Delft University of Technology

Document Version
Final published version

Licence
CC BY

Citation (APA)
Akın, A. T., Villar, A., Diakite, A., El Yamani, S., Stoter, J., Noardo, F., Atkinson, R., & Zaborowski, P. (2026). Enhancing the Interpretability of 3D City Model Validation Through Web Visualization: The Case Study of the CHEK Validation Results Viewer. *ISPRS International Journal of Geo-Information*, *15*(7), Article 282. https://doi.org/10.3390/ijgi15070282

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states "Dutch Copyright Act (Article 25fa)", this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

International Journal of *Geo-Information*

Article

Enhancing the Interpretability of 3D City Model Validation Through Web Visualization: The Case Study of the CHEK Validation Results Viewer

Alper Tunga Akın [1,2,*], Alejandro Villar [3], Abdoulaye Diakite [4], Siham El Yamani [1], Jantien Stoter [1], Francesca Noardo [3], Robert Atkinson [3] and Piotr Zaborowski [3]

1 3D Geoinformation Research Group, Department of Urbanism, Delft University of Technology, 2628 CD Delft, The Netherlands
2 Department of Geomatics Engineering, Faculty of Engineering, Karadeniz Technical University, 61080 Trabzon, Türkiye
3 Open Geospatial Consortium Europe, Technologielaan 3, B-3001 Heverlee, Belgium; fnoardo@ogc.org (F.N.)
4 CityGeometrix Pty Ltd., Sydney, NSW 2017, Australia
* Correspondence: alpertunga@ktu.edu.tr

Abstract

3D city models (3DCMs) are increasingly used in urban simulations, cadastral workflows, and digital building permit processes, and automated validation of these models has become a routine requirement. Existing validation services typically produce dense, frequently generated by SHACL engines, text-based reports that are difficult for users without a semantic-web background to interpret. This paper describes the CHEK Validation Results Viewer (CHEK VRV), a Flask-based web application that couples the OGC Data Completeness Validator with an interactive Three.js 3D viewer of the CityJSON input. SHACL violations and geometric invalidities are spatially explorable on a visual representation of the model, and the tool supports both predefined CHEK Building Block Profiles and user-uploaded custom profiles. We report a formative usability study with twelve domain experts (urban planners, data vendors, academics) and use it to identify the tool's current strengths, its limitations, and a prioritized development roadmap. As reported by the participant experts, such a presentation is easier to interpret than the raw JSON-LD report.

Keywords: 3D city models; data validation; SHACL; digital building permits; 3D visualization; web architectures; CHEK; OGC

Check for updates

Academic Editor: Wolfgang Kainz

Received: 19 March 2026
Revised: 12 June 2026
Accepted: 22 June 2026
Published: 25 June 2026

Copyright: © 2026 by the authors. Published by MDPI on behalf of the International Society for Photogrammetry and Remote Sensing. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

3D city models (3DCMs) widely provide detailed urban representations for a wide range of applications, including visualization, urban simulation, and 3D cadastre [1]. Preventing any time or economic losses in production chains of these efforts requires qualified and standards-compatible 3DCMs [2,3]. In this context, researchers have started to develop automated validation workflows that minimize human intervention and error propagation and ensure standardization in terms of both geometric and semantic aspects of 3DCM data.

Therefore, research on this type of validation is conducted using standardized data models like CityGML, CityJSON, or IFC for different domains [4,5]. Geometric validation of these models is most commonly performed against the data-model specifications themselves [2,4] and the ISO 19107 spatial schema [6]. However, if a use case involves various definitions or instances from multiple domains or application schemas, definition conversions are necessary before performing semantic validation [7]. Such conversions are often complex, error-prone, and difficult to maintain, and can cause redundant attribute mapping between sources, especially when the target domains use heterogeneous or evolving schemas [8].

To address this interoperability challenge, recent research has explored Semantic Web (SW) technologies, which offer a formal, machine-readable way to link concepts across domains and translate between different definitions [9,10]. Particularly, SHACL (Shape Constraint Language) is the new research trend, as a W3C recommendation [11]. This standard enables the validation of data-centric information management in the Architecture, Engineering, and Construction (AEC) sector, ensuring compliance with domain-specific requirements and conformity to container schemas [9].

However, the current SHACL-based validation workflows lack interactive and human-readable reporting. Even in recent studies, validation results are typically expressed as raw documents. A second approach embeds the results as metadata within the input dataset itself [9,10,12].

The software presented here, the CHEK Validation Results Viewer (CHEK VRV), addresses this gap by coupling the OGC Data Completeness Validator (DCV) and the CHEK Building Block Profiles with an interactive CityJSON-based 3D viewer. Validation results from the DCV's SHACL processor and from the val3dity geometric checks are presented within a single web interface where users can click a violation message and see the affected building or surface highlighted on the 3D model.

This paper makes three contributions. First, we describe a workflow that couples the DCV with an interactive 3D viewer so that SHACL and val3dity outputs are resolved against the same visual representation of the model. Second, we describe a JSON-LD framing strategy that preserves the RDF semantics of the validation report while presenting it in a structure consumable by a Three.js-based front end, removing the need for an in-browser SHACL processor. Third, we report a formative usability study of the resulting tool with twelve domain experts and use it to map the tool's current strengths and weaknesses to a development roadmap.

The formative usability study's purpose is to find specific interpretability and interface problems while the tool is still under development, not to test hypotheses about the tool's performance against alternatives or to estimate usability scores for a wider population. Twelve participants are a small sample in absolute terms, but samples of this size are standard in formative usability work, where seven to twenty-one evaluators have been shown to uncover most usability problems [13]. However, such samples support only limited claims. We therefore evaluate the study by what it reveals, namely problem discovery and agreement or disagreement across the participating user groups, rather than by statistical significance, and we treat the quantitative scores in Section 4 as descriptive of this sample only. The findings the study is meant to support are two: that the participating experts judged the coupled 3D presentation easier to interpret than the raw JSON-LD report, and that the weaknesses they recorded translate into priorities for the development roadmap.

The remainder of this paper is organized as follows. Section 2 reviews existing validation methods and visualization approaches and positions CHEK VRV against five reference systems. Section 3 describes the test data and software, and the design of the formative usability study. Section 4 presents the application's outputs on the Lisbon profile and the quantitative and qualitative results of the user study. Section 5 discusses these results, addresses limitations and scalability, and uses SWOT framing to translate the findings into a development roadmap. Section 6 concludes the paper.

2. Background

2.1. Essentials of Semantic Web Technologies and Applications in the AEC Industry

SW enhances the traditional web by enabling humans and computers to work together through information that carries explicit meaning, thus supporting interoperability [14]. It uses the Resource Description Framework (RDF) to represent data as interconnected resources, forming knowledge graphs (KGs) that consist of entities linked by relationships [15].

RDF expresses data as triples of subject, predicate, and object, where subjects and objects, identified by URIs, act as nodes, and predicates represent edges. KGs can describe domain concepts and properties (ontologies) using languages such as OWL (Web Ontology Language) [16] or depict relationships between individual instances. Linking ontologies with instances enables the exchange of semantically rich data, even across heterogeneous or unforeseen schemas [17,18]. SHACL provides a framework for validating RDF data by defining shapes that specify structural and semantic constraints. These shapes, organized into a shapes graph, can be viewed as a form of KG [19,20]. SHACL processors compare each shape with its target RDF node, report violations, and ensure that data meets the required definitions, enabling consistent multi-domain validation [21].

This mechanism is crucial for ensuring data integrity and compliance within complex, interconnected datasets, particularly in environments like the Architecture, Engineering, and Construction industry, where diverse data sources need to conform to specific standards and requirements [22]. Unlike traditional RDF reasoning methods based on Web Ontology Language, SHACL was primarily designed for data validation rather than inference, providing a robust mechanism for defining structural and semantic constraints on RDF graphs [23].

A recent well-fitted use case of SHACL processors is DBP efforts. The process of Building Permitting is a sequence of defining rules and construction control processes and is an authorization that permits the start of the construction of a building [24,25]. The digitalization of this process promises to eliminate the hassle of expert shortage and the subjectivity of human checking processes, as well as to simplify the management and the interpretation of regulations, resulting in time and cost savings [25,26]. DBPs are conducted with rule-based compliance checks against a comprehensive set of standards, zoning laws, and relevant regulations [27,28]. By encoding these rules as SHACL constraints, the DBP process can be streamlined. This integration allows for automated validation of building designs against regulatory requirements, leading to faster approvals and reduced human error in the permitting workflow [9].

