Trends and Reviews of Technological Engineering in Education

Vol. 2, No. 1 (2026) | ISSN 3071-2211 | Open Access

RESEARCH ARTICLE

A Spatial-Computing Authoring System for Reconfigurable Infrastructure-Inspection Learning Scenarios

Chengyang Nie[1], Yexin Tian[2,*]

[1] Computer Science Department, Carnegie Mellon University, Pittsburgh, USA 15213

[2] College of Computing, Georgia Institute of Technology, Atlanta, USA 30332

[*] Correspondence: yexintian@gatech.edu ·

Received 3 November 2025 · Accepted 25 February 2026 · Published 30 March 2026

Abstract

This technical design article specifies a spatial-computing authoring system for creating, validating, and delivering reconfigurable infrastructure-inspection learning scenarios. Civil infrastructure education increasingly asks students to reason across drawings, building information models, geographic context, point clouds, photographs, inspection records, and changing site conditions. These representations are often taught in separate tools, while instructors lack a practical way to author spatially situated tasks that connect model evidence to inspection decisions. The proposed system combines a source-adapter layer for IFC, GIS, point-cloud, and image assets; a provenance-aware scene graph; an instructor authoring workspace; a versioned scenario package; and a device-independent learner runtime. It separates observed evidence from interpretation, marks registration and source uncertainty, and treats a spatial annotation as an instructional prompt rather than a substitute for professional inspection. A worked bridge-inspection scenario demonstrates how an instructor can author an accessible learning task around a simulated girder-web indication, provide progressive cues, and assess observation, localization, evidence quality, and reasoned escalation without claiming to diagnose a real structure. The article defines design requirements, data contracts, validation rules, learning workflows, assessment evidence, accessibility provisions, governance, and a staged implementation plan. The proposed system is a design specification, not an evaluated intervention; no learning gains, inspection accuracy, or field-safety effects are claimed.

Keywords: Spatial Computing; Infrastructure Inspection; Augmented Reality; BIM; Engineering Education; Authoring Systems; Evidence-Based Learning

1. Introduction

1.1 The instructional problem

Infrastructure inspection education requires learners to connect what they see in a physical or digital scene with what they can justifiably record, infer, and communicate. A learner may need to locate a crack-like indication on a girder, identify the relevant member and orientation, distinguish an observed feature from a suspected cause, relate the feature to drawings or a model, and decide whether more evidence or expert review is required. The intellectual work is distributed across space, time, documents, and representations. It is not reducible to naming components or navigating a three-dimensional model. Students must learn a disciplined sequence: orient to the asset, inspect the relevant area, record an observation, judge the limits of the evidence, and communicate a defensible next step.

Augmented and mixed reality can place information in a learner's spatial frame, while spatial computing more broadly includes desktop, tablet, headset, and large-display applications that understand three-dimensional relationships. These technologies create opportunities to connect a component, a view, a note, and an instructional action. They also introduce new risks: an annotation may appear authoritative even when its source is uncertain; a registration offset may place a marker on the wrong member; a model may visually conceal an access constraint; and a headset may exclude a learner or compete with the scene for attention. The educational value rests on the quality of the task and its evidence, not on immersion by itself [1–8].

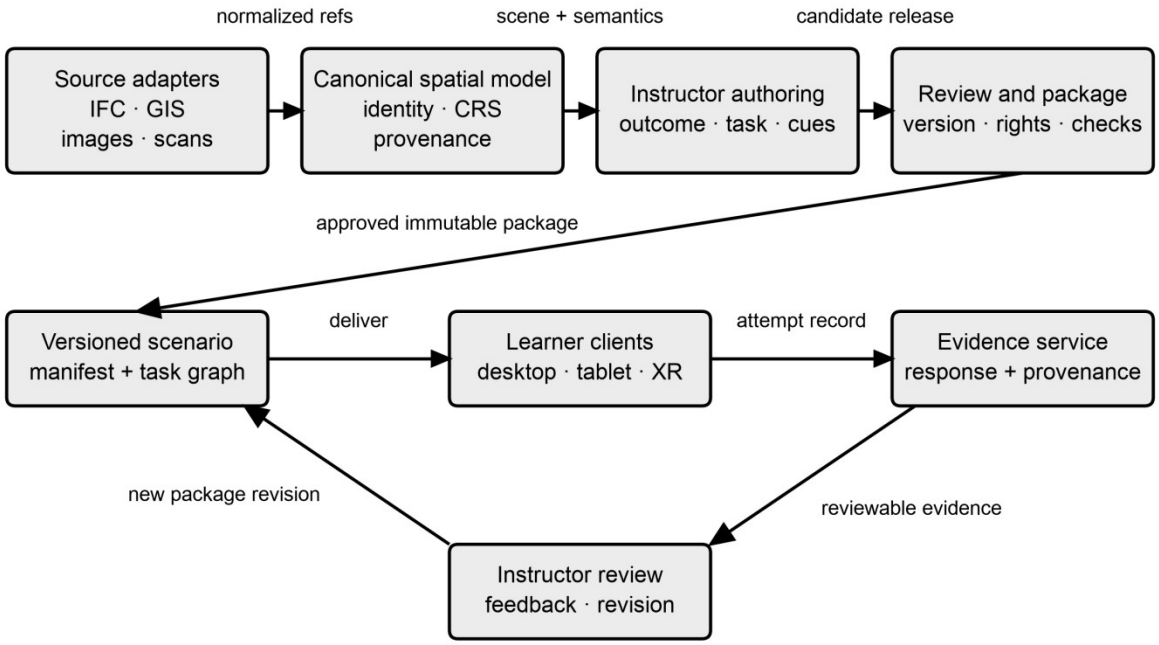
1.2 Aim, questions, and scope

The aim is to specify the architecture and instructional workflow for turning heterogeneous infrastructure information into versioned, reconfigurable inspection-learning scenarios. Four design questions guide the specification: (1) how should source assets with different semantics and coordinate systems be reconciled without erasing their provenance; (2) how should instructors author spatial tasks, evidence requirements, and feedback without writing application code; (3) how can a learner's observations and interpretations be captured in a form that is reviewable and portable; and (4) how should the system communicate uncertainty, accessibility alternatives, and the boundary between simulated practice and real inspection? The response is a layered authoring and delivery system whose package contains both spatial content and an explicit pedagogical model.

© 2026 TRTEE. Open Access under CC BY 4.0

1.3 Contributions and article organization

Figure 1 presents a layered architecture for authoring and delivering spatial-computing learning scenarios. Source adapters ingest IFC, GIS, image, and scan data and normalize them into a canonical spatial model with explicit identity, coordinate-reference, and provenance information. Instructors then associate scenes and their semantics with learning outcomes, tasks, and instructional cues. After review of the candidate release, including its version, rights, and validation checks, the approved package is issued as a versioned scenario manifest and task graph for desktop, tablet, and XR clients. Learner attempts generate reviewable evidence linked to the relevant package and source provenance, which instructors use to provide feedback and prepare a new package revision. Course packages preserve the source systems, while instructional overlays remain separate from source evidence.



Course packages do not write to live asset systems; overlays remain distinct from source evidence.

