A Glimpse of the CityGML Scenario Application Domain Extension

Frederick Auer[1], Camilo León-Sánchez[2], Giorgio Agugiaro[3]

[1] MSc in Geomatics, Delft University of Technology, Delft, The Netherlands – f.j.auer@student.tudelft.nl
[2] Laboratory of Geo-information Science and Remote Sensing,
Wageningen University & Research, Wageningen, The Netherlands – camilo.leonsanchez@wur.nl
[3] 3D Geoinformation Group, Department of Urbanism, Faculty of Architecture and Built Environment,
Delft University of Technology, Delft, The Netherlands –g.agugiaro@tudelft.nl

Keywords: CityGML, ADE, Urban Digital Twins, Scenarios management.

Abstract
This paper presents the Scenario Application Domain Extension (Scenario ADE) for CityGML 2.0 and the rationale behind its development. CityGML 2.0 and its ADEs are primarily designed to represent one coherent state of an urban environment. No concurrent hypothetical alternatives are supported, nor are simulation results connected to the precise city-model state, engine, settings and boundary conditions that produced them. In this work, scenario information is decomposed into three pillars: Context, Strategy and Configuration. These pillars are translated into design principles and used to evaluate the proposed data model for the Scenario ADE. The Scenario ADE is available in two encodings: XML and SQL, the latter for the 3DCityDB relational database. Results show that the current version can represent parallel scenario branches in a unified approach, avoid unnecessary duplication of data, preserve object and attribute lineage, and bind simulation outputs to their originating features and configurations. Our findings, based on energy simulations carried out in Rotterdam, show confirm that the Scenario ADE can act as a domain-agnostic scenario-management and lightweight provenance layer for CityGML 2.0, although its treatment of context, attribute semantics and non-simulation provenance requires further refinement.

1. Introduction

Global urbanisation and climate-change mitigation are increasing the need for precise urban energy planning. Cities already account for a substantial share of worldwide energy use and greenhouse-gas emissions, while electricity demand is expected to grow further because of electrification and increasing cooling needs. Urban areas must evaluate not only how buildings perform today, but how the same building stock may perform under warmer climates, refurbishment programmes, photovoltaic adoption or additional urban vegetation.

Semantic 3D city models are increasingly adopted as information hubs that support such analyses. The CityGML standard allows the representation of buildings, vegetation, transportation objects and other urban entities through a modular conceptual model. Domain-specific extensions can be introduced through Application Domain Extensions (ADEs). The Energy ADE (Agugiaro and Padsala, 2025), for example, adds energy-related concepts such as thermal zones, construction properties, schedules and time series. The combination of CityGML and the Energy ADE describes what an urban environment is and which energy-related properties it contains. However, it cannot describe multiple alternative futures of that same environment. As a matter of fact, planning questions are comparative by nature: A decision maker may need to compare a present-day neighbourhood with a future-climate case, a refurbishment case, a photovoltaic case or an urban-greening case. If each alternative is stored as an independent city model, the baseline data (geometry and attributes) are duplicated. The relationship among alternatives becomes implicit, and the inputs and assumptions behind individual results are easily lost. The challenge is therefore not only to represent changes, but also to preserve the reason for that change and the computational process that produced a result. For example, a heating-demand value has limited interpretive value if the user cannot identify the building to which it belongs, the model state that was simulated, the weather file, the radiation model, the simulation engine and the settings used. The Scenario ADE is intended to provide this missing management layer. It should allow multiple urban futures to coexist within one dataset and should connect each scenario to its assumptions, interventions, configurations and outputs.

1.1 Defining a scenario

The term "scenario" is used across several traditions and does not have one universally accepted definition. Scenario practice developed in military planning and later in corporate strategic planning. The literature commonly distinguishes the Intuitive Logics school, the French "La Prospective" school (Godet, 1986) and the Probabilistic Modified Trends school (Amer *et al.*, 2018). The traditions differ in their emphasis on narrative plausibility, preferred futures, quantitative trends and probability. Urban planning similarly uses predictive, normative and exploratory scenarios, often in hybrid combinations (Bradfield *et al.*, 2005). Spaniol and Rowland (2019) synthesise recurring characteristics from the Intuitive Logics literature. Scenarios a) concern a future purview shaped by external contextual factors, b) should form plausible and internally consistent narratives, and c) gain meaning as a differentiated set rather than as isolated descriptions. This synthesis is useful but cannot be applied too rigidly to urban modelling. Forecasts, back casting studies and baseline states may not satisfy every strict criterion, yet they remain operationally necessary when planners construct and compare alternatives.

For the purposes of the Scenario ADE, a scenario is treated as a structured description of a possible urban state and the conditions, actions and configurations associated with it. The emphasis lies on what must be stored so that scenarios can be compared and reproduced. We adopt the distinction made in

Exploratory Scenario Planning (XSP) (Börjeson *et al.*, 2006) between the contextual environment, over which planners have little direct control, and the transactional environment, in which actors can intervene. This distinction separates external assumptions from deliberate strategies.

In this work, we extend that two-part distinction with a third pillar: Configuration. Scenario-planning literature primarily concerns the logic and narrative of alternatives, but computational urban scenarios also depend on technical settings. A model run must therefore preserve not only the external context and the selected intervention, but also the software engine, calculation standard, input parameters and output results.

