

Shared Spatial Context

Paper 1 of 4 — engineering in the same world without working in the same model

A civil engineer is working up the paving and grading plans for a site. The job is moving water — slopes, elevations, drainage paths, finished grade.

Then there is a pump.

Someone else placed it. Process cares what it does. Electrical cares about power to it. Instrumentation may care about its controls. A vendor supplied the model. The civil engineer created none of it.

But it is now their problem twice over. It occupies space the water has to move around — and it needs a foundation, which is also their work.

The pump has crossed a boundary — not a file boundary, an engineering one. Something that belonged to another person's work has become part of mine.

That is ordinary engineering coordination. The difficulty is that our digital environments often do not reflect it.

One World, Many Digital Worlds

Engineering applications are specialized for good reasons. A civil engineer needs different tools than a piping designer. An operating facility keeps information that was irrelevant during design. A construction organization arranges the same physical space differently from the design organization. There is no reason to expect everyone to use one application, and forcing it would usually make things worse.

The problem is not specialization. It is when specialization quietly becomes separation.

One team knows exactly where something belongs in its own coordinate system. Another knows exactly where it belongs in theirs. One model is geographically located. Another uses a local project system. A third sits on an origin somebody chose for convenience years ago.

All three can be correct. The difficulty is not deciding which one is right. It is preserving the relationship between them.

The engineering question underneath is always the same: where is this thing relative to the things I am responsible for? That is a spatial question, and answering it should not require anyone to abandon the application that is useful to them.

The Federation Problem

The usual response is to federate. Gather the models, attach them, publish them, combine them in a viewer, coordinate from there.

Federation is useful. It also creates its own questions. Who owns the combined model? Who keeps it current? Which copy is authoritative? What happens when individual disciplines need to keep working independently, or when a company outside the project needs to take part? How are local origins and rotations resolved?

And underneath all of them, one more:

Why should seeing someone else's work require taking control of it?

A useful shared environment should let teams stay independent. The civil engineer keeps working in the civil model. The piping designer keeps working in the piping model. Each remains responsible for their own work — and each can see the work of others where it affects them.

That is a different kind of federation. Not a new master model. A shared spatial agreement.

Agree on Space, Not on Everything Else

A project does not need one model. It needs consistent answers to a few questions. Where is this work? How is this model positioned relative to the others? What is the relationship between my local coordinates and the project?

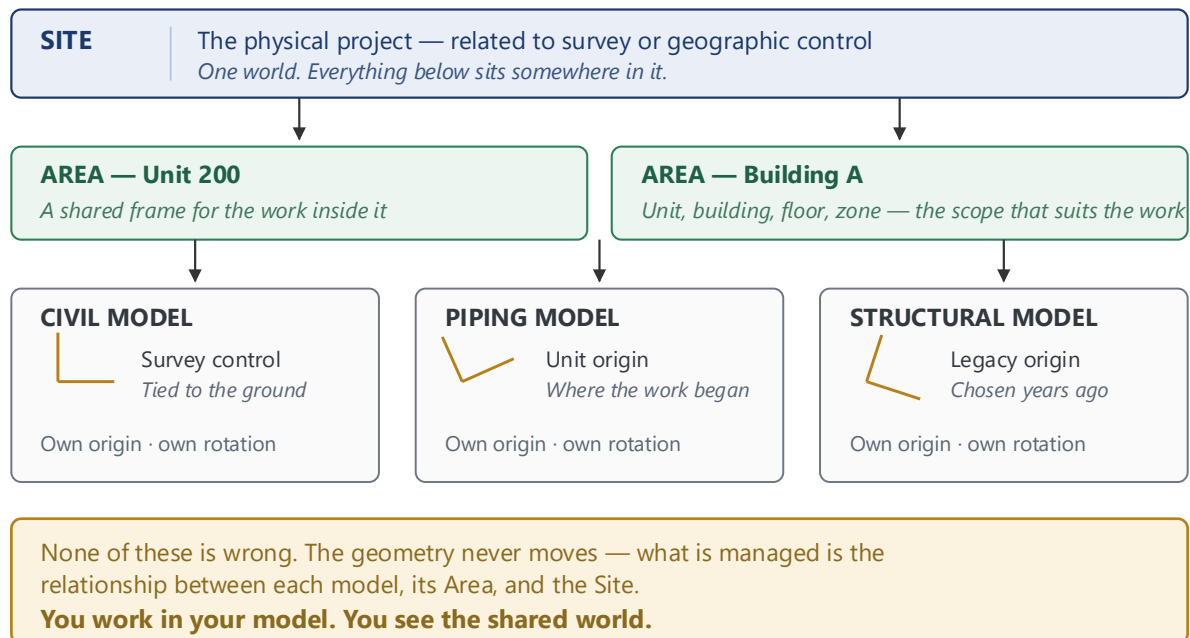
Those can be answered without forcing every participant into one coordinate system or one file structure.