Figure 1. Layered authoring and delivery architecture with explicit package and evidence boundaries

© 2026 TRTEE. Open Access under CC BY 4.0

2. Educational and technical foundations

2.1 Situated inspection and learning through evidence

Inspection is a form of situated professional reasoning. A visible feature acquires meaning through its location, relation to a component, condition, history, access context, and the limitations of the observer's method. Situated cognition emphasizes that knowledge is used in activity and in relation to the tools and representations through which work is organized [9,10]. For education, this means that an inspection scenario should preserve the relation among a learner's action, the evidence available at that moment, and the decision that follows. A disembodied quiz about defect terminology can assess recall, but it cannot by itself show whether the learner can select a useful view, maintain a spatial reference, or document what the evidence does and does not support.

Cognitive apprenticeship provides a complementary design lens: expert strategies can be made visible through modeling and coaching, followed by scaffolded practice, articulation, reflection, and gradual reduction of support [11]. In the proposed system, an instructor can author a short think-aloud demonstration, a cue that points to a relevant view, and a prompt asking the learner to explain uncertainty. The system does not infer expert judgment from a model score. Instead, it records which cues were available, what evidence the learner selected, and which response was submitted. This makes instructional support part of the record and gives an instructor a basis for distinguishing independent reasoning from successful completion with substantial help.

2.2 Spatial and immersive representations

Augmented reality overlays digital content on a view of the physical world; mixed reality commonly describes a broader interaction in which digital objects are spatially registered and can participate in the user's surrounding context. Spatial-computing instruction need not require a headset: a desktop or tablet can present the same component hierarchy, camera views, and annotation anchors with different interaction mechanics. Early AR frameworks identify the combination of real and virtual content, real-time interaction, and registration in three dimensions as defining characteristics, while later education research documents both instructional opportunities and recurring challenges such as usability, cognitive load, and technical setup [1–4]. Accordingly, this design treats the rendering device as a delivery client and keeps the learning task model independent of any one display.

© 2026 TRTEE. Open Access under CC BY 4.0

2.3 Authoring systems and the instructor's role

An authoring system determines who can create learning experiences, how much programming is required, what forms of interaction can be expressed, and how revisions are reviewed. Research on educational AR authoring tools identifies the need to support educators' workflows and to reduce the distance between pedagogical intent and technical implementation [8,12–15]. Teacher-centered design principles further stress that authoring tools should expose meaningful instructional choices while hiding unnecessary implementation complexity [8]. For infrastructure education, the challenge is to make scene composition accessible without pretending that asset alignment, structural vocabulary, or inspection evidence can be safely automated away (Table 1).

Table 1. Learning principles translated into authoring-system requirements

Learning principle	Authoring consequence	Observable design evidence
Situated reasoning	Link each task to a component, view, evidence source, and decision.	Learner can navigate from an observation to the supporting view and asset identifier.
Scaffolded participation	Let the instructor stage demonstration, prompt, and independent attempt.	Package records support state and instructor-authored cue availability.
Reflection and articulation	Collect a rationale separately from the observation.	Assessment record preserves observation, interpretation, confidence, and next step as separate fields.
Multimedia restraint	Place a cue spatially only when location matters.	Every overlay has an instructional purpose and a non-spatial alternative.
Transfer awareness	Mark simulated success as practice evidence, not qualification.	Learner and instructor views state the limits of the scenario and assessment.

3. Design goals, boundaries, and requirements

3.1 Design goals

The primary goal is authoring efficiency with pedagogical traceability. An instructor should be able to move from a stated learning outcome to a functioning scenario through a sequence of visible choices: select an asset, identify the target component, choose evidence views, define learner actions, configure supports, set a rubric, preview the task, and submit it for review. Efficiency is not measured as minimum clicks alone. It also includes whether another instructor can understand the package, reproduce it, and revise it without breaking object references or confusing a teaching overlay with source evidence. A package that is quick to author but opaque to inspect is not maintainable.

© 2026 TRTEE. Open Access under CC BY 4.0

A second goal is reconfiguration without semantic loss. Instructors should be able to change the asset instance, inspection location, evidence set, or learner sequence while preserving the underlying competency and clearly identifying which parts require renewed review. For example, an instructor might replace one synthetic bridge model with another bridge geometry while retaining a task about recording a spatially anchored observation. Such substitutions can be safe only when stable identifiers, relevant geometry, required evidence, and model limitations are checked. Reconfiguration must therefore trigger dependency checks and review status rather than silently carrying all prior approvals forward.

A third goal is device-independent participation. The same learning objective should be accessible on a headset, tablet, or desktop, though the interaction may differ. A headset can support hands-free spatial orientation; a tablet can provide a familiar camera-based view; a desktop can provide precise orbit, section, and comparison tools. The system should not make a learner’s access to a core assessment depend on comfort with a specific device, fine motor control, color discrimination, stereo vision, or a particular field of view. An alternative interaction is acceptable when it preserves the targeted reasoning and makes any change in the evidence conditions explicit.

3.2 Non-goals and safety boundary

The platform is a teaching authoring and delivery tool. It does not determine the actual condition of an in-service bridge, classify defects for maintenance action, recommend repair, or calculate structural safety. A synthetic indication in a lesson is a teaching stimulus with a known instructional purpose; it is not a validated representation of a real defect. If a course later uses actual inspection imagery, the course owner remains responsible for permissions, anonymization where required, technical context, and review by a qualified domain expert. The application must visibly distinguish simulated or curated material from live field data, and the runtime must not imply that a spatial overlay is an instrument reading.

© 2026 TRTEE. Open Access under CC BY 4.0

3.3 Functional and quality requirements

Quality requirements cover usability, interoperability, performance, reliability, privacy, accessibility, and maintainability. ISO 9241-210 provides a human-centered design process for interactive systems, while ISO/IEC 25010 offers a product-quality model that can help teams organize system-quality requirements [16,17]. These standards do not prescribe a learning design or guarantee suitability; they provide vocabulary for quality planning. The project should set measurable targets for import time, runtime frame stability, startup time, package portability, and data retention only after selecting real target devices and course conditions. The architecture is intentionally modular so that these targets can be tested per client rather than assumed globally (Table 2).

Table 2. Core requirements and acceptance evidence

ID	Requirement	Acceptance evidence
R1	Preserve source identity, version, units, coordinate reference, and licensing metadata.	Imported package report lists each asset, adapter version, source hash, and unresolved metadata.
R2	Keep source geometry, authored overlays, learner annotations, and instructor notes distinct.	Scene inspector and exported evidence report label each object by provenance and role.
R3	Support spatial and non-spatial learner clients for the same task model.	A task package runs on desktop and one spatial client with objective and rubric parity.
R4	Validate component anchors, evidence links, task transitions, and required accessibility alternatives.	Release is blocked on unresolved validation errors; warnings remain visible to reviewers.
R5	Prevent learner-generated content from changing a published course package or external asset.	Permission and integration tests show read-only course assets and isolated attempt data.
R6	Record versioned task, rubric, cue state, evidence selection, and learner response.	Instructor report reconstructs an attempt with the exact package and rubric revision.