SHACL processors' ability to express complex constraints, including cardinality, data types, and relationships, allows for comprehensive validation of building permit data, ensuring that 3DCMs adhere to the necessary technical requirements. In this direction, OGC developed the "Data completeness validator" service and "OGC Building Block Profiles" as a part of the CHEK project, which is funded by the European Union [29]. The service includes a SHACL processor to validate 3DCM data, and the profiles are SHACL graphs that include technical requirements for regulations of various European cities.

OGC CHEK Data Completeness Validator and CHEK Building Block Profiles

The validation solution developed in the CHEK project has two key components that address the validation needs of CityJSON or CityGML formatted 3D models [30,31].

CHEK Building Block Profiles: SHACL shapes graphs that encode the technical and regulatory requirements of specific European cities. These profiles formalize constraints, such as minimum building heights, volumetric limits, or the presence of mandatory attributes in a machine-readable form, enabling automated compliance checking for municipality requirements that combine data requirements and regulations. Different sets of rules are published as profiles. Datasets can then be declared (or validated) to be compliant with a given profile if they fulfill the constraints of all its validation rules. The Profiles Vocabulary [32] can be used to describe profiles in either RDF format or using the OGC Building Blocks [33], which allows users not only to include metadata and associate the profile with one or more collections of rules (e.g., as SHACL files), but also to define inheritance and/or specialization relationships, fostering reusability (i.e., if Profile A is a sub-profile of Profile B, any dataset aiming to be compliant with the former is required to also comply with the latter, without needing to redefine them, thereby reducing duplication).

Data Completeness Validator (DCV): A web-based service that processes 3DCM datasets in formats such as CityGML or CityJSON, converting the input data into RDF format, applying both geometric and semantic checks. At its core, the DCV incorporates an SHACL processor to evaluate RDF representations of the input data against pre-defined profiles. Geometric checks are employed through the open source val3dity software (version 2.6.0). The geometric invalidities are added to the validation report after the semantic validation is performed using the Building Block Profiles.

2.2. Visualization and Error Reporting Trends in SHACL Studies

While SHACL offers robust validation capabilities, the interpretability of its output, particularly in complex scenarios, poses a significant challenge for end-users [9]. To underscore the role of the CHEK VRV to fill this gap, it is crucial to analyze current trends in visualization and error reporting within SHACL-based validation frameworks. While many SHACL validation results are not inherently human-friendly, especially for individuals without a technical background, current methodologies often focus on system development and application domains rather than on evaluating the visual designs for diverse user groups. This gap highlights a need for more intuitive and accessible visual analytics tools for 3D urban data [34]. Moreover, the transformation of complex validation outputs into a comprehensible format is paramount for diverse stakeholders, necessitating sophisticated visualization techniques that can abstract technical details while retaining essential diagnostic information. This includes not only identifying defects but also providing actionable insights for their rectification, moving beyond mere error reporting to a more proactive problem-solving paradigm [35].

To establish an information exchange channel between diverse stakeholders, the CityGML Quality ADE serves as a crucial foundation. This approach enables the seamless integration of validation results directly within the data model, thereby facilitating efficient data exchange and substantially enhancing data quality across various applications [36]. In the context of employing knowledge graphs and SHACL engines for validating CityJSON models, the framework outlined in [37] delivers a comprehensive methodology for self-descriptive and transparent data verification, thereby establishing a vital communication channel among stakeholders. Moreover, it implements an effective grouping of SHACL violations by error type or SHACL shapes, yielding structured and interpretable outputs that empower users to prioritize and resolve issues based on semantic categorization rather than isolated occurrences in the data structure. Although this methodology excels at highlighting geometric errors directly on the 3D model, it notably lacks interpretable visualizations for semantic or attribute completeness errors, revealing a critical limitation in making complex validation results accessible to non-expert users.

The web-based platform developed by Emamgholian et al. [38], iTwin4PP, utilizes a visual land-use regulation assessment using Bentley's iTwin platform. This platform automates the modeling of 3D Land Use Regulations (LuRs) and integrates them into a combined BIM and GIS environment, enabling spatial analyses to detect conflicts between

LuRs and physical/planning objects. Although this platform shares a comparable visualization objective with the proposed CHEK VRV and employs rule-based compliance checking, it lacks interoperable validation mechanisms such as SHACL and is confined to LuRs for Melbourne, thereby restricting its generalizability across heterogeneous data ecosystems.

Although the IFC model viewer, incorporating validation results from [5], adopts a comparable visualization strategy to the CHEK VRV, its primary emphasis is on SHACL-based validation of BIM data, without prioritizing the provision of an open web-based tool.

Generally, validation studies only adopt triple store-based textual validation reports, in case their main objective was centered on machine-to-machine communication or report observations of experts who have domain knowledge on the specifics of the dataset or SW technologies; in other words, only a document compliance check effort against SHACL constraints. But projects like CHEK can require more sophisticated visualization and reporting mechanisms to make validation results accessible to a broader audience, including those without SW-specialized technical expertise. Especially when dealing with manual editing tasks, such as modifying a 3D model within a CAD environment or inserting attributes, immediate visual feedback on identified errors is essential for effective correction. This immediate feedback loop can significantly reduce the iterative cycle of error identification and remediation, thereby enhancing productivity and data quality in AEC workflows.

This study, CHEK VRV, addresses this gap by proposing a web application that utilizes OGC CHEK Building Profiles and a DCV service, and offers an interactive CityJSON-based visualization. The application utilizes SHACL-based semantic validation and presents the evaluated results within a user-friendly interface where users can easily locate and observe the SHACL violations directly on the 3DCM visualization. The application aims to make this class of validation report interpretable to users without RDF/SHACL training; the present study evaluates whether this aim is met for CityJSON-based workflows.

To position CHEK VRV against existing tools, Table 1 contrasts the present work with five reference systems across five comparison axes: the validation engine used, whether both semantic and geometric checks are covered, whether validation results are integrated with a 3D visualization, whether custom user-supplied profiles are supported, and whether the system is openly available.

Table 1. Comparison of CHEK VRV with existing 3D city models and BIM validation tools.

Tool/Framework	Validation Engine	Semantic + Geometric	3D Visual Integration	Custom Profile	Open Source
val3dity [4]	val3dity (geometric only)	Geometric only	None (report only)	No	Yes
CityGML Quality ADE [35]	Stores validation in data model	Both	Indirect (via host viewer)	No (fixed ADE)	Yes
iTwin4PP [37]	Rule-based (Melbourne LuRs)	Semantic only	Yes (Bentley iTwin)	No (LuR-specific)	No
Pauwels et al. [5] IFC viewer	SHACL on BIM/IFC	Both	Yes	Limited	Partial
Akın et al. [36] KG + SHACL	SHACL on CityJSON KG	Both; geometric invalidities are highlighted only	Geometric only	Partial	Partial
CHEK VRV (this work)	DCV (SHACL + val3dity)	Both	Yes—both error types on same scene	Yes (CHEK profiles + user upload)	Yes

The contribution of CHEK VRV relative to this comparison set is not the introduction of a new validation algorithm. It reuses val3dity and the DCV's SHACL processor, but the integration of both error classes on a single open, web-based 3D viewer, with support for arbitrary user-supplied SHACL profiles.

3. Methodology

This section delineates the approach for developing and implementing the CHEK VRV, outlining the architectural components, data flows, validation processes, and the details of the test data used. Furthermore, a SWOT analysis was conducted to evaluate user feedback on the CHEK VRV, offering a comprehensive perspective on its practical implementation and future development prospects, as detailed in Section 3.3.

3.1. Test Data

To effectively test the functionality of the DCV service and the CHEK VRV as well, a carefully curated selection of 3D city models is utilized. The primary format for this test data consists of CityJSON datasets, chosen for their compact structure and suitability for web-based applications, which encode the CityGML data model [39]. The current document outlines that the CHEK Validation Results Viewer is a Flask-based application that validates CityJSON models against SHACL constraints using the OGC Data Completeness Validator and CHEK Building Block Profiles.