1.2 Problem definition

Our research addresses two structural limitations in multi-temporal urban-energy data management. The first one corresponds to the absence of robust scenario management in the CityGML 2.0 ecosystem as a dataset represents only one city state. It does not natively express parallel hypothetical alternatives or the lineage among them. In practice, variants are stored as separate files. This makes comparisons scattered and leads to substantial data duplication, even when two alternatives differ only in one weather parameter or a small group of changed attributes. The second limitation is the inadequate provenance for simulation outputs. For example, energy-related values may be attached to buildings or surfaces, but the data model does not necessarily bind them to the exact engine, version, settings and input conditions that produced them. When the same building participates in several scenarios, values stored directly on the building accumulate without a reliable indication of which scenario or run settings they belong to. These limitations lead to three challenges:

- How can the lifecycle of real-world and hypothetical city objects be recorded over time?
- How can parallel alternatives of the same city be kept consistent in one dataset without duplicating the complete model?
- How can each result be linked to the specific feature, model state, engine and parameters that produced it?

The three questions reduce to one underlying requirement: a mechanism that tracks not only what has changed, but why it has changed and under which assumptions or processes.

2. Related work

A first, preliminary version of the Scenario ADE was developed at the Austrian Institute of Technology (Agugiaro, 2018). Subsequent developments took place within the European DigiTwins4PEDs project[1]. Still, the Scenario ADE which served as starting point for the development of the version presented in this paper was still rather in its early stages, in that it did not allow to reliably distinguish inputs from outputs or bind a result to the simulation settings that produced it. The assessment of the starting point version motivated a more general architecture, explicit separation of scenarios and simulations, better lineage, feature- and attribute-level change mechanisms, and a dedicated result structure.

2.1 CityGML 3.0

CityGML 3.0 has introduced mechanisms that are highly relevant to scenario management. Its *Versioning* module formalises versions, version transitions and transactions. The *Dynamizer* module addresses attributes that vary through time. These concepts provide important vocabulary, but neither mechanism alone closes the scenario and provenance gap. For better understanding and visual reference, the reader is encouraged to download the corresponding UML diagrams from the OGC's GitHub repository[2].

In the *Versioning module*, a CityGML 3.0 *Version* is a named snapshot of features. A *VersionTransition* links a predecessor and successor *Version* and carries transactions that convert one state into another. Transactions can insert, delete or replace features. *Transition* types can represent planned, realised or historical succession, as well as forks and merges. The module is inspired by version-control systems and supports branching alternatives in addition to linear change. CityGML 3.0 also distinguishes database time from real-world validity through lifecycle attributes. Stable object identity is maintained through identifiers that persist across object versions. This makes the module appropriate for both historical change and alternative planning states. However, the versioning module does not identify the simulation or analytical process responsible for a change or a result. It tracks states and transitions but does not bind them to an engine, parameter set or boundary condition. It also focuses mainly on feature-level transactions. Urban-energy scenarios frequently require attribute-level modifications, such as changing just one wall U-value or adding a calculated demand value without replacing an entire object. The Scenario ADE therefore adopts the structural vocabulary of the versioning module but extends it with scenario metadata and provenance. As the operational ecosystem is based on CityGML 2.0, the relevant versioning concepts are backported rather than used directly in CityGML 3.0.

The CityGML 3.0's *Dynamizer* module represents how a property changes over time. It references an attribute, defines a temporal interval and stores *Atomic* or *Composite Timeseries*. It can also connect to external sensors, making it suitable for observations and live urban data. This module does not represent branching alternatives. A time-varying property remains attached to one feature and one temporal signal. It cannot independently represent multiple hypothetical trajectories for the same property, nor does it organise scenario-specific results and their provenance. It is therefore complementary to, but not a replacement for, the Scenario ADE.

2.2 Other versioning approaches

Outside CityGML, distributed version-control approaches such as GeoGig (Wieland *et al.*, 2017) and Git-based CityJSON workflows treat geospatial datasets as evolving content (Idris *et al.*, 2024; Vitalis *et al.*, 2019). Their non-linear histories can represent branches and deltas effectively. Idris *et al.* state that CityJSON's flatter and more compact structure may simplify version tracking compared with the hierarchical XML structure of CityGML which is more verbose but offers better control over individual semantic surfaces. These works remain largely agnostic to the semantic structure of a scenario. They do not inherently distinguish context from strategy, store a simulation configuration or bind a result to a specific urban feature and model run. Similarly, provenance standards address a related but different problem. ISO 19115 metadata norm lineage records sources, processing steps, software, algorithms and outputs. W3C PROV (Belhajjame *et al.*, 2013) uses entities, activities and agents to describe how information was

[1] https://digitwins4peds.eu

[2] https://github.com/opengeospatial/CityGML-3.0CM/releases

generated. Metadata ADE approaches (e.g. Labetski *et al.*, 2018) have already brought lineage concepts into CityGML. These standards demonstrate that structured provenance in semantic city models is feasible. However, existing implementations mainly describe how input datasets were created. They do not organise multiple hypothetical city states and their internal results as a comparable scenario set. The gap is therefore not a lack of versioning or provenance standards individually, but the absence of a practical binding among urban changes, scenario reasoning, simulation configurations and outputs.