© 2026 TRTEE. Open Access under CC BY 4.0

4. System architecture

4.1 Architectural overview

The architecture has six logical layers: source adapters; a canonical spatial and semantic model; an authoring workspace; a scenario package and validation service; device-specific learner clients; and an evidence and instructor-review service. These layers communicate through versioned contracts. The source adapters are responsible for interpreting external formats. The canonical model stores normalized geometry references and metadata without claiming to preserve every source-specific feature. The authoring workspace composes tasks, cues, and evidence. The package service freezes a reviewed scenario version. Clients render that version and submit attempt data. The review service presents evidence to an instructor and preserves the relationship between a learner's action and the content version that produced it.

The design separates the spatial scene from the pedagogical state machine. The scene graph answers what objects and views are available and where they are located. The task model answers what the learner is asked to do, what support is permitted, what evidence is required, and what happens next. This separation allows the same task to run with a headset or desktop view without encoding instructional logic in the rendering client. A course can revise an overlay without altering the rubric, or revise a rubric while preserving the task and asset versions used in completed attempts. Every released change creates a new package revision; completed attempts retain their original package and rubric references.

4.2 Source-adapter and canonical model layer

The source-adapter layer supports Building Information Modeling files such as IFC, geographic information sources such as GIS layers and CityGML, point clouds, photographs, drawings, and tabular inspection records. IFC 4.3.2 extends the schema to infrastructure domains including roads, railways, bridges, waterways, and ports, making it a relevant exchange format for civil infrastructure scenarios [18,19]. CityGML provides an open conceptual model for storing and exchanging semantic 3D city models [20]. These formats are complementary: an IFC model can describe engineered components, while GIS and city models can provide surrounding geographic context. The adapter reports which entities and properties were read, transformed, omitted, or not understood.

© 2026 TRTEE. Open Access under CC BY 4.0

Coordinate management is explicit. Every source declares a CRS or local frame, axis order, units, datum where applicable, and an operation chain to the runtime frame. Record operation identifiers, parameters, software versions, and required grid files; a matrix alone does not represent every map projection or datum transformation [19,20]. The preview checks alignment against known control points or component anchors. Missing units or frame mappings block release until reviewed and resolved. Similar-looking extents do not establish alignment. Registration error is reported as measured or estimated, with its method, units, and applicability to the task. Estimated or unknown error cannot satisfy a metric-accuracy acceptance check without verification (Table 3).

Table 3. Canonical scene-object and evidence metadata

Record group	Required fields	Purpose in the learning package
Identity	Scenario object ID; source object IDs; object type; parent object ID.	Stable linking across views and revisions.
Spatial reference	CRS or local frame; units; axis convention; transform; registration tolerance.	Makes alignment assumptions inspectable and testable.
Provenance	Source URI or filename; revision/hash; adapter version; import date; rights statement.	Supports traceability, licensing review, and reproducibility.
Representation	Source geometry link; display geometry; simplification status; visibility state.	Separates render optimization from technical evidence.
Evidence	Evidence ID; media type; capture context; author; limitations; linked object.	Connects an inspection response to its supporting source.
Instruction	Cue ID; task state; intended objective; availability condition; alternate mode.	Makes overlays and prompts pedagogically and technically reviewable.

4.3 Authoring, package, and runtime services

The authoring workspace is organized around a project and an outcome. A project contains imported assets and mappings; an outcome defines the intended learner capability; and a scenario combines one or more tasks that elicit evidence for that outcome. The workspace provides a scene view, a component browser, a task editor, an evidence panel, a cue editor, an accessibility checklist, and a validation report. These views share selections: clicking a component in the scene highlights its semantic record and reveals attached sources. The author can move between spatial and tabular representations so that a task does not depend on visual scene manipulation alone.

© 2026 TRTEE. Open Access under CC BY 4.0

The package service creates an immutable release artifact containing the scene manifest, source links or permitted bundled assets, task graph, assessment rubric, cue definitions, localization strings, accessibility alternatives, client compatibility declaration, and license notices. A package has a unique identifier, semantic version, creation timestamp, author and reviewer identities, change log, and content hashes. If source data cannot be distributed, the package records a controlled reference and required access conditions. A client checks package compatibility before launch and displays a clear error if a required asset or feature is unavailable. This prevents partial loading from masquerading as a valid attempt.

The runtime client implements a common interaction contract but adapts display and input to the device. It may render the scene as a desktop viewport, a tablet camera view, or an optical/video-see-through headset experience. All clients expose orientation, object selection, evidence inspection, note entry, and task navigation, though gestures differ. A package declares required capabilities such as spatial anchors, image display, or audio playback. If a client cannot provide a capability, it either offers a reviewed alternative or reports that the package cannot run; it must not silently omit an instructional cue that is needed to interpret a task.

4.4 Data flow, trust boundaries, and failure handling

Failure states are instructional states. If spatial tracking is lost, the runtime pauses spatially anchored interaction and tells the learner why; it does not leave stale annotations floating over a shifted scene. If a source image fails to load, the system marks that evidence is unavailable and blocks the affected response or activates an instructor-approved alternative. If the imported model contains unresolved component identity, the authoring system can permit exploration but not publish tasks that depend on that identity. If evidence persistence fails, the learner sees the unsaved status and can export or retry according to course policy. Explicit failure handling protects the integrity of both learning activity and assessment record.

© 2026 TRTEE. Open Access under CC BY 4.0

5. Authoring workflow and scenario package

5.1 From outcome to a reviewed scenario

The workflow (Figure 2) begins with a learning outcome expressed as an observable capability, not as a desired technology feature. For example: ‘Given a simulated inspection location and a set of approved views, the learner records a spatially specific observation, identifies which evidence supports it, and states a proportionate next step.’ The author then defines the performance context, what resources are available, what counts as evidence, and which decisions are intentionally outside the learner’s authority. This phrasing clarifies that the task assesses evidence handling and communication; it does not ask the learner to determine the structural safety of a bridge.

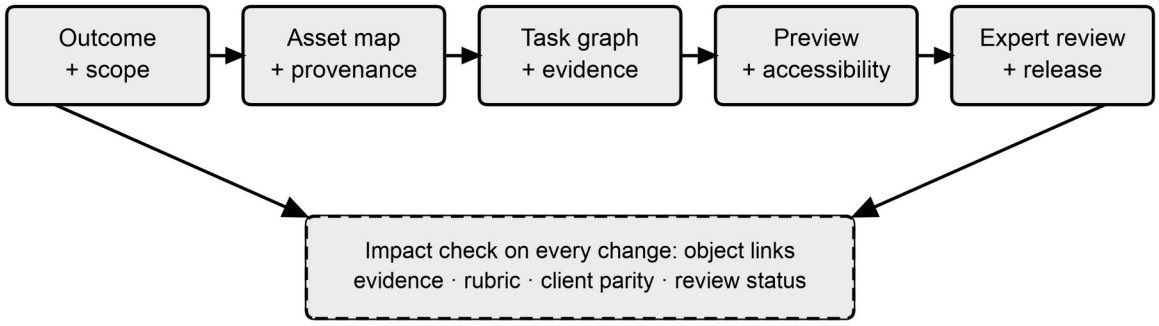


Figure 2. Instructor-led authoring, review, release, and impact-check workflow

Next, the author selects an asset and establishes its instructional scope. The asset may be a sanitized model, a synthetic model, a publicly licensed dataset, or a model created for the course. The author maps relevant objects, confirms coordinate units and transformations, chooses views, and writes a limitations note. The author selects a task template, such as orient-and-locate, compare-evidence, record-observation, or communicate-escalation. Templates are starting structures, not fixed lesson scripts. Authors can compose and rearrange task states but must satisfy a minimal set of requirements, including entry instruction, a learner action, a response record, and a completion or stop condition.