The dataset contains the definitions and rulesets of the validation profiles created for the CHEK project, as well as sample data, including:

- Profiles for the cities of Ascoli Piceno, Lisbon, Vilanova de Gaia, and Prague.
- Sample data used for the pilot case of Ascoli Piceno, provided by the municipality.
- Sample data for the pilot case in Lisbon, acquired specifically for the CHEK project.

The test datasets are designed to encompass a variety of scenarios pertinent to 3D city model validation, focusing on both geometric and semantic compliance [40]. They include:

- Real-world 3D City Models: Subsets of actual urban environments are frequently employed to ensure the practical relevance and applicability of the validation processes. These datasets can be sourced from various open data initiatives and may undergo conversion from CityGML to CityJSON for use with the CHEK VRV. The sample data described above, specifically for Ascoli Piceno and Lisbon, are examples of such real-world datasets used in the CHEK project. Academic sources further discuss the integration of CityGML and CityJSON models in projects like CHEK for developing suitable profiles and interoperable analysis software [41].
- Models with Known Geometric Irregularities: To test geometric validation capabilities, datasets include objects with common geometric and topological errors that tools like val3dity are designed to detect. This ensures the viewer can accurately pinpoint and visualize such issues. Academic discussions around CityGML software support and benchmarks, such as the GeoBIM benchmark, also highlight problems related to the implementation of the standard and the use of standardized data, which necessitate such validation efforts.
- Models Exhibiting SHACL Constraint Violations: Crucially, the test data incorporates CityJSON models that intentionally violate specific SHACL constraints defined in the OGC Data Completeness Validator and CHEK Building Block Profiles. This allows for rigorous evaluation of the semantic validation process and the viewer's ability to clearly present these violations within the 3D environment, as highlighted in the current document. Such violations might include missing attributes, incorrect data types, or non-conforming relationships, which are critical for demonstrating the interpretability features of the CHEK VRV. The various city profiles mentioned in the repository, such as

those for Ascoli Piceno, Lisbon, Vilanova de Gaia, and Prague, represent distinct sets of SHACL constraints that would be used to test for such violations.

Used Dataset Characterization

While validation profiles have been developed for all four municipalities, the availability of openly licensed and redistributable sample data is currently limited to the Lisbon pilot case, whose dataset was acquired specifically for the CHEK project under terms that permit public sharing. The datasets for the remaining three cities, though used internally during project demonstrations, are subject to municipal data governance restrictions that preclude their open publication. Consequently, the Lisbon model and its synthetically augmented counterpart serve as the primary and reproducible test case throughout this study, offering the most complete basis for transparent benchmarking of the CHEK VRV.

The Lisbon real-world model contains 44 city objects across five feature types: 8 LOD2 Building, 7 BuildingInstallation, 27 BuildingStorey, 1 OtherConstruction, and 1 Road, represented by 64 MultiSurface geometric primitives. Geometric validation against val3dity reveals 104 detected errors, all classified as RING_SELF_INTERSECTION, affecting 2 of the 64 MultiSurface primitives. At the feature level, 1 Building and 1 BuildingStorey instance are flagged as geometrically invalid, while all remaining features pass geometric checks. The synthetic counterpart reproduces this urban subset identically in structure but supplements it with deliberately injected attribute-level SHACL constraint violations. The synthetic error set covers two principal violation classes: (i) out-of-range attribute values, such as a legalHeight value exceeding the municipally defined limit; and (ii) cardinality violations, such as a building featuring multiple mainEntrance instances where the profile mandates exactly one. This controlled injection strategy ensures that each violation is precisely traceable to a specific SHACL constraint rule, enabling deterministic ground-truth evaluation of the CHEK VRV's detection and visualization capabilities.

Taken together, the paired real-world and synthetic Lisbon models provide complementary coverage of the two validation dimensions central to the CHEK VRV: geometric errors surfaced by val3dity and semantic violations surfaced by SHACL profile conformance checking. The synthetic augmentation widens the violation coverage of the Lisbon model, but it does not widen its geometric or semantic envelope. The four city-specific SHACL profiles, covering Ascoli Piceno, Lisbon, Vilanova de Gaia, and Prague, are all fully operational and in the exact same format within the CHEK VRV. The exclusive use of Lisbon data in this study reflects open data availability constraints rather than any restriction of the validation framework itself. The usability results reported here are specific to this 44-object model. While the remaining datasets differ only in building morphology and attribute-level profile constraints, the validation engine processed them without issue during internal project demonstrations, behaving consistently with the Lisbon case.

3.2. Software Description

The CHEK VRV is built as a Flask-based [42] web application that brings together server-side processing and client-side visualization to make the validation of 3D city models user-friendly. Using Flask as the core framework, the application connects to the DCV service through a set of RESTful endpoints for tasks like model submission, validation execution, and retrieving SHACL-based results. By separating the backend logic from the interactive front-end interface, the system enables users to upload CityJSON models, run validations, and instantly explore the results within a 3D viewer in real-time. The following sections break down the software architecture and components.

https://doi.org/10.3390/ijgi15070282

Software Architecture

Figure 1 shows the architecture, and Figure 2 shows the user interface of the CHEK VRV. Users upload CityJSON files through the "Choose File" button in the CityJSON File section (object 3 in Figure 2). After the file upload, users get the available profiles from the DCV service using the "Get Available Building Profiles" button, then select a profile from the dropdown list below the button. (objects 1 and 2 in Figure 2). Additionally, users can upload their own custom profiles using the "Upload Custom Profile" button (object 5 in Figure 2).

Through the "Run Validation" button, the application sends the selected profile ID and the uploaded model to the DCV service (object 6 in Figure 2). Users can view the uploaded CityJSON model in the bottom right section of the interface, through the "View Model" button (objects 4 and 10 in Figure 2). Next, after the application receives and parses the validation results, the geometric validation results with validation messages from val3dity are listed in the bottom left section of the interface (object 8 in Figure 2).

Users can highlight any invalid object from the list on the 3D model view by clicking on primitives in the list and observing the geometric invalidities at the place. The semantic results, completeness invalidities, and violation messages from the SHACL-based validation report are listed in the middle-left section (object 7 in Figure 2). If the required attribute is completely missing in the data, the message in the middle right section says, "Attribute can not be found". Otherwise, if the attribute is out of the range defined by the profile, the message shows both the declared and the desired values (object 9 in Figure 2).