3. Method

3.1 Scenario definition

The process consists of four phases. First, in the context of Urban Building Energy Modelling (UBEM) we derive empirical requirements from urban-energy simulations. SimStadt[3] workflows are examined to identify user-supplied inputs, internal libraries, derived values, output resolutions and run metadata. Baseline and exploratory runs vary selected parameters while holding others constant. The aim is not primarily to optimise the energy model, but to observe which information a scenario-management schema must preserve. In the second phase, we convert scenario theory and empirical findings into a set of design principles. In the third phase, we then evaluate successive Scenario ADE versions against these principles and against four concrete scenarios. Eventually, in the fourth and last step, we translate the UML-based conceptual model into two target encodings: XML (and its XSD file) and SQL, the latter specifically for the CityGML 3D City Database 4.x, after which practical encodings are generated and tested. Exploratory runs establish four schema implications. First, simulation inputs must be stored and linked because a small configuration choice can alter outputs. Second, enrichment data must be linked to the exact features it describes. Third, each result must be traceable to its feature, model version and run. Fourth, results must be stored at their native resolution, whether building, surface or time-series level.

3.2 Scenario pillars

The scenario definition framework translates planning theory into the storable components presented in Figure 1.

- The *Context* (External Context): This pillar strictly defines the future boundary conditions and external environment that are outside of the planner's direct control. This includes climate conditions (e.g. future weather files or temperature projections), socio-economic factors (e.g. occupancy schedules), and macro-scale grid constraints.
- The *Strategy* (Interventions): This pillar represents the deliberate physical or technical responses tested against the external context. Strategies comprise envelope modifications (e.g. altering thermal properties like insulation or window-to-wall ratios), system upgrades (e.g. integrating renewable energy sources like photovoltaic panels), and morphological changes (e.g. such as building demolition or reconstruction).
- The *Configuration* (Technical Settings): This final pillar documents the technical metadata to ensure the scenario's simulation is methodologically reproducible. It includes the simulation engine utilized (e.g. SimStadt), the temporal resolution of the inputs and results, and the algorithmic assumptions or calculation standards employed by the physics engines. Without the configuration, two simulation runs producing different numbers for the same city model cannot be compared meaningfully. It is required to record how the result was produced, supporting reproducibility as well as improving transparency and comparison. This allows differences across scenarios to be attributed to the context/strategy changes rather than hidden in configuration differences.

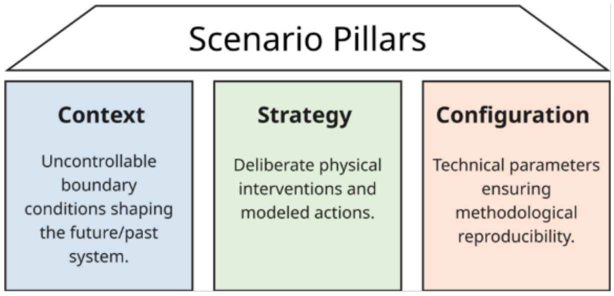

Figure 1. Conceptual pillars of a scenario

3.3 Scenario basic operations

Each scenario can consist of a combination of nine basic operations. These operations cover three levels of changes.
For a top-level feature, the operations are:
1) Insert, 2) delete or 3) replace a top-level feature

For a child feature, the operations are:
4) Insert, 4) delete, 6) replace a child feature;

For any feature attribute, the operations are:
7) Insert, 8) delete or 9) replace an attribute value.

A top-level feature is an independent city-model object such as a building or tree. A child feature is attached to an existing parent, such as a solar panel associated with a roof. Attribute operations represent changes such as introducing a new U-value, removing an obsolete value or replacing the window-to-wall ratio of a wall.

3.4 Design principles

Three main design principles are considered when designing/improving the Scenario ADE. *Domain agnosticism* requires that the Scenario ADE operate above any CityGML module or future domain ADE. *Scenario-set coherence* requires support for related alternatives that share a baseline, because a scenario gains value through comparison. *Forward compatibility* requires that CityGML 2.0 implementation choices do not prevent later migration to CityGML 3.0.

The *Context* pillar requires that boundary conditions and interventions remain independently retrievable; that narratives and temporal horizons travel with their parameters; and that parameters indicate whether they are inputs, outputs or other values. The *Strategy* pillar require that physical interventions be represented without duplicating baseline geometry and that attribute-level changes point unambiguously to their targets. The *Configuration* pillar requires reusable simulation settings, results that are traceable to both their feature and run, and structural distinction among multiple result types belonging to the same object.

4. Scenario ADE data model

The current version of the Scenario ADE 1.0 (beta 6) data model is organised into four separated but interconnected components:

- The *Core module* supplies common identity, lifecycle, and library infrastructure.

[3] https://simstadt.hft-stuttgart.de

https://doi.org/10.5194/isprs-archives-L-4-W3-2026-17-2026

- The *Versioning module* defines city-model states and the feature- or attribute-level deltas between them.
- The *Simulation module* defines reusable engines, settings, and parameters referenced by scenario results.
- The *Scenario module* gives a state and transition scenario-level meaning and stores results within the appropriate alternative.

4.1 Core module

The *Core module* (Figure 2) defines the common abstract classes and shared infrastructure used throughout the ADE. It provides lifecycle information, identifiers, and library containers without introducing concepts tied to a particular application domain. *AbstractADEFeature* is the common abstract parent for Scenario ADE feature types that require basic lifecycle metadata. The *creationDate* and *terminationDate* attributes describe the lifecycle of the encoded representation rather than necessarily the period during which the represented object is valid in the real world. *AbstractFeatureWithLifeSpan* extends *AbstractADEFeature* with identity and validity information. As in CityGML 3.0, the separation between creation or termination dates and validity (*validFrom*, *validTo*) dates supports bitemporal reasoning: an implementation can distinguish when information was stored from when the represented state is intended to apply. The *Core module* adds lifecycle properties to CityGML city objects through the ADE hook. Existing buildings, vegetation objects, etc. can therefore participate in scenario versions. *AbstractLibrary* is the abstract parent for reusable collections of scenario-related resources.