At the broadest level is the Site — the physical project, with a relationship to survey or geographic control. Within it are Areas: units, buildings, floors, zones, or whatever scope suits the work. The terminology matters less than the idea. Models then become Members of an Area.

Each model keeps its own internal coordinates. Its geometry does not move. Its author does not rebuild it around a new origin. What is known instead is the relationship: model to Area, Area to Site. From those two relationships, the model can be placed correctly relative to everything else.

Agreeing on Space

Each model keeps its own origin. The relationship is what is shared.



Local

Coordinates Are Not Wrong

Site, Area, Member. Each model keeps its own origin; the relationship is what is shared. Coordinate differences are often treated as though somebody made a mistake. Sometimes they did. More often they are

useful. A discipline puts its origin where the work begins. A building model uses a local project system. A civil model relates directly to survey control. A legacy model has an origin that made perfect sense to its author.

The mistake is assuming one of them must be imposed on all the others before coordination can begin.

A transformation is usually the better answer. Position, rotation, elevation, and scale where units differ — applied without touching the original geometry. That is more than a technical convenience. It preserves responsibility. The civil engineer does not modify the piping model in order to position it. The piping engineer does not move the civil model. The environment manages the relationship; the models stay independent; the world becomes shared.

Seeing Is Not Owning

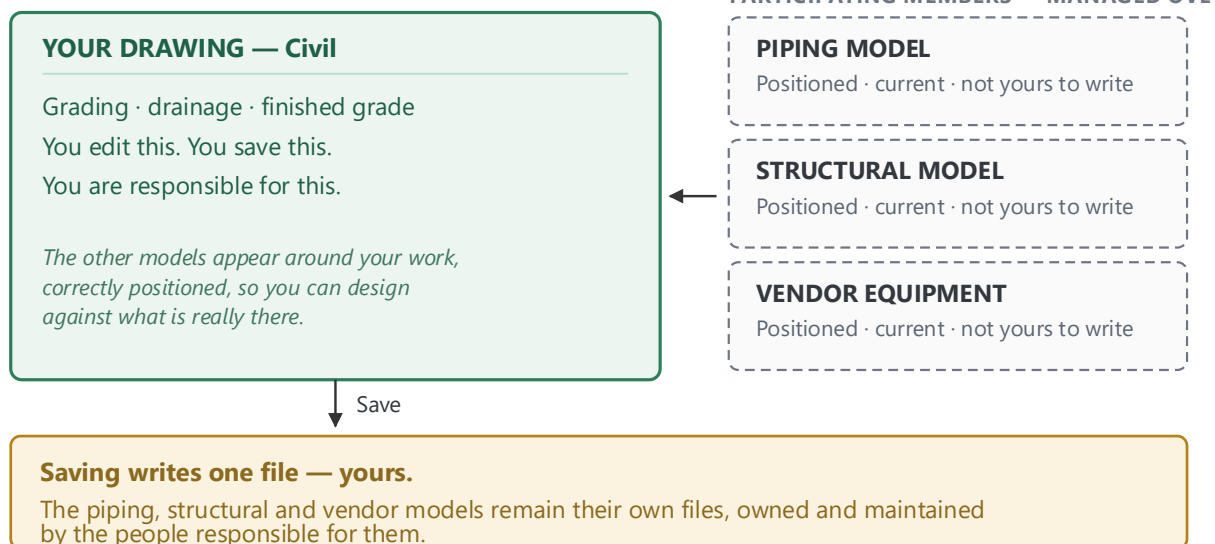
The civil engineer opens a drawing and sees the civil model as usual. The relevant piping, equipment and structures appear as well, positioned correctly. They provide context. The engineer designs around them.