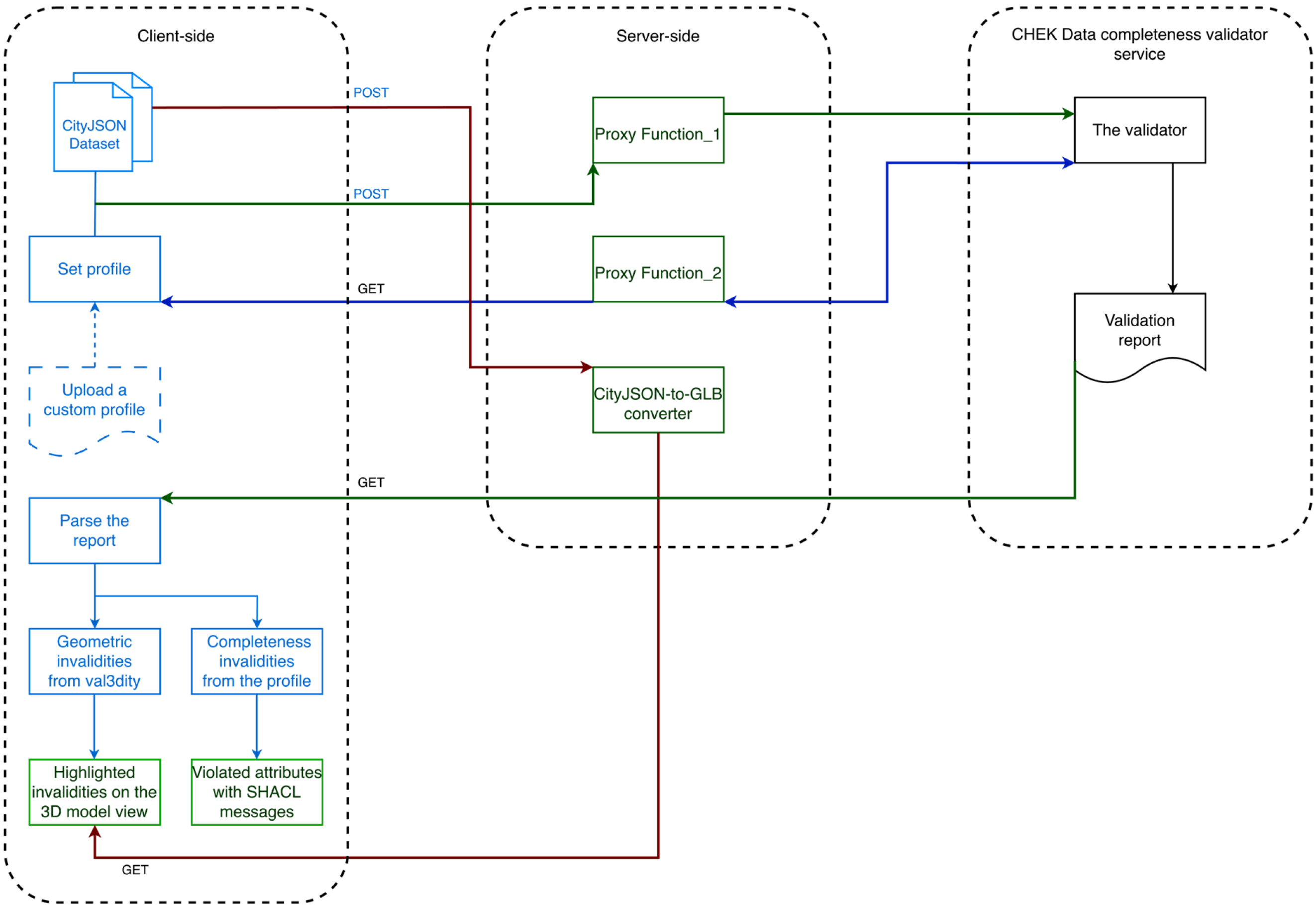

Figure 1. The architecture of the CHEK VRV. Each different colored arrow set stands for a specific content transfer. The red arrow follows the visualization traffic, which ends with GLB fetching. The green arrow stands for the validation process. The blue arrow stands for the SHACL profile selection from the DCV service.

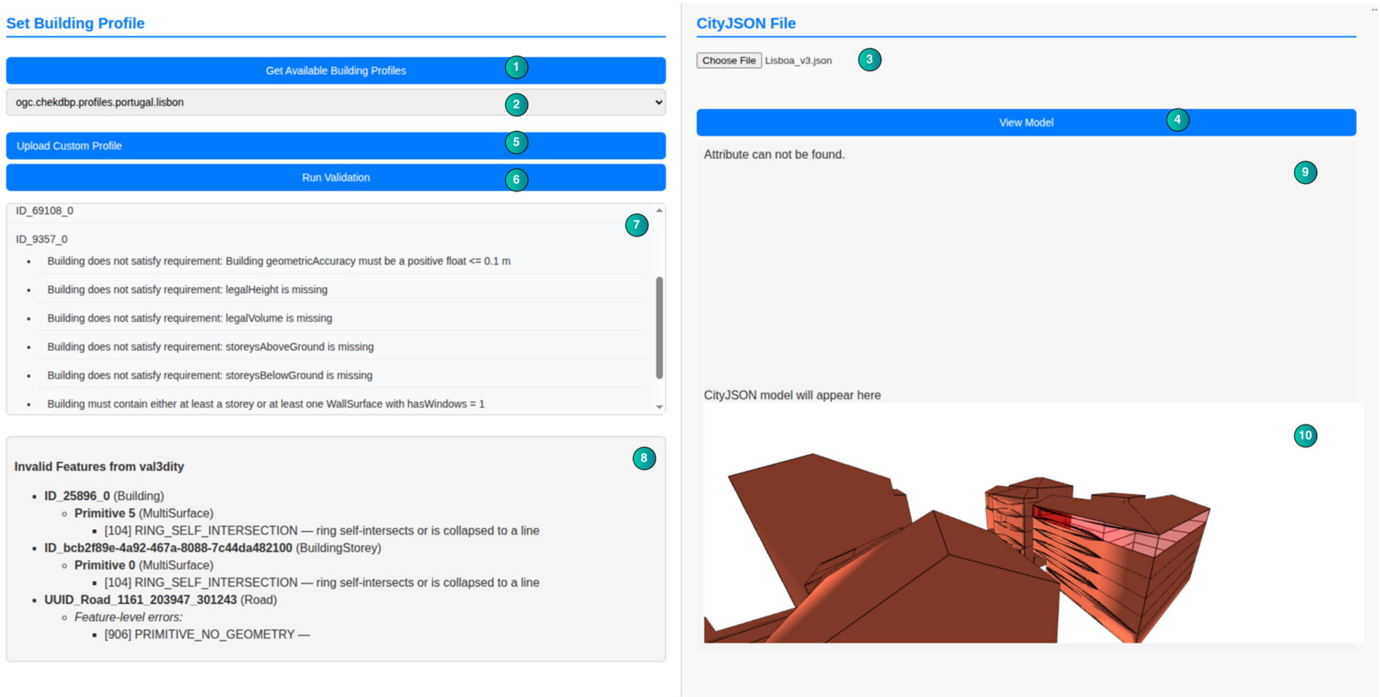

Figure 2. The user interface of the CHEK VRV.

The application repository has a descriptive step-by-step README to enable easy deployments on any servers. The application is online at https://chekvalidityviewer.tudelft.nl/3dinteractivevalidation/ (Latest access: 21 June 2026). In addition, the README file includes an endpoint mapping table for any reader who wants to deploy or extend the system without reading the source.

3.3. User Experience Evaluation

To assess the efficacy and usability of the CHEK VRV, a user experience evaluation was conducted as a formative usability study, focusing on factors such as intuitive interface design, clarity of validation feedback, and overall efficiency in identifying and resolving model discrepancies. User feedback was collected through an online questionnaire, and the responses were analyzed quantitatively (means and distributions of the closed-response items) and qualitatively (inductive coding of the open-ended responses). SWOT framing is used in the discussion (Section 5.3) as a supporting tool for translating findings into a development roadmap, to obtain insights into the platform's strengths, weaknesses, opportunities for enhancement, and potential threats to widespread adoption [43].

Participants and Questions

Twelve participants were recruited as a convenience sample from the CHEK project's network of stakeholders and adjacent practitioners. The sample composition (six academics/researchers, three data vendors/3D modelers, two urban planners/municipal officers, one student) reflects the intended user population for the tool but is not statistically representative. The goal is to expose specific usability and interpretability issues that can guide development, not to make point estimates of population-level usability scores.

The instrument was an online questionnaire (28 items) covering the five categories summarized in Table 2.

Participants completed the questionnaire after a guided walk-through of the tool using a CHEK-supplied sample dataset (Lisbon). The walk-through took approximately 5 min; total response time, including the questionnaire, averaged 7 min per person.

Table 2. Questionnaire structure.

Category	Items	Response Scale
Consent and demographics	3	Binary/multiple choice
Technical proficiency (self-report)	5	1–5 Likert
Model ingestion and profile selection (ease)	3	1–5 Likert (very difficult → very easy)
Validation execution, navigation, structural logic	3	1–5 Likert
Usability and error-message interpretability	7	1–5 Likert and binary
Cognitive load and comparison with raw report	3	1–5 Likert (minimal → exhaustive effort)
Future-enhancement prioritization	3	1–3 ranking
Qualitative feedback	1	Open text

The demographic breakdown in Table 3 reveals a very clear user persona: the participants are experts in geometry, 3D visualization, and CAD authoring, but they are intermediates when it comes to backend SW technologies (RDF/OWL) and structured data validation. As exactly half of the users had never even seen a JSON-LD report before, it successfully serves a demographic that possesses deep spatial and architectural knowledge but requires an intuitive, visual 3D interface to bridge their gap in SW expertise.

Table 3. Participant demographics and technical proficiency.