4.2 Versioning module

The *Versioning module* (Figure 3) is for the largest part a backport of it homologous for CityGML 3.0, despite some minor changes and additions. It represents city-model states and the changes between them. It separates the definition of a state from the operations that transform one state into another. This supports linear histories as well as branches of alternative future states. *Version* represents a coherent structural state of the city model. It contains references to the features that belong to that state and it defines what exists in a state, but it does not describe the operations that produced that state. Features, referenced through XLinks, can participate in multiple versions without being duplicated. *VersionTransition* represents the directed relationship between 2 Versions: a predecessor and a successor. When its attribute *ClonePredecessor* is set as True, the target state inherits unchanged source members, and the transition only needs to encode the delta. Several transitions may originate from the same version, thereby forming parallel scenario branches. *FeatureTransaction* represents an insertion, deletion, or replacement involving a whole feature. It is applicable to top-level features, such as buildings and trees, and to child features, such as surfaces or photovoltaic objects. For insertion, property *newFeature* identifies the introduced feature. For deletion, property *oldFeature* identifies the removed feature. Replacement uses both elements. Property *parentFeature* preserves hierarchical placement: for example, it can identify the roof or building to which a photovoltaic feature must be attached.

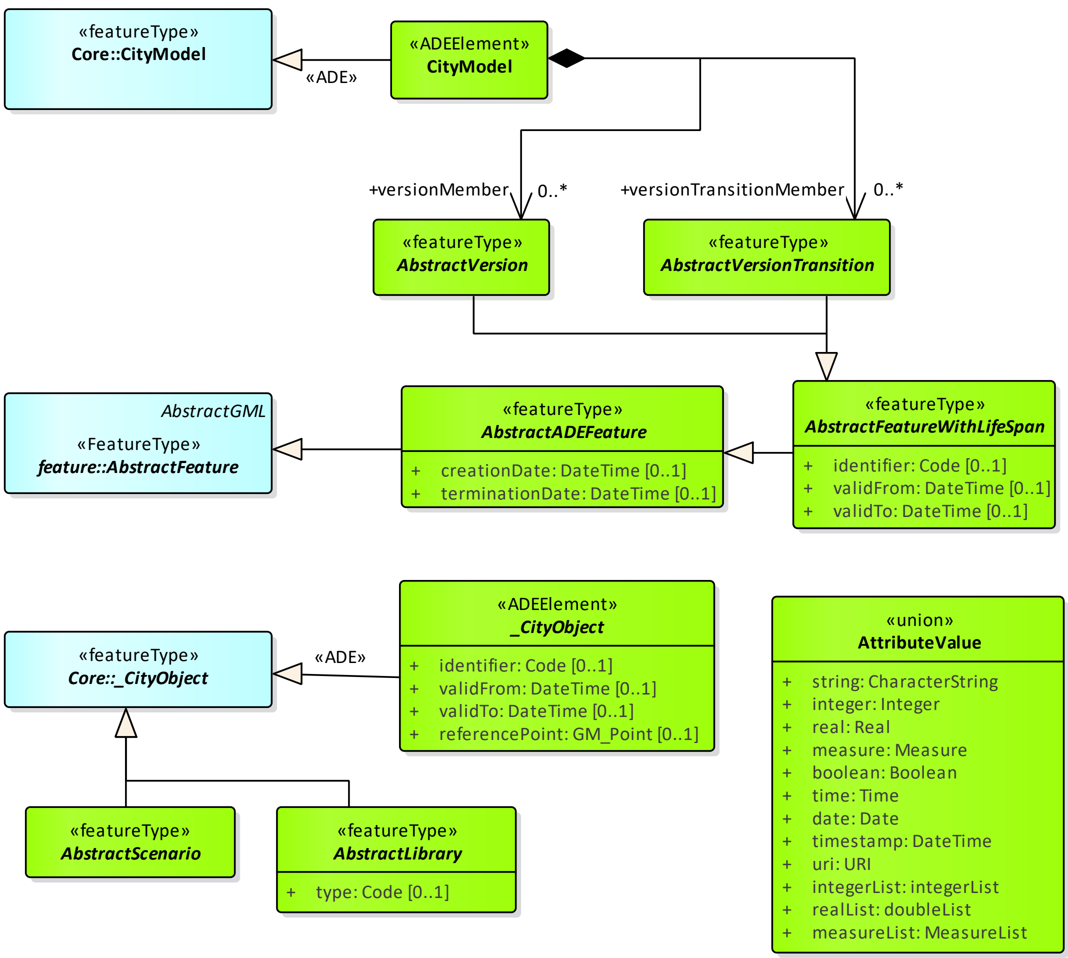

Figure 2. UML diagram of the *Core module* of the Scenario ADE 1.0 (beta 6).