© 2026 TRTEE. Open Access under CC BY 4.0

Finally, the author previews the scenario on each target client, inspects the validation report, and requests review. Reviewers verify the technical representation, instructional alignment, accessibility alternatives, privacy and rights, and the wording of learner-facing limits. A release requires human approval; automated checks can block known defects but cannot approve technical correctness. Any revision to geometry, evidence, rubric, or task logic is logged and triggers an impact check and renewed approval for affected review areas before release. This version discipline supports reuse across course sections while preserving an inspectable record of what learners experienced (Table 4).

Table 4. Authoring stages, controls, and release outputs

Stage	Author action	System output or control
1. Specify	Write outcome, context, learner role, and scope boundary.	Outcome record and explicit excluded decisions.
2. Assemble	Import assets, map objects, confirm coordinate metadata and rights.	Provenance report; unresolved mappings block publication.
3. Compose	Build tasks, cues, feedback, evidence fields, and transitions.	Inspectable task graph and support-state definitions.
4. Preview	Run package in target clients and test alternate modes.	Client-specific preview log and accessibility checklist.
5. Review	Request technical, pedagogical, privacy, and accessibility review.	Reviewer decisions with comments and package revision.
6. Release	Approve version, distribution, and retention settings.	Signed immutable package and learner-facing scope notice.

5.2 Reconfiguration model

Reconfiguration is managed through parameterized scenario slots. An instructor can define a task using a target object role such as ‘primary girder’ rather than hard coding a single source file identifier. When a new asset is substituted, the author maps the role to a candidate object and reviews geometry, semantic class, evidence availability, visibility, and task constraints. The system identifies dependent cues and task references and presents a change impact view. If a replacement changes an instructional assumption, for example, the required view is occluded, or the component hierarchy differs, the author must revise the task or reject the substitution.

© 2026 TRTEE. Open Access under CC BY 4.0

A scenario can also vary task conditions while holding the learning objective stable. The author may create alternative inspection locations, different evidence orderings, or a deliberate missing-evidence case. These variants are authored and reviewed rather than randomly generated without constraint. The package can select a variant based on course assignment or instructor choice, but the variant identifier is included in the attempt record. When the course compares learner attempts, instructors can see whether different variants were used and whether the evidence conditions were equivalent. Reconfiguration thereby supports practice diversity without confusing variation with comparable assessment.

6. Spatial data model and interoperability

6.1 Identity, semantics, and coordinate reference

Interoperability requires more than a file import button. An IFC object identifier, a GIS feature identifier, a point-cloud region, a photograph, and a course activity may refer to related but non-identical entities. The canonical model uses a scenario object ID as a stable reference within a package and maintains explicit links to all external IDs. Those links are typed: ‘represents’, ‘located on’, ‘photographs’, ‘derived from’, or ‘instructionally associated with’. Typed relations help the author and reviewer understand whether two objects are semantically equivalent, geometrically related, or merely included together in a lesson. Ambiguous matches remain unresolved until a human selects or rejects a mapping.

Coordinate reference metadata is stored at the asset and object levels. A source can define a geodetic coordinate reference, a local engineering frame, or both; the scene manifest records how each is transformed into the runtime frame. The authoring preview shows units, orientation axes, bounding extents, and selected known points. If two sources use distinct datums or local origins, the transformation chain is documented rather than collapsed into an unexplained offset. Standards such as ISO 19650 provide an information-management framework for built-asset information, while open exchange formats such as IFC and CityGML provide relevant model structures [18–21]. The proposed system uses these sources as inputs, not as proof that any imported model is complete or fit for a specific teaching claim.

© 2026 TRTEE. Open Access under CC BY 4.0

For local scenario use, geometry can be mapped into an efficient display frame while retaining the source frame and transformation chain. An inverse is retained only where it is defined and valid for the task. Simplification, cropping, and other lossy processing are recorded separately; an inverse coordinate map cannot recover discarded source detail. Each anchor records whether it was checked against a known reference, reviewed by an instructor, estimated, or left unverified. These labels do not replace measured error. Tasks that rely on metric distance or exact location require a justified tolerance and verification under the intended conditions. Unverified precision-dependent anchors block release; broad-orientation tasks may use explicitly reviewed, less precise mappings.

6.2 Evidence lineage and asset transformations

The scenario package keeps three evidence classes distinct: source evidence supplied to the learner; learner-created evidence such as selected view, annotation, note, or response; and instructional material such as a model answer or feedback cue. An assessment report can show all three classes in chronological order. The learner-facing interface can make the classes visually distinct using labels and patterns as well as color, and the exported report uses text labels so that interpretation does not depend on color perception. If a source is simulated, synthetic, or pedagogically altered, the classification appears in the learner's context and in the exported evidence summary.

6.3 Interoperability boundaries

Interoperability should be defined as a set of testable exchanges, not a promise that every asset will work everywhere (Table 5). The first supported exchange set can be deliberately small: IFC 4.3 for selected component geometry and properties; GeoPackage or CityGML for geographic context; common image formats for photographs; and a documented point-cloud subset for scan-derived material. The adapter reports unsupported entities and preserves a source manifest. Later formats can be added behind the same canonical contracts. Standards evolve, and exchange schemas differ in their treatment of geometry, semantics, and references; therefore, each adapter version must be tested with a conformance corpus and a documented support matrix [18–20].

Table 5. Interoperability contract by source class

Source class	Minimum metadata required	Import-time checks
IFC / BIM	Schema version; file hash; units; project and object IDs; coordinate context.	Entity support, geometry validity, duplicate IDs, units and transform review.
GIS / city model	CRS; layer or model version; feature IDs; units and extent.	CRS availability, extent plausibility, geometry validity, layer provenance.
Point cloud	Source identifier; coordinate frame; units;	Frame mapping, density summary,

© 2026 TRTEE. Open Access under CC BY 4.0

Source class	Minimum metadata required	Import-time checks
	registration status; processing notes.	registration note, display budget.
Photograph / video frame	Source rights; capture context when known; timestamp; crop or transform history.	Resolution, orientation, privacy review, provenance and learner-visible limitations.
Inspection record	Schema or column map; date/context; asset and component reference; authoring rights.	Field mapping, missing values, sensitive information, and source-to-object links.

7. Instructional interaction and scaffold policy

7.1 Task graph and learner actions

The scenario is represented as a directed task graph. Each state includes learner-facing instruction, available scene objects and evidence sources, allowed actions, response fields, support configuration, and transition condition. Actions may include orienting the scene, selecting a component, opening a photograph, comparing two views, placing an observation marker, writing a note, rating evidence adequacy, or requesting instructor help. The task graph does not infer a structural diagnosis from these actions. It controls the instructional sequence and records the learner's choices so that an instructor can examine the reasoning path.

The graph supports branches that reflect evidence quality rather than a hidden score threshold. A missing component reference can trigger clarification; a photograph without an observation link can trigger an evidence prompt. Missing evidence or declared uncertainty can lead to escalation. The author defines each branch and its triggering evidence in instructional language. A test panel lists states and transitions and tests normal, incomplete, unsupported, and stop cases. Cyclic graphs require explicit iteration bounds and reachable stop or escalation routes; bounded path tests must not be described as exhaustive coverage of all possible attempts.