Category	Metric	Value	Key Insights & Notes
Professional Roles	Academic/Researcher	50% (n = 6)	Strong mix of data producers and end-users, leaning heavily toward research and spatial data production.
	Data Vendor/3D Modeler	25% (n = 3)	
	Urban Planner/Municipal Officer	17% (n = 2)	
	Student	8% (n = 1)	
Sectoral Experience	Average/Median	6.17 Years/6.00 Years	Highly experienced pool. Urban Planners are the most senior (Average: 9 years), followed by Academics (6.5 years) and Data Vendors (5 years).
Technical Proficiency	CAD or 3D Visualization Software	4.67/5.0 (Expert)	Highest skillset, near-perfect scores across the board.
	3D City Modeling (CityJSON/CityGML)	3.83/5.0 (High)	Highly familiar with 3D city data structures.
	Validation Procedures	2.75/5.0 (Moderate)	Less prominent in their routines.
	Semantic Web Tech (RDF, OWL)	2.25/5.0 (Novice)	The group's weakest area (Data Modelers dropped to 2.0).
Prior Familiarity	Interacted with JSON-LD/Validation	50% Yes (n = 6) 50% No (n = 6)	Exactly half of the users had never seen a JSON-LD validation report before.
UX Implication	User Persona Profile	Expert: CAD & 3D Geometry Novice: Semantic Web & RDF	Validates the core mission of CHEK VRV: Bridging the gap for spatial experts who need visual interfaces to understand complex semantic validation.

4. Results

4.1. Application Demonstration

To illustrate the validation workflow, we report the outputs produced when the Lisbon sample dataset is validated against the Lisbon Building Block Profile. The profile's UML diagram is reproduced in Figure 3. Among its constraints, the Lisbon profile requires that every Building object carry attributes such as legalHeight, legalVolume, storeysBelowGround, and material descriptors for wall and roof surfaces, and that geometric accuracy meets a defined tolerance.

https://doi.org/10.3390/ijgi15070282

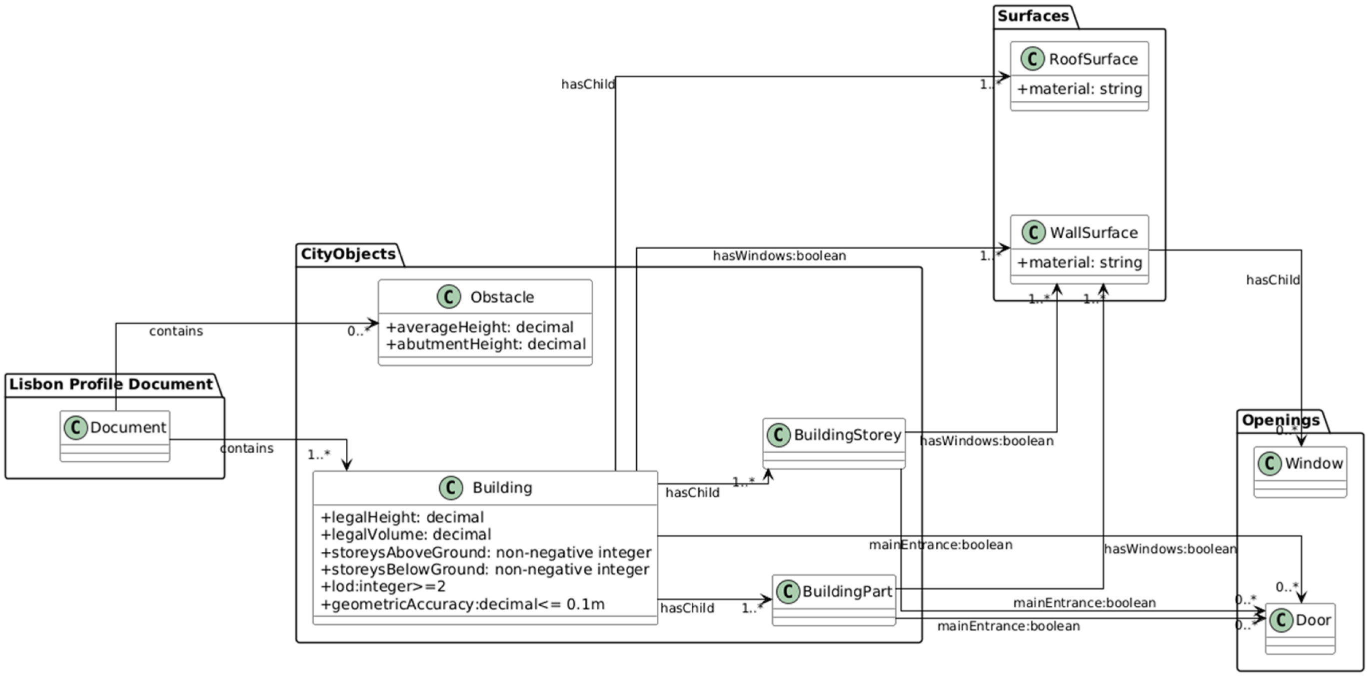

Figure 3. The UML diagram of the Lisbon profile.

Figure 4 shows the CHEK VRV interface following a complete validation run against building ID_25896_0. The left panel presents the full set of SHACL violations reported for this object, including missing legalHeight, missing legalVolume, missing storeysBelowGround, absent material attributes on WallSurface and RoofSurface, and a geometricAccuracy value that fails the profile threshold, alongside the val3dity geometric errors detected in the dataset, which include RING_SELF_INTERSECTION errors on a Building primitive and a BuildingStorey primitive, and a PRIMITIVE_NO_GEOMETRY error on the Road object. The right-hand attribute panel displays the attributes currently stored on the selected object (measuredHeight, roofType, storeysAboveGround), making it immediately apparent which required attributes are present and which are absent without requiring the user to consult the profile documentation separately. In the 3D viewport, the geometrically invalid surface is highlighted in red, spatially locating the error within the building's roof geometry.

The design intent visible in this figure is the co-location of three distinct information layers, semantic violations, geometric errors, and stored attribute values within a single screen state. A user can move from identifying that legalHeight is missing, to confirming which attributes the object does carry, to locating a geometric defect on the same roof, without navigating away from the viewer or cross-referencing external validation reports.

To support a clearer understanding of the functions described above, a narrated demonstration video with subtitles is provided alongside this paper. The application is publicly accessible, and the test data used in this study is available through the repository linked in Section 3.1. The video, the live application, and the test data can be accessed at the following links:

- Demonstration Video: https://github.com/tudelft3d/3dinteractivevalidation/blob/main/usage_video.gif (Latest access: 21 June 2026).
- CHEK VRV Application: https://chekvalidityviewer.tudelft.nl/3dinteractivevalidation/ (Latest access: 21 June 2026).
- Test Data Folder: https://github.com/tudelft3d/3dinteractivevalidation/tree/main/data (Latest access: 21 June 2026).