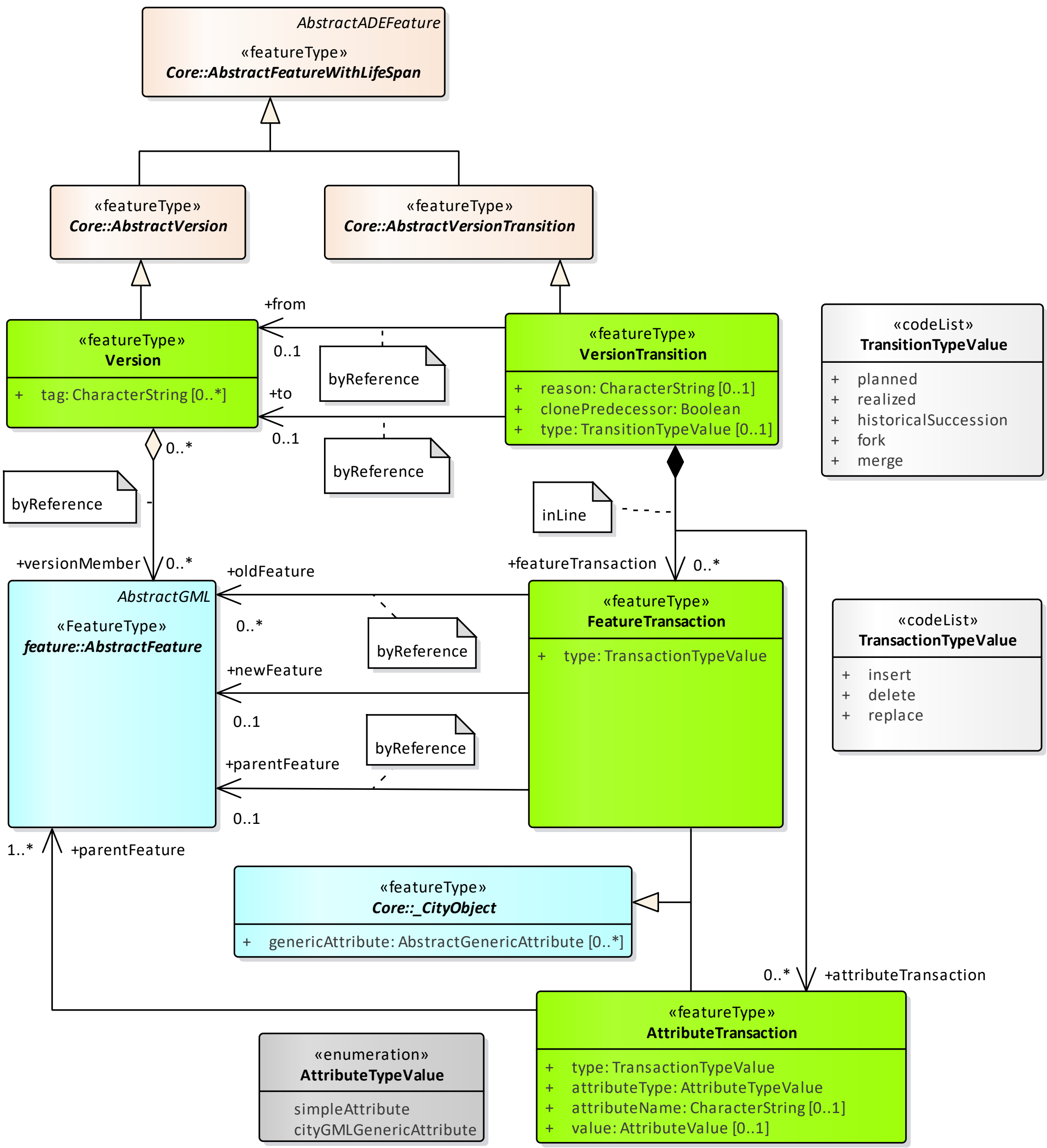

Figure 3. UML diagram of the *Versioning module* of the Scenario ADE 1.0 (beta 6).

AttributeTransaction represents an insertion, deletion, or replacement of an attribute within feature. Multiple *parentFeature* references allow one uniform change to be applied to several objects through one transaction. The value is encoded using the shared *AttributeValue* type. The attribute can be any attribute defined by CityGML (or an ADE), or a *generic attribute*. The only condition is that the named attribute exist and be valid for the referenced feature.

4.3 Simulation module

The *Simulation module* (Figure 4) stores computational resources independently from individual scenarios. Its classes can be referenced repeatedly, enabling configurations and engine descriptions to be reused across scenario alternatives. *SimulationLibrary* acts as the top-level container. It contains reusable instances of simulation engines and settings. The library separates the definition of computational resources from the scenario results that use them.

SimulationEngine identifies the software responsible for a simulation or analytical operation. A single engine instance can be referenced by several *ScenarioResults* objects. This avoids repeating identical engine metadata and enables queries across all scenarios generated with a particular tool or version.
Settings represents a reusable configuration for a simulation or processing workflow. It contains one or more *Parameter* elements and may be referenced by multiple *Scenario* runs. A *Settings* object can describe, for example, the calculation standard, temporal resolution, radiation model, weather-file reference, refurbishment criterion, or photovoltaic-system assumptions.

4.4 Scenario Module

The *Scenario module* (Figure 5) provides the semantic container that connects a scenario narrative and temporal horizon to a city-model version, the transition that defines its strategy, the simulation settings, and the results generated for that alternative.

