compliance data	18
Lack of a clear government direction towards ACC	12
Cultural resistance to accepting ACC	7
Lack of investment in ACC	5
Lack of technology/tools to support checking as-built assets	4
No business models factoring in (a) reduced costs for assessment, (b) faster turnaround for assessment, (c) ability to pre-check before formal submission	4
Lack of awareness of the meaning automation of regulations, requirements, standards, and its benefits	4
Lack of generative design tools based on regulations/requirements	3
Lack of implementation of smart contracts	3
Lack of standardized APIs for compliance checking tools	3
Insufficient professional development and training in compliance checking	3
Poor compliance checking process definition, standardisation and management	2
Lack of explicit linkages between requirements, designers and product suppliers and their data	2
No services to enable certification of software as performing ”correct”	2
Poor structured product data standards	2
Existing of negotiated regulations decreasing the transparency of regulations	2
Lack of formal data ”Chain of custody”	1
Lack of dual automated and engineered paths to ease transition	1

The approach assumes that each requirement contains 1) exactly one subject, comparative relation, quantity value, quantity unit, 2) at most one compliance checking attribute, deontic operator indicator, and quantity relation, and 3) zero or more instances of subject and quantity restriction.

Nawari [4] proposes a generalized, adaptive framework (GAF) for a computer supported review process of design models. The framework is based on an open data standard (ICF). While the framework has not been implemented in practice, it defines five key steps that are critical for every automated compliance checking process.

1) Classification of regulations into four groups: (1) provisory (explicit rules), (2) content (definitions), (3) dependent (on provisory clauses), and (4) ambiguous (subjective provisions).
2) Description of data exchange requirements in the form of an information delivery manual (IDM).
3) Development of a feature extraction algorithm for all objective (unambiguous) data that translates the regulations into a object-based model.
4) Formalization of subjective data using fuzzy logic allowing to extract and represent information ambiguous regulations.
5) Linking data generated in the above steps and the to be checked design model.

Common to all the approaches discussed so far is the reliance on Industry Foundation Classes (IFC) as a mean to enable data modeling and achieve interoperability.

A. *Outstanding challenges and roadmap*

While research has made some progress towards solving specific problems in ACC (like extracting information from regulations or modeling regulations so that they can be processed computationally), ACC has not seen yet a wide adoption in industry. To identify root causes for this resistance, Beach et al. [3] surveyed industry professionals (n=60). Table II provides an overview the identified root causes together with the frequency by how many respondents it was brought up.

Based on the identified challenges, Beach et al. [3] developed in a workshop together with industry professionals a roadmap that illustrates 24 capabilities that they deem necessary to achieve adoption of ACC on a wide scale. We summarize this roadmap in Table III. The category column indicates whether the capability is of technical (T), political (P) or cultural (C) nature. The roadmap has also been segmented into four incremental phases. While not described explicitly by Beach et al., we conjecture that this roadmap could also be used to assess the current state of ACC adoption in a project or even nation-wide. This would allow one to identify and focus on the capabilities that need further development.

It is noteworthy that each stage contains a mixture of at least two of technical, cultural and political capabilities, indicating that progressing to the next stage requires work and engagement in all three fields. While we have seen that research has produced technical solutions and proposals, a lack in engaging in cultural and political capabilities will hinder progress in ACC adoption.

IV. VERIFIABILITY OF PROJECT SPECIFIC AND TRVINFRA REQUIREMENTS

The first capability necessary to adopt ACC is to be able to analyze and understand which regulations and requirements are suitable for automation (see Table III). One fundamental quality attribute of a requirement is its verifiability, that is, the quality that allows one to decide whether the requirements has been fulfilled or not. Requirements verifiability is a complex requirements quality attribute that can be further broken down, i.e. a requirement must be understandable, complete, unambiguous, and atomic to be verifiable [23].

TABLE III
ROADMAP FOR THE ADOPTION OF ACC (ADAPTED FROM BEACH ET AL. [3])

