PERSPECTIVE

Turning Construction Data into Understanding

Construction does not have a shortage of data. It has a shortage of data that carries enough meaning for people, software, and AI to reason about the work consistently.

By Mark Klusza · CTO, Inertia Systems, Inc. · September 16, 2026 · 8 min read



Data is not self-explanatory

A modern construction project can contain millions of data points. Models include objects and parameters. Drawings contain lines, symbols, notes, dimensions, and callouts. Project management solutions contain RFIs, submittals, observations, inspections, and correspondence. Schedules

contain activities and dependencies. Cost systems contain codes, commitments, changes, and forecasts. Field tools add photographs, scans, videos, measurements, and daily records.

The common assumption is that once this information is digital, it is ready to be searched, automated, and analyzed. It is not. A value such as '2 HR' could describe a fire rating. 'Level 03' could be a building floor, a model reference, or part of a document name. 'Approved' could mean approved as noted, approved for fabrication, accepted by an inspector, or simply moved to a completed status. Data becomes useful only when its meaning is clear in context.

This is the role of semantic language: to describe not just values, but what those values represent and how they relate to other project concepts. Semantics turns isolated information into statements that systems can interpret.

> *Data becomes useful only when its meaning is clear in context.*

Syntax moves information; semantics carries meaning

Construction technology has become good at exchanging files and records. APIs can move data from one solution to another. Common schemas can define fields, formats, and data types. These are important capabilities, but they do not guarantee shared understanding.

Syntax describes how information is structured. It can specify that a record has an identifier, a name, a date, and a status. Semantics explains what kind of record it is, what the identifier refers to, what event the date represents, and what the status means within a controlled process. Two systems can exchange syntactically valid data while assigning different meanings to the same fields.

For example, transferring a record labeled 'Door 101' does not tell the receiving system whether 101 is a room number, mark, type, tag, schedule row, or asset identifier. Semantic structure can state that the record represents a specific door instance, located in a specific wall, connecting defined spaces, classified by a door type, and referenced by particular documents. The data is no longer merely transferred; it is understood in relation to the project.

Semantic statements create a construction knowledge graph

A practical semantic structure expresses information as connected statements: an object has a type; an object is located in a space; a requirement applies to an object; a document defines a

requirement; an activity installs an object; a company performs an activity; an inspection verifies a requirement; a photograph provides evidence for an inspection.

Each statement is simple. Together they form a graph of the project. This graph can connect information without forcing every source system into a single hierarchy. A door can simultaneously belong to a building location, architectural system, procurement package, schedule activity, inspection plan, and owner asset register. Those relationships are many-to-many because construction itself is many-to-many.

This matters because conventional folder structures and tables tend to flatten those relationships. A document can be placed in one folder even though it affects many rooms and systems. A spreadsheet row can reference one status even though several approvals contribute to readiness. A semantic graph allows the same record to participate in every relevant context while retaining one governed identity.

Understanding the drawing like a superintendent

The importance of semantics becomes especially clear in 2D drawings. To a basic computer vision system, a drawing is an image containing lines, shapes, and text. To an experienced superintendent, it is a representation of construction intent. The superintendent recognizes rooms, assemblies, dimensions, access conditions, sequencing constraints, trade interfaces, detail references, and potential conflicts.

That understanding depends on relationships. A symbol is interpreted differently based on the sheet discipline, nearby labels, legend, scale, view type, location, and linked detail. A note gains meaning from the object or region to which it applies. A detail callout connects one graphical area to another document context. The drawing is not a collection of independent marks; it is a language.

Semantic processing makes those relationships explicit. It can connect a graphical door symbol to a governed door object, its schedule entry, specification section, submittal, RFI history, model representation, inspection requirements, and field evidence. The familiar drawing can then become an interface into the broader project knowledge structure rather than a static endpoint.

Semantic context changes search into reasoning

Traditional search retrieves documents that contain matching words. Semantic search can retrieve information based on concepts and relationships, even when the exact wording differs. More

importantly, a semantic system can reason across connected records to answer operational questions.

Consider the question, 'Can we install the doors on Level 4 next week?' Keyword search might return the door schedule, submittals, RFIs, and schedule activity. A semantically connected system can evaluate which door instances are in the Level 4 scope, whether their types are approved, whether required openings and frames are ready, whether relevant RFIs changed the requirements, whether procurement or delivery constraints remain, whether predecessor work is complete, and which inspections are required.

The answer is not stored in any single document. It is assembled from relationships. Just as importantly, the system can show the records and evidence behind the conclusion. That traceability is essential when decisions affect safety, quality, cost, schedule, or contractual obligations.

Better semantics produce safer automation

Automation without semantic context is brittle. A workflow may route every item with a matching label to the same person, even when the label describes different concepts. An AI model may summarize the wrong revision or associate a photograph with the wrong location. A dashboard may count records consistently while combining statuses that represent different stages of work.

Semantic constraints reduce these risks. They can define that an inspection verifies a requirement, that a superseded drawing cannot be treated as the current governing record, that an installed object must be a specific instance rather than only a type, and that evidence has a source, timestamp, location, and responsible party. These rules give automation boundaries.

They also allow confidence to be decomposed. A system may be highly confident that it classified a door, less confident that it matched the correct instance, and uncertain whether the applicable requirement changed in a recent addendum. Exposing those distinctions supports targeted human review. It is far more useful than presenting one opaque confidence score.

Semantic structure is an organizational discipline

Semantic language is not created by purchasing a graph database or adding AI to a document repository. It requires teams to define identifiers, classifications, relationships, status meanings, source authority, revision rules, and evidence requirements. Technology implements those decisions; it does not make them automatically.

The work should begin with high-value questions and workflows. Identify the information needed to answer each question, the concepts involved, the relationships among them, and the authoritative sources. Map the terms used by different disciplines. Decide where synonyms are acceptable and where a governed distinction is necessary. Establish how changes are versioned and how exceptions are handled.

This foundation should be practical and extensible. Construction language evolves across project types, regions, owners, and trades. A semantic model must accommodate local vocabulary while connecting it to shared concepts. It should make the system easier for users, not require field teams to become data modelers.

The next construction solution is a meaning layer

For years, the industry has focused on digitizing content and moving it into centralized systems. That work was necessary, but storing more data does not solve the problem of understanding it. The next generation of construction technology must operate as a meaning layer across the tools and documents teams already use.

Systems provide repeatable workflows. Ontology defines the shared concepts and relationships. Semantic language expresses project information in a form that people, software, and AI can interpret consistently. Together, these capabilities create the conditions for reliable construction intelligence.

The result is not simply faster search or better dashboards. It is the ability to ask a project question and receive an answer grounded in the current documents, governed objects, responsible parties, workflow status, and supporting evidence. It is the ability to see the ripple effect of a change before it becomes rework. It is the ability to carry construction intent from design through installation and into operations without repeatedly losing the story.

Construction data becomes valuable when the system knows what it means. Semantic language is how that meaning becomes explicit, connected, and usable at scale.

Read online: inertiasystems.com/resources/blog/turning-construction-data-into-understanding

© 2026 Inertia Systems, Inc. All rights reserved.