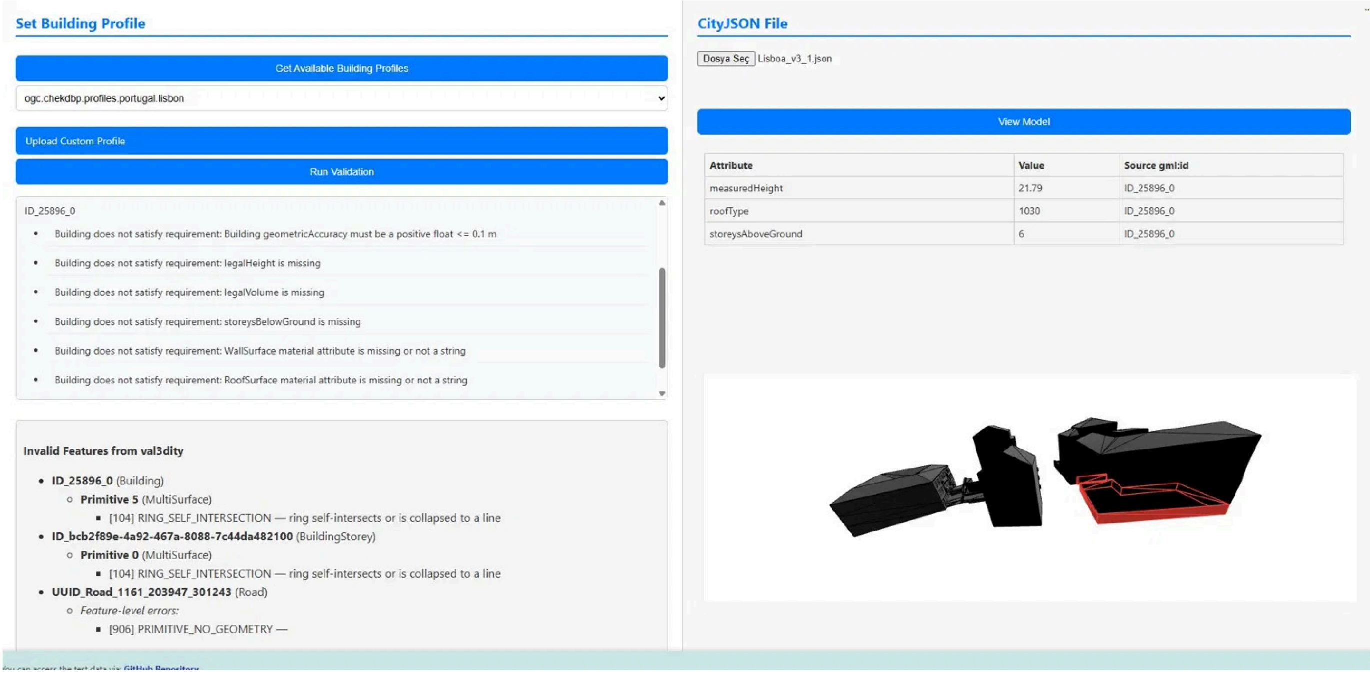

Figure 4. The snapshots of invalidity samples from the application interface.

4.2. Quantitative Survey Results

Table 4 summarizes the closed-response items. We report mean scores and the full distribution, where the latter is informative; for binary and multiple-choice items, we report counts.

Table 4. Closed-response survey results (n = 12).

Item	Mean (1–5)	Distribution/Note
Ease: uploading CityJSON	4.58	7 ranked as 5, 5 ranked as 4
Ease: fetching municipal profiles	4.42	—
Ease: uploading custom SHACL profile	3.92	Two participants scored 2
Visibility of system status (latency)	3.50	Lowest-scored UI item
3D viewer intuitiveness	4.92	11 ranked as 5, 1 ranked as 4
Structural logic of report hierarchy	4.17	—
Spatial mapping (semantic → geometric)	4.33	—
Mental effort: raw JSON-LD report	3.25	Higher = more effort
Mental effort: CHEK VRV viewer	1.50	Higher = more effort
Correction-workflow actionability	4.42	—
Highlighting granularity	—	9 sufficient, 3 require finer
Easier to locate (geometric vs. semantic)	—	9 geometric, 3 semantic
Diagnostic distinction (missing vs. mismatch)	—	11 excellent, 1 poor

The ranking exercise on future enhancements (1 = highest priority) returned: direct CityGML-to-GLB conversion (mean rank 1.75, ranked first by 7 of 12); semantic hierarchy preservation in GLB (mean rank 2.08, ranked first by 4 of 12); attribute-based styling (mean rank 2.17, ranked first by 1 of 12).

Three findings stand out, and we return to them in the discussion. (i) The 3D viewer scored at the ceiling of the scale (4.92)—within the constraints of self-report, this is as clear a signal as the instrument can deliver. (ii) The self-reported mental-effort gap between the raw JSON-LD report and the viewer is large (3.25 vs. 1.50), though we caution that this is a same-respondent comparison after a 25 min guided walk-through and so is not a clean A/B contrast. (iii) Geometric errors are easier to locate than semantic errors (9 of 12 participants).

4.3. Qualitative Results

The open-ended item attracted ten substantive responses (the other two were empty or "no comment"). We coded these inductively into four themes:

- Integration with authoring tools (n = six responses). Six of the ten respondents asked for live synchronization with native authoring tools—most frequently QGIS and Revit. One participant wrote: "Increasing the use of automated error correction tools, more detailed visual error reporting, and live synchronization with popular BIM/GIS software will significantly improve productivity."
- CityGML support (n = 4). Four respondents flagged CityGML as a blocker for their own workflows.
- Loading-state feedback (n = 3). Three respondents commented on the lack of progress indication during long validation runs—the same issue surfaced in the closed-response item on system-status visibility (3.50/5).
- Granularity (n = 3). Three respondents requested face-level highlighting rather than the current building/condominium-level highlighting.

5. Discussion

5.1. Interpretation of the Quantitative Findings

The 3D viewer received a near-ceiling intuitiveness score (4.92/5). Two factors plausibly account for this: the participant pool is heavily skilled in CAD and 3D visualization (mean self-rated proficiency 4.67/5), so a Three.js-based viewer is in their visual idiom; and the walk-through used a small (8-building) dataset on which the viewer is responsive. We would not extrapolate this score to other user populations or larger models without further evidence.

The mental-effort gap between the raw report and the viewer is the largest single contrast in the data. It should be noted that this is a same-respondent self-report comparison made immediately after a guided walk-through, with all the ordering and demand-characteristic risks that imply, is acknowledged here and not advertised away. What we are willing to claim is that twelve practitioners with non-trivial domain expertise reported a substantial preference for the visual presentation. A controlled time-on-task study is needed before that preference is converted into a performance claim.

Among the user study findings, the preference for geometric over semantic error localization was perhaps the most noteworthy, as it reveals a discoverability asymmetry that the current interface does not fully resolve; this observation is examined in detail in Section 5.2.

5.2. Why Are Semantic Errors Harder to Visualize?

In a geometric error, the referent of the error is a region of space (a self-intersecting solid, a non-planar surface, etc.), and the natural visual response is to mark that region. In a missing-attribute error, the referent is not a region of space at all: it is a tuple (object, schema property) where the property has no value to display. The viewer can do no better than highlight the object that lacks the property and rely on text to communicate what is missing.

This was visible in the qualitative responses. Two participants explicitly described missing-attribute errors as "harder to see" precisely because there was nothing to see: the building looked, in the 3D scene, exactly like every other building. The mismatch case is in between: there is a value to compare against the required range, so the side-panel can show the comparison, which is presumably why the diagnostic-distinction item scored 11 of 12.

This is an inherent property of attribute-completeness validation, not a defect of the present implementation. We see two avenues for partial remediation: thematic coloring of buildings by which attributes they carry (so the absence is visible in aggregate), and an attribute-completeness summary view that operates on the model as a whole rather than per object. Both are out of scope for the present paper at this time.

5.3. SWOT-Framed Synthesis

- Strengths: The 3D viewer's intuitiveness (4.92/5), the integration of geometric and semantic results in a single view, and the clarity of the missing-vs-mismatch diagnostic distinction (11 of 12) are the three findings on which the tool's current form is on firm ground.
- Weaknesses: The lowest-scored item was visibility of system status during validation (3.50/5), confirmed by qualitative comments on loading feedback. Semantic-error discoverability is the second area where the tool is materially weaker than it is for geometric errors.
- Opportunities: The ranking exercise (Section 4.2) and the qualitative responses (Section 4.3) converge on these near-term development priorities, in order: direct CityGML support, semantic hierarchy preservation in GLB, attribute-based styling, and live synchronization with authoring GIS/BIM tools.
- Threats: The dependence on the DCV remote service means that DCV downtime or API changes propagate to the tool; this is an operational risk inherited from the architecture rather than a flaw in the implementation and is the main reason the repository pins specific DCV versions. In addition, as noted in qualitative feedback, ensuring validation continuity and stability, specifically through "dynamic data" updates, poses a persistent challenge against static validation workflows.

5.4. Scalability and LOD Considerations

The present implementation has been profiled only on the test datasets reported in Section 3.1 (up to 412 buildings at LOD1.3). On the development hardware (a mid-range laptop with 24 GB RAM and integrated graphics), the CityJSON-to-GLB conversion is currently a bottleneck and is memory-bound rather than CPU-bound: for an 8-building model, the conversion completes in approximately 1 s. For a whole city-scale model (thousands of buildings) on the same hardware, the Three.js renderer could be unresponsive to interaction.

Two specific changes would be required to push the tool into the thousand-plus-building regime. First, the GLB conversion would need to be chunked, either by tiling or by spatial index, so that the working set is bounded independently of model size, and the converted chunks would need to be streamed to the front end progressively. Second, the renderer would need frustum-culling and level-of-detail switching, so that geometry outside the current view is not rendered at full resolution. Neither change is conceptually difficult, but both are non-trivial engineering tasks in terms of upcoming update priority.