No.	Capability	Category
Stage 1 - Research		
1	Cataloguing and prioritizing regulations that are suitable for automation	T
2	Engaging policy makers/implementors in the digitisation agenda	P
3	Presentation of the case for digitisation of compliance checking to establish funding to conduct proof of concept prototype	P
Stage 2 - Development of pilot or proof of concept		
4	Development of rule processes to track decisions, feedback, and uncertainty	T
5	Detailed mapping of digitised regulation/requirement/standards processes	T
6	Digitisation to be given voice with policy-implementors to ensure future support	P
7	Development of an understanding of parallel regulations	P
Stage 3 - Industrialisation of pilot or proof of concept		
8	Persistent data linkages between requirements and supplied product to prevent variation on specification	T
9	Chain of custody of materials and data	T
10	Accommodate multiple data models and multiple data dictionaries	T
11	Specification of a continual feedback loop process to incorporate appeals/derogations/determinations data in reviewing regulations	T
12	Production of audience specific guidance on digitisation of regulations or requirements	C
13	Detailed evidence-based business model for digitization of regulatory compliance	C
14	Explore routes to export developed toolchains to international audience and exploit international developments	C
15	Creation of standard data and criteria for social, environment and economic impact assessments	P
16	Conducting Impact assessment of digitisation of regulations	P
Stage 4 - Scaling of industrialized product or process		
17	Investigation of relationship between regulations and identification of overlaps and gaps	T
18	Enabling development of generative design based on regulations and requirements	T
19	Consistent/Structured data models and APIs (Application Programming Interface) for compliance checking	T
20	Continuously checking the quality of assets using calibrated instrumentation along with other data sources	T
21	Definition of precise digitised regulation clauses	T
22	Calculation method validation services	C
23	Develop robust inspection methods/rules to reduce dependence on human inspectors	C
24	Professional development and training in compliance checking for all that interface with it - including clients and supply chain.	C

Trafikverket has hundreds of regulatory documents (TRVInfra), each containing dozens to hundreds of requirements. Furthermore, each project elicits specific requirements originating from the different stakeholders and the goals that are targeted with the project. Hence, in order to render this task tractable within the scope of this project, we have analyzed a sample of project specific requirements (Eastlink) and TRVInfra requirements. The overarching goal was to gauge the potential for automated verification of the sampled requirements.

To drive the analysis, we asked the following questions:

Q1 Is the requirement verifiable?
Q2 If the requirement is not verifiable, why not?
Q3 Is the requirement, in principle, automatically verifiable?
Q4 If the requirement is not automatically verifiable, why not?
Q5 Can we identify archetypes of verification procedures?

We analyzed in total 129 requirements. One researcher (BTH) performed an initial review of the requirements and classified each requirement as automatically verifiable or not (Q3). In addition, he provided a brief description to motivate this decision (Q4). Furthermore, he extracted the information that is needed to verify the requirement and formulated a simple procedure how the requirement could be verified. That procedure was then used to identify verification archetypes, i.e. common procedures that can be used a variety of different requirements (Q5). The information extracted for Q3-Q5 was reviewed by a practitioner (HochTief Vicon) and amended in case the feedback required a change in the assessment made by the researcher. Upon inspection of the answers to Q4, the researcher found that it would make sense to differentiate between requirements that are not verifiable at all and requirements for which an automated verification seems difficult or impossible. We added therefore Q1 and Q2, as it makes sense, from a reporting perspective, to start with these questions, even though they were analyzed last.

A. *Verifiable requirements (Q1, Q2)*

Out of the 129 requirements, 122 (95%) were verifiable whereas 7 (5%) were assessed as not verifiable. The requirements and the reasons for non-verifiability are illustrated in Table IV. Seven out of 129 requirements is a relatively low number and is acceptable if the requirements are clarified or made more specific in other related requirements. If this is the case (we did not investigate that), it should still be ensured that the non-verifiable requirement has an explanation/qualification where the more specific requirement(s) can be found. While some of the non-verifiable requirements are associated with advisory text (råd), this additional information should rather be encoded in a requirement that can be verified.

B. *Automatically verifiable requirements (Q3, Q5)*

Out of the 122 verifiable requirements, we classified 103 (84%) as automatically verifiable. While analyzing the requirements, we have developed a set of six verification archetypes. A verification archetype is a fundamental check that describes

