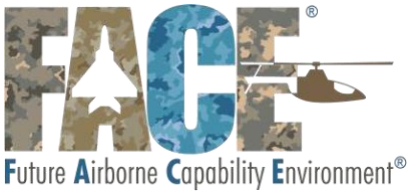

It's not a Sprint, It's an Ultramarathon

A Retrospective on 15 Years of Data Modeling

The Open Group FACE® and SOSA® Consortia 2025 Technical Interchange Meeting (TIM),

by:

Chris Allport
Co Founder, Skayl, LLC

Gordon Hunt
Co Founder, Skayl, LLC

Dave Lombardi
Principal Data Architect, Skayl, LLC

June 2025

Table of Contents

Executive Summary

This paper chronicles a 15-year journey of developing the FACE® Data Architecture, highlighting key milestones, technical challenges, and transformative insights along the way. What began as a vaguely defined effort quickly evolved into a structured, standards-driven discipline. The authors reflects on early challenges like vague terminology, ad hoc validations, and the lack of formal tooling – challenges eventually overcome through iterative testing, creative scripting, and cross-standard semantic comparison.

By chronicling the evolution of a data architecture that steadily dismantles traditional barriers to interoperability, this paper seeks to add a little context to the last fifteen years of data modeling. It touches upon some of the early successes and addresses some of the challenges with maintaining a standard. Blending the technical and anecdotal, this paper seeks to inform and amuse with a light touch.

Introduction

The very first meeting should have tipped me off. In hindsight, it probably should have sent me running for the hills, never to look back and wonder what that kindly professorial type was saying. This meeting created a singularity of sorts. And yet, while I call it a meeting, it was so much more. It was a combination of an interview, a collaboration, a training, and possibly the psychological equivalent of a deep cavity search. That said, I do not recall being offered a job. I'm rather certain I offered nothing to the discussion. Training – or the beginning of training – certainly did occur. And…my brain hasn't been right since.

> "But don't begin until you count the cost. For who would begin construction of a building without first calculating the cost to see if there is enough money to finish it?" – Luke 14:28

In the beginning (no, not that one), it was not entirely clear where we were going. We were defining the very words that made up the data architecture. Strike that – we were making up the words that made up this thing we were building. We did not even call it a data architecture yet.

Upon joining the effort, there was something called a "Message Type." At the time, I had no idea why this was so wildly unpopular. It was later renamed to "Interface Type Relation" (which we affectionately – or obediently – called ITR). As soon as the model was updated with the latest name, it changed once more to "View." Although the change itself was wildly unpopular, this much shorter name was highly preferred over its immediate predecessor.

After about three years, we had enough (knowledge, data, and imagination) to run a test that changed everything for us. More on that in a bit. The point is, we could not count the cost at the onset. We had no idea how long it would take to construct the building. If you had told me that I would spent the next 15 years of my life doing this, I would have confidently asserted that you were a liar and changed the topic.

Sprinting is fun. We see results. We put in some effort, we affect change, we take the feedback, we adjust, and then we do it again. Sometimes I think we created "agile" just so we can live from sprint-to-sprint.

And here we are with a decade and a half of data modeling experience behind us. It hasn't always been fast, nor has it always been slow. Statistically, one might say it was steady, but for those of us in the trenches, much of it has come in fits and starts.

In the spirit of this year, this paper seeks to find a little perspective in the journey and make a little sense out of Luke's admonition.

The Test

Most of our Data modeling experience started in a budget-friendly Unified Modeling Language (UML) modeling tool. Initially, we didn't have metamodels and validation was ad-hoc. We did not have tooling, nor were we allowed to develop tooling.

We were, however, allowed to write automation scripts inside the existing tooling. I think this was a "color of money" thing, but I don't really remember. Fortunately for us, this modeling tool had a scripting engine that would allow us to manipulate the models mechanically. This is where we define ad-hoc validation. Run a script until it dies. Figure out what's wrong with script…unless it's something wrong with the model. Wash. Rinse. Repeat.

Anyway, after about 3½ years of building models and basic scripts, we conceived of a demonstration (or test) that would validate our findings. To perform this test, we needed one more script and lots of patience. Lots of patience.

What did it do? It compared 5 standards with each other. UAS Control Segment Architecture, NATO Standardization Agreement STANAG 4586 Edition 4 Vol 1 (effectively Edition 2 Amendment 2), NATO STANAG 4586 Edition 4 Vol 2 (effectively Edition 3), the "Army Private Messages", and the Universal Command and Control Interface (UCI) (the message set behind Open Mission System (OMS)). By "compared," I mean that it checked the semantics of all the fields of all the messages in each standard against all the fields of all messages of all the other standards.

That first script took 14 hours to run.