7.2 Cues, feedback, and fading

Cues are authored as supports with a stated instructional purpose. A spatial highlight may direct attention to the correct girder segment during initial familiarization; a text prompt may ask the learner to identify the viewing direction; a source panel may offer an inspection terminology excerpt; and a worked example may model the difference between observation and interpretation. Every cue has an availability condition, a content owner, and a fade rule. Authors can make a cue available during guided practice, require the learner to request it during supported practice, and remove it during an independent attempt. The cue policy is visible to instructors and learners.

© 2026 TRTEE. Open Access under CC BY 4.0

Feedback should address the quality of the learner's process and evidence. Feedback can indicate that the selected view does not support the stated location, that the observation uses an interpretive label where a description was requested, or that the next step exceeds the learner's role. It should not invent a diagnosis or imply certainty from a synthetic feature. An explanatory cue can point back to the relevant evidence and ask the learner to revise. This design follows formative-feedback principles that emphasize information useful for closing a gap between current and intended performance rather than praise or score alone [22,23]. The course team remains responsible for the content and technical review of feedback language.

Fade rules are competency-specific and reversible. A learner who demonstrates an independent evidence-linking step can proceed without that prompt, while a later task with a new asset can restore orientation support. The system does not treat prior success as proof that support is never needed again. Authors can set a rule such as 'after two instructor-reviewed attempts with all required evidence completed, hide the location prompt in the next parallel task'; the rule is illustrative and should be validated locally. An instructor can reinstate a support or hold progression with a recorded reason. Safety and accessibility information are never faded as rewards for performance (Table 6).

Table 6. Example support policy for an inspection-learning scenario

Support level	Learner experience	Evidence captured
Demonstration	Expert models how to orient, identify a member, and distinguish observation from inference.	Which example views and explanation were shown.
Guided practice	Spatial focus cue and structured response fields are available.	Cue exposure, selected object, evidence links, and revised response.
Supported practice	Learner requests help; context panel remains available; spatial highlight is reduced.	Help requests, response revisions, and evidence sufficiency judgment.
Independent attempt	Task prompt remains; direct answer cues are hidden; source evidence remains accessible.	Unassisted task path, evidence record, rationale, and uncertainty statement.
Instructor review	Instructor may reopen a cue, request revision, or conclude that evidence is insufficient.	Reviewer action, comment, package version, and final instructional status.

© 2026 TRTEE. Open Access under CC BY 4.0

7.3 Spatial interaction and non-spatial equivalence

The non-spatial client provides equivalent access to the task model through a component tree, named views, image panels, and a tabular evidence list. A learner can complete the same observation and evidence-linking activity without wearing a headset or using a camera overlay. The course should identify whether assessment criteria depend on spatial orientation itself or on the reasoning that can be expressed through multiple interfaces. If spatial orientation is a stated outcome, the alternative mode may assess a related but not identical capability and should be documented as such. This explicitness is preferable to claiming equivalence where interaction conditions materially differ.

8. Worked design example: simulated bridge inspection

8.1 Scenario brief and preparation

The scenario is a course exercise involving a synthetic bridge model and curated image set. The learning objective is for a student to create a specific, evidence-linked observation and recommend an appropriate next educational or professional step. The exercise is not a structural assessment. The bridge asset contains a deck, two primary girders, cross-frames, bearings, and an inspection access zone. A simulated photograph shows a narrow linear indication on a girder web; the teaching package explicitly labels the image and model as synthetic. The feature is selected to prompt careful description and evidence review, not to represent a known defect type or condition rating.

Before release, the author imports the model, verifies its coordinate frame and units, maps the target girder and nearby cross-frame to scenario IDs, and selects a stable orientation view. The author links the synthetic image to the intended component and records its synthetic origin, crop history, and limitations. The course reviewer confirms that the image does not contain real asset identifiers or imply a live maintenance concern. The author then configures the task to let learners inspect a broad view, select the target component, open the image, place or describe an observation, identify the supporting evidence, and choose the next step from a bounded set.

© 2026 TRTEE. Open Access under CC BY 4.0

8.2 Learner sequence

In the inspection state, the learner chooses a view and selects the target member. The system can show a non-answer cue such as ‘confirm the member and side before recording a location.’ The learner opens the image, enters an observation in plain language, and connects it to the image and selected component. The response form asks for a description, location reference, evidence source, confidence or uncertainty statement, and a proportionate next step. It separates ‘what is visible’ from ‘what it might mean’ so that a learner is not encouraged to present an unsupported explanation as an observation. The system records the response as authored learner evidence, not as an engineering record.

If the learner writes ‘fatigue crack’ without describing visible evidence, the feedback asks the learner to separate the observed line from a causal interpretation and to identify which source supports each statement. If the learner places a marker on the deck while citing the girder image, the task requests a location check. If the learner selects ‘close the bridge immediately,’ a role-boundary prompt asks what information and authority would be required before such an operational action. These are instructional branches written and reviewed by domain experts; they do not attempt to predict actual defect severity. A learner can choose ‘insufficient evidence’ and explain what additional view or qualified review would be appropriate.

The independent attempt uses a parallel target region and hides the direct navigation highlight. It retains access to source images and the component tree because selecting relevant evidence is part of the objective. The learner completes the same structured record, then views a model response that explains the difference between observed feature, location, evidence source, uncertainty, and next step. The instructor reviews the attempt with the cue-state history and rubric; if a learner made a location error, the instructor could assign a targeted navigation practice. The system records a learning status such as ‘ready for further practice’ or ‘instructor review required,’ not a professional certification (Table 7).

Table 7. Bridge scenario task states and evidence

Task state	Learner action	Evidence and instructional response
Orient	Identify deck, girder, cross-frame, and inspection side.	Selected object and view; feedback addresses orientation only.
Inspect	Open a synthetic image and describe a visible feature.	Observation text, image ID, object ID, and uncertainty statement.
Link evidence	Connect the observation to a view and location.	Evidence link and spatial reference; mismatch triggers a check prompt.
Reason next step	Choose a bounded next step and explain why.	Rationale distinguishes further evidence, expert review, and operational authority.

© 2026 TRTEE. Open Access under CC BY 4.0

Task state	Learner action	Evidence and instructional response
Reflect	Compare response with a reviewed example.	Learner identifies one evidence strength and one limitation.

8.3 Why the scenario is reconfigurable

The instructional package can be adapted to a second bridge asset by remapping the role ‘primary girder’ and attaching a new synthetic or authorized image. The authoring system checks whether the new member has an equivalent component identity, whether a relevant view exists, and whether the image can be linked with acceptable confidence. If the asset lacks a corresponding cross-frame or if the image direction is unknown, the author must revise the orientation task and learner instructions. A reconfiguration report lists these differences rather than hiding them behind a visually similar scene. The educational outcome can remain stable only if the task still elicits the same target evidence.