On LOD specifically, the present implementation has been tested on a LOD2 dataset only, which reflects the scope of the openly shareable test data currently available. However, this does not constitute a functional limitation of the CHEK VRV with respect to higher LOD inputs. The attribute completeness rules defined in the CHEK profiles already specify constraints up to LOD4, including rules governing openings and interior structures that are only semantically meaningful at LOD3 and above; the validator will evaluate these rules against any uploaded model regardless of its LOD. Furthermore, the error highlighting mechanism operates on object identifiers rather than on geometry type or LOD level. When a user clicks a reported violation, the viewer resolves the corresponding city object by its ID and highlights it in the 3D scene. A LOD3 or LOD4 object carrying a constraint violation would therefore be located and highlighted in exactly the same way as a LOD2 object. The gap between the tested LOD range and the full profile rule set is therefore a data availability constraint on the test suite, not an architectural limitation. As LOD3 and LOD4 compliant datasets become openly available, they can be ingested and validated without modification to the viewer or the validation pipeline.

5.5. Where the Tool Works Best and Where It Does Not

The user-study results, combined with our own experience using the tool during development, indicate the following pattern.

- Where it works best: Small-to-medium models with a mix of geometric errors and SHACL mismatch violations on attributes with bounded ranges. In this regime, every class of finding has a clear visual referent, geometric errors highlight the offending surface, mismatches are accompanied by the declared-vs-required side-panel, and the rendering is fast enough that interaction is fluid.
- Where it works less well: Three scenarios. (i) Large models (thousands of buildings), covered in Section 5.4. (ii) Many violations of the same type, where the list of violations becomes unmanageable, and the user needs an aggregate summary rather than a per-violation drill-down; the tool currently does not provide such a summary. (iii) Missing-attribute violations on attributes that have no obvious spatial referent (e.g., a missing administrative-document link), covered in Section 5.2.

5.6. Limitations

The current implementation has not been profiled beyond the available test datasets. The CityJSON-to-GLB conversion is memory-bound, and the Three.js renderer is expected to become unresponsive on interaction for city-scale models with thousands of buildings on mid-range hardware. Chunked conversion, progressive streaming, and frustum-culling are identified as necessary engineering steps before the tool can be considered suitable for large-scale deployments. Additionally, the lowest-rated item in the user study was visibility of system status during validation, with qualitative comments specifically noting insufficient loading feedback. This is a known interface deficiency that has not yet been addressed.

6. Conclusions and Future Work

This paper has described the CHEK VRV, a Flask-based web application that integrates the OGC DCV with an interactive Three.js viewer so that SHACL semantic violations and val3dity geometric errors are resolved against the same 3D visual representation of a CityJSON model. We have reported a formative usability study with twelve domain experts, in which the 3D viewer received a near-ceiling intuitiveness score and the participants self-reported a substantial preference for the visual presentation over the raw JSON-LD validation report.

The application has been demonstrated on the CHEK project's profiles. Its applicability to other profiles depends on whether the source data is available in or convertible to CityJSON.

The ranking exercise in the user study indicates that direct CityGML support is the most-requested next step, consistent with the continued prevalence of CityGML in municipal workflows [44]. The DCV naturally supports CityGML now, so there is a need for a CityGML-to-GLB or a CityGML-CityJSON-GLB conversion path for enabling visualization if the CityGML is chosen for input. The second priority is semantic hierarchy preservation through the GLB conversion, so that thematic visualization and attribute-based styling can be supported.

CHEK VRV does not introduce new validation algorithms. Its contribution is in the integration of existing OGC and W3C components with a 3D viewer, and in the user study that maps the resulting tool's strengths and weaknesses onto a concrete development roadmap.

Author Contributions: Conceptualization: Alper Tunga Akın, Alejandro Villar, Jantien Stoter and Francesca Noardo; Methodology: Alper Tunga Akın and Alejandro Villar; Software: Alper Tunga Akın and Alejandro Villar; Validation: Alper Tunga Akın and Alejandro Villar; Formal Analysis: Alper Tunga Akın and Alejandro Villar; Investigation: Alper Tunga Akın, Alejandro Villar and

Abdoulaye Diakite; Resources: Siham El Yamani, Jantien Stoter, Francesca Noardo, Robert Atkinson and Piotr Zaborowski; Data Curation: Alper Tunga Akın, Alejandro Villar and Siham El Yamani; Writing—original draft preparation: Alper Tunga Akın and Alejandro Villar; Writing—review and editing: Alper Tunga Akın, Alejandro Villar, Abdoulaye Diakite, Siham El Yamani, Jantien Stoter, Francesca Noardo, Robert Atkinson and Piotr Zaborowski; Visualization: Alper Tunga Akın and Alejandro Villar; Supervision: Siham El Yamani, Jantien Stoter, Francesca Noardo, Robert Atkinson and Piotr Zaborowski; Project Administration: Jantien Stoter, Siham El Yamani, Francesca Noardo, Robert Atkinson and Piotr Zaborowski; Funding Acquisition: Siham El Yamani, Jantien Stoter, Francesca Noardo, Robert Atkinson and Piotr Zaborowski All authors have read and agreed to the published version of the manuscript.

Funding: The work described in this paper has been co-funded by the European Union and the United Kingdom under the Horizon Europe CHEK—Change Toolkit for Digital Building Permits project [28] (Grant Agreement No. 101058559).

Institutional Review Board Statement: Ethical review and approval were not required for this study, as it involved an anonymous, minimal-risk User Experience Evaluation of a software interface. No personally identifiable information (PII) or sensitive personal data was collected.

Informed Consent Statement: All participants provided explicit digital informed consent before their participation, confirming that they understood the purpose of the study and voluntarily agreed to participate.

Data Availability Statement: The repository of the study is shared at: https://github.com/tudelft3d/3dinteractivevalidation (Latest access: 21 June 2026).

Conflicts of Interest: Author Abdoulaye Diakite was employed by the company CityGeometrix Pty Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Biljecki, F.; Stoter, J.; Ledoux, H.; Zlatanova, S.; Çöltekin, A. Applications of 3D City Models: State of the Art Review. *ISPRS Int. J. Geo-Inf.* **2015**, *4*, 2842–2889. [CrossRef]
2. Alam, N.; Wagner, D.; Wewetzer, M.; von Falkenhausen, J.; Coors, V.; Pries, M. Towards Automatic Validation and Healing of CityGML Models for Geometric and Semantic Consistency. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *II-2/W1*, 1–6. [CrossRef]
3. Krämer, M.; Haist, J.; Reitz, T. Methods for Spatial Data Quality of 3D City Models. In Proceedings of the Eurographics Italian Chapter Conference, Trento, Italy, 14–16 February 2007; pp. 167–172. [CrossRef] [PubMed]
4. Ledoux, H. val3dity: Validation of 3D GIS Primitives According to the International Standards. *Open Geospat. Data Softw. Stand.* **2018**, *3*, 1. [CrossRef]
5. Pauwels, P.; van den Bersselaar, E.; Verhelst, L. Validation of Technical Requirements for a BIM Model Using Semantic Web Technologies. *Adv. Eng. Inform.* **2024**, *60*, 102426. [CrossRef]
6. *ISO 19107:2003*; Geographic Information—Spatial Schema. International Organization for Standardization: Geneva, Switzerland, 2003.
7. Vinasco-Alvarez, D.; Samuel, J.; Servigne, S.; Gesquière, G. Towards Limiting Semantic Data Loss in 4D Urban Data Semantic Graph Generation. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *VIII-4/W2-2021*, 37–44. [CrossRef]
8. Nys, G.-A.; Billen, R. From Consistency to Flexibility: Handling Spatial Information Schema Thanks to a Middleware in a 3D City Modeling Context. *Trans. GIS* **2022**, *27*, 115–133. [CrossRef]
9. Hagedorn, P.; Pauwels, P.; König, M. Semantic Rule Checking of Cross-Domain Building Data in Information Containers for Linked Document Delivery Using the Shapes Constraint Language. *Autom. Constr.* **2023**, *156*, 105106. [CrossRef]
10. Pauwels, P.; Zhang, S.; Lee, Y.-C. Semantic Web Technologies in the AEC Industry: A Literature Overview. *Autom. Constr.* **2016**, *73*, 145–165. [CrossRef]
11. Shapes Constraint Language (SHACL). Available online: https://www.w3.org/TR/shacl/ (accessed on 3 June 2025).
12. Coors, V.; Betz, M.; Duminil, É. A Concept of Quality Management of 3D City Models Supporting Application-Specific Requirements. *PFG–J. Photogramm. Remote Sens. Geoinf. Sci.* **2020**, *88*, 3–14. [CrossRef]
13. Nielsen, J.; Landauer, T.K. A Mathematical Model of the Finding of Usability Problems. In Proceedings of the INTERACT '93 and CHI '93 Conference, Amsterdam, The Netherlands, 24–29 April 1993; pp. 206–213. [CrossRef]
14. Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web. *Sci. Am.* **2001**, *284*, 34–43. [CrossRef]