TABLE IV
NON-VERIFIABLE REQUIREMENTS

Source document	Requirements text	Reason for non-verifiability
TRVINFRA-00224 - Vägöverbyggnad / Överbyggnad väg, Dimensionering och utformning (K109662)	Roads should be designed and constructed to achieve acceptable smoothness.	Ambiguous (what is acceptable?)
TRVINFRA-00231 - Avvattning / Avvattning, Dimensionering och utformning (K111335)	Drainage should primarily be arranged in such a way that the natural water balance is maintained. Dewatering should be arranged so that water is managed as locally as possible. Water shall be transported by the shortest possible route before being infiltrated or discharged.	Ambiguous (natural water balance unclear; expression such as ”x as possible” are ambiguous); not atomic (several requirements in one)
TRVINFRA-00231 - Avvattning / Avvattning, Dimensionering och utformning (K111359)	Roads and railways must be designed so that their life is not shortened by high water levels in the superstructure.	Ambiguous (what is high?), unspecific (shortened life time, what are the tolerances?)
TRVINFRA-00233 - Tunnel / Tunnelbyggande (K43886)	The bank body must be designed so that the track or drainage is not damaged by freezing.	Ambiguous (we assume that it should not be damaged by freezing water, but could also be simply air temperature)
TRVINFRA-00233 - Tunnel / Tunnel construction (K44303)	Transducers and detectors shall be installed to the extent necessary for the operation, monitoring and maintenance of the tunnel.	Ambiguous/incomplete (what is the extent necessary?)
TRVINFRA-00008 - Ban- och stationsutformning / Personskydd mot järnväg (K29206)	Foundations shall be designed to withstand all applicable load cases from the column, without residual displacement, deformation or cracking.	Ambiguous/incomplete (what are the applicable load cases?)
Projektspecifika krav för Ostlänken (263630)	When choosing the location of a technology yard, local conditions must be taken into account, such as the surrounding terrain, accessibility to service roads / access roads and landscape conditions.	Incomplete (what are the decision criteria?)

what should be verified to ensure that the requirement is fulfilled. Please note that, as we analyzed only a sample of 129 requirements, these archetypes may be incomplete and more can be added at when a larger of requirements is analyzed.

Table V provides the names of the six archetypes, an example requirement and the occurrences we counted in the 129 analyzed requirements. In the example requirement, we highlight the object which needs to be verified in *italics* and the property of that object that needs to be verified in **bold**. The **location** on an object describes where an object shall be placed within a model. The **distance** is similar, describes however the relationship between objects in terms of distance. An **internal attribute** is a property of an object that can likely be modeled as part of the digital building model. An **external attribute** is a property that can likely be found only in external documentation, linked to from the model. **Geometrical attributes** are physical dimensions of objects, as they are either specified in the model as attributes (a special case of an internal attribute) or derived from the actual object geometry. The **existence** of an object describes that an object must exist in a model.

Note that the verification archetypes are a simplification and capture only the essence of what needs to be verified in the requirement. For example, a requirement may specify that something is *not* located at a particular place. Another, more complex requirement may specify that the distance between two objects shall be a certain value, *only if* certain conditions exist. Exactly how these semantics can be captured and extracted from requirements, as to determine which automated verification techniques shall be applied, is a matter of future research.

C. Not automatically verifiable requirements (Q4)

Out of the 122 verifiable requirements, we found 19 (16%) to be not automatically verifiable. Table VI provides a few of these requirements with a motivation why we assessed them a not automatically verifiable.

We believe that, with help of domain experts, these requirements could be clarified to a degree that an automated verification is possible or at least established whether the requirements should rather be verified during or after the construction, instead of the design phase.

V. AUTOMATED COMPLIANCE CHECKS - AN EXAMPLE IMPLEMENTATION

To demonstrate automated compliance checks, we have transferred the textual requirements description into a logical syntax, which can be verified with a model-based rule check. As an example, we used the requirement K122148. At first, we had to identify those objects which must be addressed to verify the requirement. The identification of these objects was enabled with the translation of the textual description ”superstructure” into the classification value, which is the PDS 09 value ”31BJ–”.

Furthermore, the reference to a table (Table 19-1 in TRVInfra-00224) of the requirement description defined the attributes that needed to be used for the verification. For

TABLE V
VERIFICATION ARCHETYPES AND THEIR OCCURRENCES IN THE 129 REQUIREMENTS

Verification archetype	Example	Occurrences
Localization of an object	*Fences* must be **placed behind the technology building, seen from the railway**, if the property boundary allows this.	9
Distance between objects	*Cross-connections*, between up and down tracks, shall be provided with a **maximum distance of 40 km**.	9
Internal attribute(s) of an object	If the support layer thickness is greater than 120 mm, a **coarser 0/45** *support layer* should be selected for stability reasons.	14
External attribute(s) of an object	*Railings on railway bridges* shall be designed with **safety nets**.	31
Geometrical attributes of an object	The *hardened walkways* in the track tunnel should be **1,2 m wide** (minimum free width).	20
Existence of an object	There should be *emergency lighting* in service and access tunnels.	20