9. Assessment model and evidence record

9.1 Construct definition and rubric

Assessment begins by defining the construct narrowly. In the example, the construct is a learner’s ability to produce a traceable observation and a proportionate next step using a specified set of simulated evidence. It is not structural diagnosis, code compliance, or safety certification. A rubric can separately assess component identification, spatial specificity, descriptive clarity, evidence linkage, separation of observation and inference, uncertainty communication, and role-appropriate escalation. Separate dimensions make the feedback useful and reduce the risk that fluent writing compensates for missing evidence. The course team should review the rubric against the learning outcome and the exact evidence available in each scenario version.

The attempt record includes learner ID or approved pseudonymous identifier, course and session, package and rubric versions, selected client, task variant, cue states, source evidence opened, component selections, responses, revisions, help requests, instructor comments, and timestamps needed to reconstruct sequence. Data minimization applies: the system should capture only what supports instruction and review. If gaze, voice, video, or body motion is not required by the learning objective, it should not be collected by default. The instructor report presents evidence with provenance labels and lets reviewers distinguish learner-entered content from system-generated status information.

© 2026 TRTEE. Open Access under CC BY 4.0

A rubric result is formative and local to the course. If the instructor uses thresholds to decide whether to move to another practice task, the threshold should be explicit, reviewable, and subject to human oversight. A missing response is not necessarily evidence of poor competency; it may indicate a device or accessibility failure. A low confidence statement can be a sign of appropriate caution rather than weak performance. The scoring guidance therefore distinguishes substantive errors from insufficient evidence, client failure, and requested assistance. Any high-stakes use would require a separate validity argument and governance process; this design does not establish one.

9.2 Evidence quality and decision rules

Decision rules should be deterministic where possible and readable by instructors. For example, the scenario can require that the learner select an object, attach at least one source view, and enter a descriptive observation before the task can be submitted. More interpretive criteria, such as whether the next step is proportionate, remain for instructor review. A rule engine can support completeness checks, but it should not infer correctness from word matching alone. If optional text analytics are later introduced, the system must show the text span and rule behind a suggestion, label uncertainty, and permit the instructor to reject the suggestion without altering the original learner response (Table 8).

Table 8. Example formative rubric for the worked scenario

Dimension	Developing evidence	Sufficient evidence for this practice task
Component and location	Names a general area or selects an adjacent component.	Identifies the modeled member and gives a usable side or region reference.
Observation	Uses only a conclusion or vague label.	Describes the visible feature in neutral, observable language.
Evidence link	Cites a source that does not clearly show the described feature.	Links a relevant view and identifies its limits.
Observation vs. inference	Presents possible cause as established fact.	Labels interpretation as tentative and separates it from observation.
Next step	Makes an unsupported operational decision.	Requests appropriate additional evidence or qualified review and explains why.

© 2026 TRTEE. Open Access under CC BY 4.0

9.3 Instructor review and learner reflection

The instructor review panel groups evidence by task state rather than presenting one composite score first. An instructor can open the learner's chosen view, compare it with the response, inspect the cue history, and read the rationale. The panel supports comments tied to a response field or evidence link. It also displays package limitations, source classification, and the task rubric so that reviewers do not assess against information the learner could not see. If the record is incomplete because a source failed to load, the interface labels the attempt as technically compromised and prevents an automatic formative status from being mistaken for a learner performance judgment.

Reflection asks the learner to identify what evidence was most useful, what remained uncertain, and what they would do differently in another scenario. This supports metacognitive awareness without turning self-confidence into a proxy for technical correctness. The instructor can compare a learner's reflection with the actual task path and discuss any mismatch. A learner who notices that the available evidence is insufficient should be able to say so without being penalized for declining to overstate a conclusion. Reflective prompts and revision opportunities are consistent with feedback and self-regulated learning frameworks [22–25], but their effects in this specific system remain to be evaluated.

10. Implementation and verification plan

10.1 Minimum viable system

An initial implementation should constrain the technical surface. It can support one synthetic bridge model, one image set, one desktop client, one tablet or headset client, a small group of task templates, and a single instructor review workflow. This scope is sufficient to test the authoring contract, component mapping, task graph, evidence record, and package export without first solving every infrastructure format or device integration. The synthetic asset avoids dependence on restricted live-asset data and allows controlled validation of labels and evidence. The first release should make its limitations visible and should not be marketed as an inspection or asset-management product.

© 2026 TRTEE. Open Access under CC BY 4.0

10.2 Verification stages

Verification proceeds from structural checks to instructional validation. Unit tests check adapters, coordinate transforms, object linking, task transitions, evidence serialization, and access permissions. Integration tests verify that the same package renders on target clients and that learner events refer to the correct source and task versions. Scenario tests cover normal, incomplete, and failure paths. Human reviews check whether a qualified instructor can author and inspect a package, whether the cues match their stated purpose, and whether the evidence record is understandable. None of these tests alone demonstrates a learning effect. They establish that the system implements the design and can be evaluated responsibly.

Spatial registration is tested with known anchors, varied lighting or display conditions where relevant, and an agreed tolerance for the task. If the scenario asks for broad orientation rather than metric placement, its tolerance can be looser; if a learning claim depends on exact component location, the tolerance must be justified and verified. Client performance tests record frame stability, load time, tracking-loss behavior, and recovery. Accessibility tests cover keyboard and touch paths, screen-reader labeling for relevant controls, contrast, non-color indicators, motion settings, and the availability of non-immersive alternatives. Test outcomes are stored with client, package, and device versions so the course team knows which combinations were verified (Table 9).

Table 9. Staged implementation and verification gates

Stage	Build or review focus	Exit evidence
1. Content prototype	Synthetic asset, one scenario, component map, and paper task flow.	Domain and instructor review of scope, terminology, and evidence.
2. Authoring alpha	Import, scene graph, task editor, cue configuration, package preview.	Author can create and explain a complete scenario without code changes.
3. Runtime beta	Desktop plus one spatial client; evidence capture and instructor review.	Cross-client task parity and reconstructable attempt record.
4. Accessibility and governance	Alternative interactions, privacy, rights, security, and failure handling.	Review findings resolved or documented with an approved mitigation.
5. Educational pilot	Usability and evidence interpretation in a course context.	Protocol, consent where applicable, analysis plan, and limitations report.
6. Broader validation	Retention, transfer, implementation, and subgroup considerations.	Independent evidence supports only the claims tested.

© 2026 TRTEE. Open Access under CC BY 4.0

10.3 Research and evaluation agenda

A first educational evaluation should examine whether instructors can author a valid scenario and whether learners understand the difference between the modeled object, a source image, and an instructional cue. Usability tasks can test navigation, evidence selection, response entry, and recovery from tracking or loading errors. Instructors can review de-identified attempt records and describe the limits of inferences about learners' reasoning. These outcomes focus on feasibility and interpretation rather than whether immersion improves performance. A later study can compare spatial and non-spatial clients while holding task content, cue policy, instructor guidance, and assessment constant as much as practical.

11. Accessibility, privacy, and governance

11.1 Accessible authoring and participation

Accessibility is addressed in both authoring and runtime. Authors need keyboard-operable controls, clear selection state, text labels for scene objects, and the ability to enter or revise task content without manipulating a 3D gizmo. Learners need alternatives to gaze, pinch, controller, or head movement; captions for audio; adjustable text size; readable contrast; and a stable mode for people who experience motion discomfort. WCAG 2.2 provides web-content accessibility guidance, while Universal Design for Learning offers a framework for multiple means of engagement, representation, and action and expression [26,27]. Neither replaces testing with learners who use the system under actual course conditions.