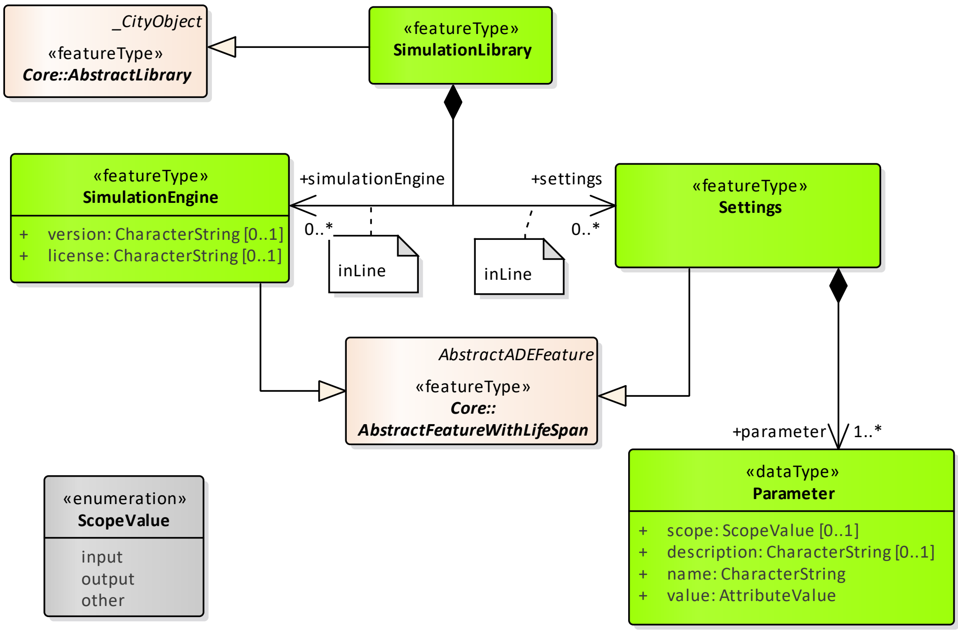

Figure 4. UML diagram of the *Simulation module* of the Scenario ADE 1.0 (beta 6).

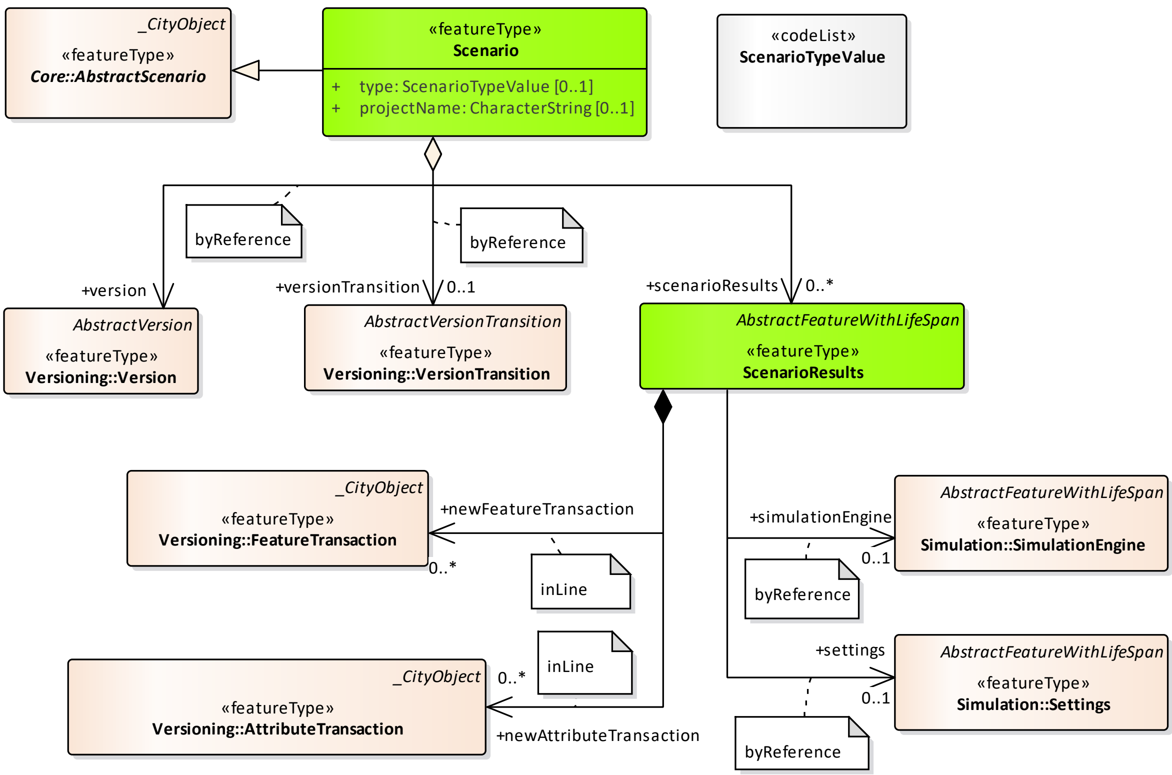

Figure 5. UML diagram of the *Scenario module* of the Scenario ADE 1.0 (beta 6).

Class *Scenario* represents one interpretable alternative within a related scenario set. As mentioned before, the *Version* identifies what the city model contains under the scenario. The *VersionTransition* identifies the strategy or sequence of changes that leads to it.

Class *ScenarioResults* is the scenario-specific container for computational outputs. It keeps results inside the scenario boundary while linking each result back to its target feature and generating configuration. For example, an annual heating-demand value can be represented by an *AttributeTransaction* whose *parentFeature* points to a building. A roof-level photovoltaic result can point to the corresponding roof surface. A richer time-series object may instead be carried through a *FeatureTransaction*. Because each *ScenarioResults* instance belongs to a scenario, outputs from several alternatives remain distinguishable even when they refer to the same building or surface. A related set of scenarios is formed through shared and connected *Version* and *VersionTransition* instances. A baseline version can lead to a future-climate version, from which, for example, refurbishment, photovoltaic, and greening alternatives branch. The resulting directed structure constitutes a so-called "scenario multiverse": a set of comparable alternatives with explicit lineage and a common origin. The structure of the Scenario ADE modules connects scenario meaning, model state, intervention, computational configuration, and output provenance without requiring duplication of the baseline city model.