TABLE VI
EXAMPLES OF NOT AUTOMATICALLY VERIFIABLE REQUIREMENTS

Source document	Requirements text	Reason for no automated verification
TRVINFRA-00003 - Ban- och stationsutformning / Spårgeometri (K30636)	Track length measurement shall refer to the track centre.	Process requirement
TRVINFRA-00003 - Ban- och stationsutformning / Spårgeometri (K30750)	Mileage boards must be made according to standard drawing 3-803122	Complex external information
TRVINFRA-00307 - Signalsystem / Byggnation (K125706)	Balises shall be mounted with the long sides parallel to the rails. The maximum permitted deviation is 3°, which corresponds to a maximum difference of 18 millimetres between dimensions A and B in the figure below.	Verifiable at construction time.
TRVINFRA-00233 - Tunnel / Tunnelbyggande (K65617)	For concrete and shotcrete, excluding high-strength and self-compacting concrete, it shall be assumed that they do not split at temperatures of the concrete surface below of 500C.	Not sure if even verifiable
PM Projektspecifika krav för Ostlänken (502169)	Level of use for track body, in bank, tunnel and intersection, is defined as the level 1.0 meter below RÖK (rail top).	Missing domain knowledge to make an assessment.

our example, the climate zone and the material type would define the least thickness of the related road layers. That meant that two attributes were required to be assigned to all relevant model objects with the classification values ”31BJ–”. These are the suggested attributes ”DCAT_ClimatezoneReq” and ”DCAT_MaterialTypeReq” as shown in Figure 2 under the DCAT properties section of the Model check definition. These attributes, for the sake of the test purpose of project, were assigned manually to the test models.

In addition, Table 19-1 indicates the minimum thickness of the relevant road layers based on the assigned material type climate zone. This geometrical attribute was calculated using basic scripting functions within the 3D BIS demonstrator tool and the property “DCAT_ThicknessReq” was automatically calculated and assigned to the proper model objects.

Then, we setup once the model based rule which:

- Identifies the objects that need to be checked (PDS09 value ”31BJ–”) and ignores the rest
- Uses the extracted information for the combination of the specific DCAT_ClimatezoneReq and DCAT_MaterialTypeReq attributes and the calculated DCAT_ThicknessReq property

If the DCAT_ThicknessReq is equal or more than the expected result, then all checked objects are highlighted green

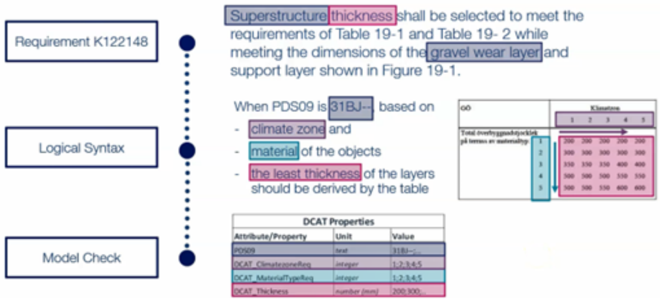

Fig. 2. Logical syntax for requirement K122148 and model check attributes

in the model area. Furthermore, if all objects fulfill the defined rule, the check state for the rule is set to “Passed”.

VI. CONCLUSIONS

One of the key root causes for not adopting automated compliance checks in industrial practice is the lack of precise, digitalisable regulations [3]. In order to understand if this is also the case for the regulations in use at Trafikverket, we have studied a sample of requirements drawn from TRVInfra and Eastlink. We found that the vast majority (95%, 122 out of 129) are verifiable, and 83% (103 out of 122) of these are probably automatically verifiable with existing technologies. Furthermore, we identified a set of six verification archetypes that cover the 122 analyzed requirements. This means that the implementation of one archetype can already cover a large set of requirements. For example, the check for the existence of an object was found in 20 out of 103 requirements (19%). These initial results are encouraging and call for a more in-depth study of the complete corpus of requirements at Trafikverket. However, a manual assessment of thousands of requirements is prohibitively expensive. Therefore, future work can investigate the feasibility of such analysis through clustering [24], i.e. by grouping requirements by their similarity and then analyze these clusters, presumably less in number than the original requirements, w.r.t. to their verifiability.

References

[1] A. L. C. Ciribini, M. Bolpagni, and E. Oliveri, "An innovative approach to e-public tendering based on model checking," *Procedia Economics and Finance*, vol. 21, pp. 32–39, 2015.

[2] R. Amor and J. Dimyadi, "The promise of automated compliance checking," *Developments in the Built Environment*, vol. 5, p. 100039, Mar. 2021.