11.2 Privacy, licensing, and data retention

Infrastructure learning assets can reveal sensitive locations, security features, maintenance histories, or personal information. Course owners should select data appropriate for instruction, remove unnecessary identifiers, review distribution rights, and define who can access source assets and learner responses. The package should record rights and licensing metadata, but metadata itself cannot grant reuse permission. The platform should provide role-based access, encryption in transit and at rest, and access audit logs for learner records, and configurable retention. Data collection should be proportional to the learning purpose; no biometric or continuous behavior tracking is required by this design.

© 2026 TRTEE. Open Access under CC BY 4.0

Learners should be told what attempt information is collected, who can see it, how long it is retained, and whether it is used for grading or research. If an institution conducts research using course records, it should follow its applicable ethics and consent processes; this article reports no human-participant study. The system can separate instructional records from research exports and provide de-identification controls, but a local governance review remains necessary. Reuse of attempt data for model training, automated profiling, or external analytics is outside the default design and should require a separate, explicit authorization process.

11.3 Human authority and algorithmic restraint

The system can check completeness, compare identifiers, and guide an author through required review steps. It should not autonomously determine that a bridge is safe, diagnose a defect, rank learners for employment, or issue a professional credential. Any optional automated text suggestion is presented as a draft that cites the learner response or source record it uses; it cannot overwrite the original evidence or make a final assessment decision. Instructors can reject suggestions and must be able to see the underlying evidence. This boundary is especially important because spatial overlays and polished visualizations can create an impression of technical certainty beyond the data.

Governance assigns clear responsibilities. The course owner approves the learning outcome, assessment, asset use, and retention policy. A domain reviewer checks technical terms, component mapping, and the limits of the synthetic or curated content. Accessibility review tests the authored alternatives. A platform administrator manages accounts and release permissions. The instructor decides how to use learner evidence for course feedback. A release record identifies each approval and the version approved. If a defect is discovered in a package, administrators can suspend distribution, notify affected instructors, and preserve prior attempt records for transparent review.

© 2026 TRTEE. Open Access under CC BY 4.0

12. Discussion, limitations, and conclusion

12.1 Design implications

The architecture treats reusability as controlled adaptation. A scenario can move across assets or devices only when its semantics, evidence, and assessment conditions remain sufficiently aligned. This view avoids a common interoperability trap in which a lesson is considered portable because the geometry loads. Educational meaning depends on which object is selected, what source is available, which cue appears, and how the learner response is judged. Versioned packages and change-impact reports make those dependencies inspectable, but only human review can decide whether a changed package still teaches the same capability. Related BIM curricula and AR/MR inspection applications provide technical precedents, without establishing the educational effects of this proposed system [28–36]. BIM/GIS integration research underscores the need to preserve information flow and semantic mappings across model boundaries [37,38]. Spatial-affordance, multimedia-learning, and cognitive-load frameworks remain useful checks against equating richer rendering with learning [39–41].

12.2 Limitations

This article presents an architectural and pedagogical design, not an implemented or evaluated system. The layers, package model, tables, task graph, rubric, and bridge scenario are design artifacts. No prototype performance, learner outcome, instructor workload, accuracy improvement, cost saving, or transfer effects are reported. The suggested metadata and review rules require refinement against real authoring teams, target devices, and source formats. In particular, the canonical spatial model may need changes to support discipline-specific component taxonomies and very large point clouds without compromising traceability.

The worked example uses synthetic material and intentionally avoids real diagnostic claims. It cannot demonstrate how learners respond to actual asset variability, environmental conditions, fatigue, restricted access, team communication, or the uncertainty of field evidence. A static image and modeled bridge do not represent the full professional context. The system also cannot ensure that a course’s rubric is valid or that a technically correct coordinate transformation yields a pedagogically meaningful task. Those questions require domain review and empirical work beyond this specification.

© 2026 TRTEE. Open Access under CC BY 4.0

12.3 Conclusion

This article has specified a spatial-computing authoring system for reconfigurable infrastructure-inspection learning scenarios. The design connects source adapters, a provenance-aware scene model, instructor-authored tasks and cues, immutable scenario packages, device-independent learner clients, and reviewable evidence records. It places learning outcomes and evidence requirements at the center of authoring, makes coordinate assumptions and asset transformations visible, and separates observations from interpretations. A simulated bridge scenario illustrates how students can practice localization, evidence selection, uncertainty communication, and role-appropriate escalation while remaining clearly outside operational inspection and diagnosis.

The system’s value remains a proposition to be tested. A useful next step is a limited prototype with one synthetic asset and two learner clients, followed by expert review and usability testing. Later studies can examine whether instructors can author and maintain scenarios, whether learners understand the evidence relationships, and whether any targeted skill transfers to supervised practice. Pending such studies, the architecture should be understood as a detailed implementation blueprint and a set of testable design commitments, not as evidence that spatial computing improves inspection competence.

© 2026 TRTEE. Open Access under CC BY 4.0

Acknowledgements

N/A.

Author contributions

C.Y.N. is responsible for Data Curation, Preparation; Resources, Writing – Original Draft; Y.X.T. is responsible for Conceptualization, Methodology, Supervision, Validation.

Conflicts of Interest

No claim.

Use of AI-assisted tools

The authors used AI-assisted tools for language drafting, document formatting, and figure preparation, and take full responsibility for the accuracy and integrity of the manuscript.