4.5 Implementation

The Scenario ADE 2.0 has been developed following a well-known and established approach, which has already been described and published by van den Brink *et al*. (2013) and already used in other ADEs For this reason, it will be only briefly summarised here. The conceptual model has been designed in UML using Enterprise Architect. The resulting UML diagrams were exported using ShapeChange[4] to a XSD file, which has been used also for the creation of the DDL scripts for the 3DCityDB schema using the 3DCityDB ADE Manager (Yao *et al.*, 2018).

5. Tests with real-world data

Tests with real-world datasets have been conducted in the districts Feyenoord and Prinsenland in Rotterdam within the DigiTwins4PEDs project (Gao *et al.*, 2025). The CityGML LoD2 models contain the buildings of the study area which were enriched with Dutch open data like the energy performance certificate, the basic registry of buildings and addresses and the measured window-to-wall ratios for building and thematic surfaces supplied by the Municipality of Rotterdam. The baseline dataset is based on the collected data and the building archetypes defined by the TABULA project[5] for the Netherlands.

The first scenario corresponds to the variation of the solar irradiance algorithm used for the energy simulations in SimStadt. The second scenario is based on the detailed window-to-wall ratio of the thematic surfaces. The third scenario corresponds to the forecasting weather from the Royal Meteorological institute of the Netherlands for 2050. The final scenario consists in tree planting as a natural-based solution to reduce the energy demand of buildings for cooling (Rahmawati, 2025). Due to space limits, these scenarios are not presented and discussed in detail here, however more information and many concrete examples are provided in (Auer, 2026). The XML implementation demonstrates that the conceptual model is serializable and schema valid. It also reveals tooling constraints. The successive, successful import of these datasets into the 3DCityDB confirms that the Scenario ADE is not limited to small XML examples. It can operate as a scenario-management layer in a persistent city-model database and can support lineage reconstruction, provenance queries and comparison also at neighbourhood scale.

6. Conclusions

This paper has provided an overview of the current version of the Scenario ADE (beta 6) for CityGML 2.0 and the rationale behind its development. The Scenario ADE stems from the impossibility of CityGML 2.0 to support concurrent hypothetical alternatives, in that simulation results are not connected to precise city-model state, engine, settings and boundary conditions that produced them. In this work, scenario information is decomposed into three main conceptual pillars: *Context*, *Strategy* and *Configuration*. The Context pillar contains external assumptions, the Strategy pillar contains deliberate changes, and the Configuration pillar contains the simulation apparatus and outputs. These pillars are translated into design principles for the proposed data model of the Scenario ADE as they allow to convert scenario theory into entities and relations that can be evaluated in a data model. After reviewing existing literature to reach a clear definition of what a scenario is, and the analysis of alternative modelling approaches – including the new conceptual modules added by CityGML 3.0 to deal with "slow" and "quick" changes within a city (i.e. the *Versioning* and *Dynamizer* modules) – the current version backports CityGML 3.0's *Versioning* module, eventually building upon it and extending it.

The current version of the Scenario ADE can represent object lifecycles and alternative branches, store only the delta from a shared predecessor, avoid duplication for attribute-level changes, identify the parent of inserted child features, reuse simulation configurations and connect each output to its feature and run. In other words, it can track states, time and provenance into one layer that links what changed, why it changed and which assumptions or processes produced the resulting values.
Nine basic operations are identified and supported: insertion/deletion/replacement of a whole top-level feature, insertion/deletion/replacement of a whole child feature, insertion/deletion/update of any attribute (including CityGML *generic attributes*) of any feature.

Although the Scenario ADE is developed in an energy context, it deliberately avoids dependency on any ADE, e.g. the Energy ADE. This decision expands its potential use to other urban domains. The Scenario ADE is available in two encodings: XML and SQL, the latter for the 3DCityDB relational database. In particular, the database implementation shows that scenarios, versions, transitions, settings and results can be stored persistently and queried together.

Tests and results based on real datasets from case studies in Rotterdam (The Netherlands) show that the current version can represent parallel scenario branches in a unified approach, avoid unnecessary duplication of data, preserve object and attribute lineage, and bind simulation outputs to their originating features and configurations. The Scenario ADE can act as a domain-agnostic scenario-management and lightweight provenance layer for CityGML 2.0, although its treatment of context, attribute semantics and non-simulation provenance requires further refinement.

6.1 Limitations and future work

Although the final model is intentionally agnostic, tests have been conducted only within the urban energy domain. Therefore, the schema must be evaluated in other domains to determine whether the three pillars and transaction model hold.
A dedicated *Context* scope for parameters would improve the model. The current parameter scopes distinguish input, output and other values. Extending the scope vocabulary would allow downstream tools to select boundary conditions independently from ordinary calculation settings. A dedicated narrative element on *Scenario* would similarly separate the storyline from generic descriptive text. These changes are comparatively small but would bring the physical schema closer to the theoretical decomposition. Another limitation concerns provenance. The simulation library provides a lightweight custom provenance model for computed results, but deliberate planning actions may

[4] https://github.com/ShapeChange
[5] https://episcope.eu

https://doi.org/10.5194/isprs-archives-L-4-W3-2026-17-2026 | © Author(s) 2026. CC BY 4.0 License.

not originate from a software engine. A municipality, planner or policy document cannot be represented with the same clarity as a *SimulationEngine* and *Settings* combination. The model therefore handles computational provenance more effectively than human or institutional decision provenance.