15. RDF 1.1 Primer. Available online: https://www.w3.org/TR/rdf11-primer/ (accessed on 3 June 2025).
16. Hitzler, P.; Krötzsch, M.; Rudolph, S. *Foundations of Semantic Web Technologies*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2009. [CrossRef]
17. Ding, L.; Xiao, G.; Pano, A.; Fumagalli, M.; Chen, D.; Feng, Y.; Calvanese, D.; Fan, H.; Meng, L. Integrating 3D City Data through Knowledge Graphs. *Geo-Spat. Inf. Sci.* **2024**, *28*, 780–799. [CrossRef]
18. van den Brink, L.; Janssen, P.; Quak, W.; Stoter, J. Linking Spatial Data: Automated Conversion of Geo-Information Models and GML Data to RDF. *Int. J. Spat. Data Infrastruct. Res.* **2014**, *9*, 59–85. [CrossRef]
19. Cimmino, A.; Fernández-Izquierdo, A.; García-Castro, R. Astrea: Automatic Generation of SHACL Shapes from Ontologies. In *The Semantic Web*; Harth, A., Kirrane, S., Ngonga Ngomo, A.-C., Paulheim, H., Rula, A., Gentile, A.L., Haase, P., Cochez, M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; p. 497. [CrossRef]
20. Pareti, P.; Konstantinidis, G. A Review of SHACL: From Data Validation to Schema Reasoning for RDF Graphs. *arXiv* **2021**, arXiv:2112.01441.
21. Calbimonte, J.-P.; Ciortea, A.; Kampik, T.; Mayer, S.; Payne, T.R.; Tamma, V.; Zimmermann, A. Autonomy in the Age of Knowledge Graphs: Vision and Challenges. *Trans. Graph Data Knowl.* **2023**, *1*, 13:1–13:22. [CrossRef]
22. Kücükavci, A. A Data-Centric Framework for HVAC Information Management: To Facilitate HVAC Rightsizing and Compliance Checking. Ph.D. Thesis, Aarhus University, Aarhus, Denmark, 2024.
23. Anim, J.K.; Robaldo, L.; Wyner, A. A SHACL-Based Approach for Enhancing Automated Compliance Checking with RDF Data. *Information* **2024**, *15*, 759. [CrossRef]
24. Fauth, J.; Gade, P.N.; Kaiser, S.B.D.; Raj, K.; Pedersen, J.G.; Olsson, P.; Nisbet, N.; Ventura, S.M.; Hirvensalo, A.; Granja, J.; et al. Investigating Building Permit Processes across Europe: Characteristics and Patterns. *Build. Res. Inf.* **2024**, *53*, 417–434. [CrossRef]
25. Noardo, F.; Malacarne, G.; Mastrolembo Ventura, S.; Tagliabue, L.C.; Ciribini, A.L.C.; Ellul, C.; Guler, D.; Harrie, L.; Senger, L.; Waha, A.; et al. Integrating Expertises and Ambitions for Data-Driven Digital Building Permits—The EUnet4DBP. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *XLIV-4/W1-2020*, 103–110. [CrossRef]
26. Hagedorn, P.; Fauth, J.; Zentgraf, S.; Seiß, S.; König, M.; Brilakis, I. OntoBPR: An Ontology-Based Framework for Performing Building Permit Reviews Using Standardized Information Containers. *Adv. Eng. Inform.* **2025**, *66*, 103369. [CrossRef]
27. Urban, H.; Fischer, S.; Schranz, C. Adapting to an OpenBIM Building Permit Process: A Case Study Using the Example of the City of Vienna. *Buildings* **2024**, *14*, 1135. [CrossRef]
28. Noardo, F.; Wu, T.; Ohori, K.A.; Krijnen, T.; Stoter, J. IFC Models for Semi-Automating Common Planning Checks for Building Permits. *Autom. Constr.* **2021**, *134*, 104097. [CrossRef]
29. Change Toolkit for Digital Building Permits. Available online: https://chekdbp.eu/ (accessed on 3 June 2025).
30. CHEK Profiles as OGC Building Blocks. Available online: https://github.com/ogcincubator/chek-profiles-bblocks (accessed on 3 June 2025).
31. Noardo, F.; Atkinson, R.; Simonis, I.; Villar, A.; Zaborowski, P. OGC Data Exchange Toolkit: Interoperable and Reusable 3D Data at the End of the OGC Rainbow. In *Recent Advances in 3D Geoinformation Science*; Kolbe, T.H., Donaubauer, A., Eds.; Lecture Notes in Geoinformation and Cartography; Springer: Cham, Switzerland, 2024; p. 761. [CrossRef]
32. The Profiles Vocabulary. Available online: https://www.w3.org/TR/dx-prof/ (accessed on 3 June 2025).
33. OGC Specification Building Blocks. Available online: https://ogcincubator.github.io/bblocks-docs/ (accessed on 3 June 2025).
34. Miranda, F.; Ortner, T.; Moreira, G.; Hosseini, M.; Vuckovic, M.; Biljecki, F.; Silva, C.; Lage, M.; Ferreira, N. The State of the Art in Visual Analytics for 3D Urban Data. *Comput. Graph. Forum* **2024**, *43*, e15112. [CrossRef]
35. Shen, S.; Lu, S.; Shen, L.; Sheng, Z.; Tang, N.; Luo, Y. Ask Humans or AI? Exploring Their Roles in Visualization Troubleshooting. *arXiv* **2024**, arXiv:2412.07673.
36. Betz, M.; Coors, V. An Application Domain Extension for Storing Validation Results of CityGML Structures. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *VIII-4/W1-2021*, 11–16. [CrossRef]
37. Akın, A.T.; Usta, Z.; Stoter, J.; Ohori, K.A.; Cömert, Ç. Leveraging Knowledge Graphs and Semantic Web Technologies for Validating 3D City Models. *Int. J. Geogr. Inf. Sci.* **2026**, *40*, 1867–1893. [CrossRef]
38. Emamgholian, S.; Pouliot, J.; Shojaei, D.; Losier, L.-M. A Web-Based Planning Permit Assessment Prototype: ITwin4PP. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *XLVIII-4/W4-2022*, 37–44. [CrossRef]
39. Ledoux, H.; Ohori, K.A.; Kumar, K.; Dukai, B.; Labetski, A.; Vitalis, S. CityJSON: A Compact and Easy-to-Use Encoding of the CityGML Data Model. *Open Geospat. Data Softw. Stand.* **2019**, *4*, 4. [CrossRef]
40. Villar, A.; Akın, A.T. *CHEK 3D City Model Data Requirements, Profiles, and Sample Data, V1.0*; Zenodo: Brussels, Belgium, 2025. [CrossRef]
41. Atkinson, R.; Zaborowski, P.; Noardo, F.; Simonis, I. Smart Cities—Systems of Systems Interoperability and OGC Enablers. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *X-4/W3-2022*, 19–26. [CrossRef]
42. Flask. Available online: https://flask.palletsprojects.com/en/stable/ (accessed on 3 June 2025).

43. Uzun, B.; Yıldırım, V.; Çoruhlu, Y.E.; Yıldız, O.; Terzi, F.; Atasoy, B.A. Enhancing Turkish Land Readjustment via a Combination of IPA-Based on SWOT and Workshops. *Land Use Policy* **2024**, *144*, 107245. [CrossRef]
44. Vinasco-Alvarez, D.; Samuel, J.; Servigne, S.; Gesquière, G. Leveraging Rules on Integrated 3D Semantic City Models for Validating Scenarios of Urban Evolution. *Data-Centric Eng.* **2025**, *6*, e47. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.