© 2026 TRTEE. Open Access under CC BY 4.0

References

1. Azuma, R. T. (1997). A survey of augmented reality. Presence: Teleoperators and Virtual Environments, 6(4), 355–385. https://doi.org/10.1162/pres.1997.6.4.355
2. Dede, C. (2009). Immersive interfaces for engagement and learning. Science, 323(5910), 66–69. https://doi.org/10.1126/science.1167311
3. Wu, H.-K., Lee, S. W.-Y., Chang, H.-Y., & Liang, J.-C. (2013). Current status, opportunities and challenges of augmented reality in education. Computers & Education, 62, 41–49. https://doi.org/10.1016/j.compedu.2012.10.024
4. Akçayır, M., & Akçayır, G. (2017). Advantages and challenges associated with augmented reality for education: A systematic review of the literature. Educational Research Review, 20, 1–11. https://doi.org/10.1016/j.edurev.2016.11.002
5. Makransky, G., & Petersen, G. B. (2021). The Cognitive Affective Model of Immersive Learning (CAMIL): A theoretical research-based model of learning in immersive virtual reality. Educational Psychology Review, 33, 937–958. https://doi.org/10.1007/s10648-020-09586-2
6. Radianti, J., Majchrzak, T. A., Fromm, J., & Wohlgenannt, I. (2020). A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. Computers & Education, 147, 103778. https://doi.org/10.1016/j.compedu.2019.103778
7. Dengel, A., Iqbal, M. Z., Grafe, S., & Mangina, E. (2022). A review on augmented reality authoring toolkits for education. Frontiers in Virtual Reality, 3, 798032. https://doi.org/10.3389/frvir.2022.798032
8. Silva, M. M. O., Roberto, R. A., Radu, I., Cavalcante, P. S., Schneider, B., & Teichrieb, V. (2024). Development of design principles for AR authoring tools for education based on teachers' perspectives. IEEE Transactions on Learning Technologies, 17, 677–690. https://doi.org/10.1109/TLT.2023.3327592
9. Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. Educational Researcher, 18(1), 32–42. https://doi.org/10.3102/0013189X018001032
10. Lave, J., & Wenger, E. (1991). Situated learning: Legitimate peripheral participation. Cambridge University Press.
11. Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), Knowing, learning, and instruction: Essays in honor of Robert Glaser (pp. 453–494). Lawrence Erlbaum Associates. https://doi.org/10.4324/9781315044408-14
12. Ashtari, N., Bunt, A., McGrenere, J., Neben, N., & Young, J. E. (2020). Creating augmented and virtual reality applications: Current practices, challenges, and opportunities. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/3313831.3376722
13. Gandy, M., & MacIntyre, B. (2014). Designer's augmented reality toolkit, ten years later: Implications for new media authoring tools. Proceedings of the 27th ACM Symposium on User Interface Software and Technology (pp. 627–636). https://doi.org/10.1145/2642918.2647369
14. Reipschläger, P., & Dachselt, R. (2019). DesignAR: Immersive 3D-modeling combining augmented reality with interactive displays. Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces (pp. 29–41). https://doi.org/10.1145/3343055.3359718
15. Chidambaram, S., Reddy, S. S., Rumple, M. D., Ipsita, A., Villanueva, A. M., Redick, T. S., Stuerzlinger, W., & Ramani, K. (2022). EditAR: A digital twin authoring environment for creation of AR/VR and video instructions from a single demonstration. Proceedings of the 2022 IEEE International Symposium on Mixed and Augmented Reality (pp. 326–335). https://doi.org/10.1109/ISMAR55827.2022.00048
16. ISO. (2019). ISO 9241-210:2019 Ergonomics of human-system interaction—Part 210: Human-centred design for interactive systems. International Organization for Standardization. https://doi.org/10.3403/30388991
17. ISO/IEC. (2023). ISO/IEC 25010:2023 Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Product quality model. International Organization for Standardization. https://doi.org/10.3403/30421227
18. ISO. (2018). ISO 19650-1:2018 Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM)—Information management using building information modelling—Part 1: Concepts and principles. International Organization for Standardization. https://doi.org/10.3403/30333754u
19. ISO. (2024). ISO 16739-1:2024 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries—Part 1: Data schema. International Organization for Standardization. https://doi.org/10.3403/30452651
20. Open Geospatial Consortium. (2021). OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard, Version 3.0 (OGC 20-010). https://doi.org/10.62973/20-010

© 2026 TRTEE. Open Access under CC BY 4.0

21. Zhu, J., & Wu, P. (2022). BIM/GIS data integration from the perspective of information flow. Automation in Construction, 136, 104166. https://doi.org/10.1016/j.autcon.2022.104166

22. Shute, V. J. (2008). Focus on formative feedback. Review of Educational Research, 78(1), 153–189. https://doi.org/10.3102/0034654307313795

23. Hattie, J., & Timperley, H. (2007). The power of feedback. Review of Educational Research, 77(1), 81–112. https://doi.org/10.3102/003465430298487

24. Zimmerman, B. J. (2002). Becoming a self-regulated learner: An overview. Theory Into Practice, 41(2), 64–70. https://doi.org/10.1207/S15430421TIP4102_2

25. Chi, M. T. H., de Leeuw, N., Chiu, M.-H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. Cognitive Science, 18(3), 439–477. https://doi.org/10.1207/s15516709cog1803_3

26. World Wide Web Consortium. (2023). Web Content Accessibility Guidelines (WCAG) 2.2. W3C Recommendation. https://www.w3.org/TR/2023/REC-WCAG22-20231005/

27. CAST. (2024). Universal Design for Learning Guidelines version 3.0. https://udlguidelines.cast.org/

28. Sacks, R., & Barak, R. (2010). Teaching building information modeling as an integral part of freshman year civil engineering education. Journal of Professional Issues in Engineering Education and Practice, 136(1), 30–38. https://doi.org/10.1061/(ASCE)EI.1943-5541.0000003

29. Ahn, Y. H., Cho, C.-S., & Lee, N. (2013). Building information modeling: Systematic course development for undergraduate construction students. Journal of Professional Issues in Engineering Education and Practice, 139(4), 290–300. https://doi.org/10.1061/(ASCE)EI.1943-5541.0000164

30. Shirazi, A., & Behzadan, A. H. (2015). Design and assessment of a mobile augmented reality-based information delivery tool for construction and civil engineering curriculum. Journal of Professional Issues in Engineering Education and Practice, 141(3), 04014012. https://doi.org/10.1061/(ASCE)EI.1943-5541.0000229

31. Sacks, R., & Pikas, E. (2013). Building information modeling education for construction engineering and management. I: Industry requirements, state of the art, and gap analysis. Journal of Construction Engineering and Management, 139(11), 04013016. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000759

32. Pikas, E., Sacks, R., & Hazzan, O. (2013). Building information modeling education for construction engineering and management. II: Procedures and implementation case study. Journal of Construction Engineering and Management, 139(11), 05013002. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000765

33. Martins, A. C. P., Castellano, I. R., César Júnior, K. M. L., de Carvalho, J. M. F., Bellon, F. G., de Oliveira, D. S., & Ribeiro, J. C. L. (2024). BIM-based mixed reality application for bridge inspection. Automation in Construction, 168, 105775. https://doi.org/10.1016/j.autcon.2024.105775

34. Nguyen, D.-C., Nguyen, T.-Q., Jin, R., Jeon, C.-H., & Shim, C.-S. (2022). BIM-based mixed-reality application for bridge inspection and maintenance. Construction Innovation, 22(3), 487–503. https://doi.org/10.1108/CI-04-2021-0069

35. May, K. W., KC, C., Ochoa, J. J., Gu, N., Walsh, J., Smith, R. T., & Thomas, B. H. (2022). The identification, development, and evaluation of BIM-ARDM: A BIM-based AR defect management system for construction inspections. Buildings, 12(2), 140. https://doi.org/10.3390/buildings12020140

36. Tan, Y., Xu, W., Chen, P., & Zhang, S. (2024). Building defect inspection and data management using computer vision, augmented reality, and BIM technology. Automation in Construction, 160, 105318. https://doi.org/10.1016/j.autcon.2024.105318

37. Liu, X., Wang, X., Wright, G., Cheng, J., Li, X., & Liu, R. (2017). A state-of-the-art review on the integration of building information modeling (BIM) and geographic information system (GIS). ISPRS International Journal of Geo-Information, 6(2), 53. https://doi.org/10.3390/ijgi6020053

38. Zhu, J., Wright, G., Wang, J., & Wang, X. (2018). A critical review of the integration of geographic information system and building information modelling at the data level. ISPRS International Journal of Geo-Information, 7(2), 66. https://doi.org/10.3390/ijgi7020066

39. Dalgarno, B., & Lee, M. J. W. (2010). What are the learning affordances of 3-D virtual environments? British Journal of Educational Technology, 41(1), 10–32. https://doi.org/10.1111/j.1467-8535.2009.01038.x

40. Mayer, R. E. (2009). Multimedia learning (2nd ed.). Cambridge University Press.

41. Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. Cognitive Science, 12(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4

© 2026 TRTEE. Open Access under CC BY 4.0