[3] T. H. Beach, J.-L. Hippolyte, and Y. Rezgui, "Towards the adoption of automated regulatory compliance checking in the built environment," *Automation in Construction*, vol. 118, p. 103285, Oct. 2020.

[4] N. O. Nawari, "Generalized adaptive framework for computerizing the building design review process," *Journal of Architectural Engineering*, vol. 26, no. 1, p. 04019026, 2020.

[5] B.-H. Goh, "E-government for construction: The case of singapore's corenet project," in *Research and practical issues of enterprise information systems II*, pp. 327–336, Springer, 2007.

[6] C. S. Han, J. Kunz, and K. H. Law, "Making automated building code checking a reality," *Facility Management Journal*, pp. 22–28, 1997.

[7] X. Tan, A. Hammad, and P. Fazio, "Automated code compliance checking for building envelope design," *Journal of Computing in Civil Engineering*, vol. 24, no. 2, pp. 203–211, 2010.

[8] S. Zhang, J.-K. Lee, M. Venugopal, J. Teizer, and C. Eastman, "Integrating bim and safety: An automated rule-based checking system for safety planning and simulation," in *Proceedings of CIB W99 Conference*, pp. 1–13, 2011.

[9] J. Melzner, S. Zhang, J. Teizer, and H.-J. Bargstädt, "A case study on automated safety compliance checking to assist fall protection design and planning in building information models," *Construction Management and Economics*, vol. 31, no. 6, pp. 661–674, 2013.

[10] J. P. Martins and A. Monteiro, "Lica: A bim based automated code-checking application for water distribution systems," *Automation in Construction*, vol. 29, pp. 12–23, 2013.

[11] J. C. Cheng and M. Das, "A bim-based web service framework for green building energy simulation and code checking," *Journal of Information Technology in Construction (ITcon)*, vol. 19, no. 8, pp. 150–168, 2014.

[12] Y. Lee, C. Eastman, and J. Lee, "Automated rule-based checking for the validation of accessibility and visibility of a building information model," in *Computing in Civil Engineering 2015*, pp. 572–579, 2015.

[13] S. Macit, M. E. İlal, G. Suter, and H. M. Günaydın, "A hybrid model for building code representation based on four-level and semantic modeling approaches," in *CIB W78 Conference*, pp. 517–526, 2015.

[14] T. H. Beach, Y. Rezgui, H. Li, and T. Kasim, "A rule-based semantic approach for automated regulatory compliance in the construction sector," *Expert Systems with Applications*, vol. 42, no. 12, pp. 5219–5231, 2015.

[15] J. Zhang and N. M. El-Gohary, "Semantic nlp-based information extraction from construction regulatory documents for automated compliance checking," *Journal of Computing in Civil Engineering*, vol. 30, no. 2, p. 04015014, 2016.

[16] J. Dimyadi, P. Pauwels, and R. Amor, "Modelling and accessing regulatory knowledge for computer-assisted compliance audit," *Journal of information technology in construction*, vol. 21, pp. 317–336, 2016.

[17] B. Zhong, C. Gan, H. Luo, and X. Xing, "Ontology-based framework for building environmental monitoring and compliance checking under bim environment," *Building and Environment*, vol. 141, pp. 127–142, 2018.

[18] N. Bus, F. Muhammad, B. Fies, and A. Roxin, "Semantic topological querying for compliance checking," in *eWork and eBusiness in Architecture, Engineering and Construction*, pp. 459–464, CRC Press, 2018.

[19] C. Zhang, "Requirement checking in the building industry: enabling modularized and extensible requirement checking systems based on semantic web technologies," 2019.

[20] N. O. Nawari, "A generalized adaptive framework (gaf) for automating code compliance checking," *Buildings*, vol. 9, no. 4, p. 86, 2019.

[21] M. Messaoudi and N. O. Nawari, "Bim-based virtual permitting framework (vpf) for post-disaster recovery and rebuilding in the state of florida," *International Journal of Disaster Risk Reduction*, vol. 42, p. 101349, 2020.

[22] E. Hjelseth and N. Nisbet, "Capturing normative constraints by use of the semantic mark-up RASE methodology," in *Proceedings of CIB W78-W102 Conference*, pp. 1–10, 2011.

[23] M. Unterkalmsteiner and T. Gorschek, "Requirements quality assurance in industry: why, what and how?," in *23rd International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ)*, (Essen, Germany), pp. 77–84, Springer, 2017.

[24] R. Zhang and N. M. El-Gohary, "A Clustering Approach for Analyzing the Computability of Building Code Requirements," pp. 86–95, Mar. 2018.