When it comes to future work and planned improvements, besides overcoming the afore-mentioned limitations, the first priority is the migration to CityGML 3.0. A native implementation will remove several CityGML 2.0 workarounds, align directly with the standard versioning module and provide a cleaner path towards other encodings. The conceptual distinction between the Scenario ADE and the underlying versioning mechanism should however remain: CityGML versioning supplies states and transitions, while the Scenario ADE supplies scenario meaning, configuration and provenance. Additionally, stronger application profiles and validation should be defined, e.g. via Codelists, Schematron rules or domain profiles that may constrain transaction types, permissible target attributes and required parameter metadata without removing the domain-agnostic core.

Acknowledgements

The current version of the Scenario ADE has been developed mainly within the European project DigiTwins4PEDs (Utilisation of urban digital twins to co-create flexible positive energy systems for districts). The project is funded by the European Commission under the Horizon Europe Partnership scheme (Funding Indicator: 03EN3081A). The DUT Call 2022 also contributes to the Urban Transition Mission of Mission Innovation as part of the MICall 2022 initiative. Finally, the authors would like to thank Thomas H. Kolbe (TU München, Germany) for the constructive discussions and suggestions.

References

Agugiaro, G., 2018. 3D City Database extension for the CityGML Scenario ADE 0.2 PostgreSQL Version—Documentation. https://github.com/gioagu/3dcitydb_scenario_ade/blob/954cf0b1204af43124538974edc0e97eb96a8d12/documentation/3DCityDB_Scenario_ADE_0.2_Documentation.pdf

Agugiaro, G., Padsala, R., 2025. A proposal to update and enhance the CityGML Energy Application Domain Extension. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci*. X-4/W6-2025, pp. 1-8. https://doi.org/10.5194/isprs-annals-X-4-W6-2025-1-2025

Amer, M., Tugrul, U.D., Jetter, A., 2018. Chapter 6: A Review of Scenario Planning. *Technology Roadmapping*. World Scientific Publishing Co. Pte. Ltd., pp. 177–232. https://doi.org/10.1142/9789813235342

Auer, F.J., 2026. Testing and Enhancing the Scenario ADE in the context of building energy performance simulations in Rotterdam. MSc thesis, Delft University of Technology, The Netherlands. https://repository.tudelft.nl/record/uuid:42e6250f-7dfb-49e2-bfc1-3277b915bdcb

Belhajjame, K., B'Far, R., Cheney, J., Coppens, S., Cresswell, S., Gil, Y., Groth, P., Klyne, G., Lebo, T., McCusker, J., Miles, S., Myers, J., Sahoo, S., Tilmes, C., 2013. PROV-DM: The PROV Data Model. https://www.w3.org/TR/prov-dm/ (accessed 17 July2026).

Börjeson, L., Höjer, M., Dreborg, K.-H., Ekvall, T., Finnveden, G., 2006. Scenario types and techniques: Towards a user's guide. Futures 38, 723–739. https://doi.org/10.1016/j.futures.2005.12.002

Bradfield, R., Wright, G., Burt, G., Cairns, G., Van Der Heijden, K., 2005. The Origins and Evolution of Scenario Techniques in Long Range Business Planning. *Futures*, 37, 795–812. https://doi.org/10.1016/j.futures.2005.01.003

Gao, W., Agugiaro, G., León-Sánchez, C., 2025. Data-driven energy simulations to evaluate positive energy district potential in Rotterdam. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci*. X-4/W7-2025, pp. 25-32. https://doi.org/10.5194/isprs-annals-X-4-W7-2025-25-2025

Godet, M., 1986. Introduction to la prospective: Seven key ideas and one scenario method. *Futures,* 18, 134–157. https://doi.org/10.1016/0016-3287(86)90094-7

Idris, N.H., Azri, S., Ujang, U., 2024. Reviewing the Versioning in 3D City Model to Track Building History and Progress. IOP Conf. Ser.: Earth Environ. Sci. 1412, 012035. https://doi.org/10.1088/1755-1315/1412/1/012035

Labetski, A., Kumar, K., Ledoux, H., Stoter, J., 2018. A Metadata ADE for CityGML. *Open Geospatial Data, Software and Standards*, 3(16). https://doi.org/10.1186/s40965-018-0057-4

Rahmawati, A., 2025. Scenario-based energy simulation: Modelling tree planting strategy to reduce heating and cooling demand under 2050 climate conditions. MSc Thesis. Delft University of Technology, The Netherlands. https://repository.tudelft.nl/record/uuid:bf97005e-bd6c-4d76-b5e8-91a94c40d99f

Spaniol, M.J., Rowland, N.J., 2019. Defining scenario. *Futures & Foresight Science*, 1, e3. https://doi.org/10.1002/ffo2.3

Van den Brink, L., Stoter, J., Zlatanova, S., 2013. UML-Based Approach to Developing a CityGML Application Domain Extension. *Trans. GIS*, 2013, 17, pp. 920–942, https://doi.org/10.1111/tgis.12026

Vitalis, S., Labetski, A., Arroyo Ohori, K., Ledoux, H., Stoter, J., 2019. A data structure to incorporate versioning in 3D city models. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-4/W8, 123–130. https://doi.org/10.5194/isprs-annals-IV-4-W8-123-2019

Wieland, M., Pittore, M., 2017. A Spatio-Temporal Building Exposure Database and Information Life-Cycle Management Solution. *ISPRS Int. J. of Geo-Information*, 6. https://doi.org/10.3390/ijgi6040114

Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., Kolbe, T.H., 2018. 3DCityDB – A 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data, Software and Standards*, 3. https://doi.org/10.1186/s40965-018-0046-7