But saving the civil drawing does not save anyone else's work. The engineer has not taken possession of the piping model, and the structural engineer's file has not become part of the civil engineer's responsibility.

Coordination should increase awareness without transferring ownership. A person can see something without becoming responsible for editing it. That is what allows federation to support independence rather than undermine it.

Seeing Is Not Owning

Context arrives in your drawing. Responsibility stays where it was.



It also changes how coordination actually happens. Much of it is discovery rather than formal clash detection. The engineer was not looking for the pump. It simply became visible, because the shared context revealed that it occupies the same world.

Other models arrive as context. Saving touches only the drawing you own. That discovery raises questions. Does it need a foundation? Does it affect drainage? Does it need access? Should someone else change their design? The shared environment has answered none of them. It has done something more basic: it made the relationship visible. Engineering decisions can only respond to conditions that are known.

Spatial ViZN

Spatial ViZN is one implementation of this approach — a way to work in a coordinated drawing set without managing each other's files.

A project establishes a Site. Teams create Areas for the scopes that matter to them. Models become Members. When a participant opens or refreshes a drawing, the other relevant Members load as managed overlays, positioned relative to the active drawing. The participant keeps working in their own file. The other models remain their own files. Saving does not modify anyone else's work.

Teams can also define shared Views — named extents, directions and scales — so one participant can say, in effect: look here. A repeatable point in a spatial conversation, without anyone's drawing becoming the permanent owner of the camera.

The implementation matters less than the principle. Shared spatial context should be available where people work — not only in a separate viewer, not only after publishing, and not only after assembling a new master model.

**You work in your model.
You see the shared world.**

What This Is Not

This is not a new master model. Nothing is merged, published, or taken into anyone else's custody.

It is not a viewer you have to leave your work to visit. The context arrives where the drawing is open.

It does not require every participant to adopt one coordinate system, or to rebuild a model around somebody else's origin.

And it is not clash detection. Clash detection asks whether two things collide. This asks the earlier question — whether you can see the other thing at all.

Spatial ViZN is not a replacement for Navisworks, ACC, a common data environment, or anyone's authoring tool. Those remain good at what they do.

The World Is the Common Model

Perhaps the mistake is thinking that the common model has to be a file.

The common model already exists. It is the physical world, and every digital model represents some portion of it. The goal of spatial coordination is not to merge those representations into one new object. It is to preserve their relationship to the world and to each other.

The pump is still in one place. The foundation is still under it. The drainage still has to move around it. The electrical connection still has to reach it. Different teams may represent those conditions differently, work in different applications, keep different coordinate systems, and never open the same source file. None of that changes the physical relationship.

Boundaries come from applications, disciplines, organizations and time. Space is usually where they first become visible. Something appears in my work area. It affects what I am doing. Now we have to coordinate.

So the simplest useful question may be:

Where is it, how does it affect my work, and who do I work with?

A shared spatial environment does not answer every engineering question. It does something more fundamental: it allows different participants to encounter the same world, and to recognize that something which was never part of their work now matters to it.

Which makes the next question unavoidable. The engineer can see the pump. They know where it is. They understand that it affects their work.

But what is it?

That is where the next paper begins.