But the result was incredible! Through purely automated means, we could see where standards aligned and where they differed. We could see where the standards talked about the same concepts but used different units. We could calculate a mapping from one standard to another and provide a listing of data needed to satisfy the mapping (along with equations to perform unit the necessary unit transforms). Manual standard-to-standard mappings were no longer necessary.

This work had been completed before the FACE® Technical Standard sought to adopt a data architecture. That came next.

A Series of Fortunate Events

In the realm of data architecture, the next couple of years were a blur of activity. The FACE Consortium convened a series of meetings to design the data architecture. The Data Architecture Working Group (DAWG) was born and started building the FACE Data Architecture The UAS Control Segment (UCS) standard was deemed sufficiently complete and transitioned to a commercial standard.

Hindsight being what it is, it is easy to throw darts at the first version of the FACE Data Architecture (released in earnest in FACE Technical Standard v2.1). Regardless of that, it introduced many improvements over the UCS architecture that inspired it.

The differences are fodder for nerds, geeks, and their ilk. They are equally good fodder for trolls. Regardless, the improvements resulted in a model that could stand alone – unsupported by diagrams. Up to this point, diagrams had been absolutely necessary to help decode the model. The model could only partially be interpreted without the diagrams. Labels applied to the diagrams themselves provided the necessary context.

Another substantial innovation was the FACE Conformance Test Suite (CTS). This introduced the idea that the model should be testable by a third party. Imagine that – a compiler for data models! This meant that all the information had to live in the model. The model format itself was the canonical form. Diagrams were supplemental and informative – no longer essential and normative.

This made it possible to develop tools to process models! Valid models could be loaded into tools – and for once, we could talk about what it meant to have a valid model. Models could be swapped and interchanged. The model format was managed rather than the entirety of the model content! This opened so many more doors for building and using models! And it offered so many more challenges for addressing model differences, model governance, and even model maintenance.

As we continued to work the standard, we quickly ran up against some of the limits of the original design. This led to the development of the query and template language that was affectionately dubbed "QBIRT." It took us several years to bring everything together but eventually FACE Technical Standard v3.0 was released with significantly more data architecture capability.

Even still, we continue to evolve and grow the standards (both FACE & UDDL® Technical Standards). They do not solve every problem yet, but we are continually making progress towards better.

Excellent Impact

The FACE Data Architecture (and the FACE Technical Standard itself) set out to solve many problems and it has done a remarkable job doing so. It has done fantastic job of providing a way to document the semantics of interfaces, so their meanings are clear and unambiguous. It provides a consistent representation such that models can be processed without cooperation between tool providers. There are not "proprietary" or "custom" extensions allowed in .face files. This allows for actual model interchange – something UML tool providers have struggled to achieve.

The value of the data model (a significant aspect of a Data Architecture) does not end with semantics. Fundamentally, the data model captures the information that passes through the interfaces of an application. In other words, it documents the data an application sends and receives. This means the data model can be compared with the actual application (in compiled, object code format) and there is automated confirmation (using the CTS) that the application matches the data model.

[Taking things one step further, CTS ensures applications only use allowed software interfaces. This deters developers from pretending to use the proper type-specific interfaces while sending the actual application data out a backdoor (this example is based on a true story).]

Unfortunate Shortfalls

At best, the FACE Technical Standard only acts as a deterrent for bad behavior – it does not prevent it. The trademark guidelines provide clear instructions for how one might claim conformance to the standard; however, time will tell if it has the teeth to make things stick.

At a recent conference for another standard, I asked their technical leads how they ensure cooperation among their participants. Their response? Peer pressure. When other participants join the network to communicate and have "proprietary" data, the other participants needle them until they disclose their data formats. While this was very likely to be true, I believe it only to be true within a cooperative lab environment. Regardless of how far this altruism extends, it demonstrates a really important point – compliance is only limited by a desire to cooperate.

We still see a significant reluctance to opening application interfaces. Do we really believe in MOSA? How far does that belief extend? Are we completely sold out for MOSA as long as our bottom lines are protected? As long as our vendor-lock is guaranteed?

The standard has limited ability to make people play nice. At least it tries. And where folks refuse to play well together, well, one hopes that the cutlines of modules in the FACE Technical Standard make it possible to detect (and, optimistically, correct) such behavior.

Going the Distance and Whatnot

In spite of all the remarkable success we had with data models in the early days, data modeling remains difficult. One of the most difficult challenges is finding professionals with the right combination of skills and knowledge to build the models. An effective data modeler requires a rare combination of subject matter expertise, technical skill, organized thinking, and exceptional memory.

The ideal modeler has all these skills, but it is challenging to find them all in a single person – especially one who can be dedicated to modeling (instead of running the project). While none of these skills are particularly optional, it is possible for a talented modeler to leverage a number of different experts to build a model. Of course, when that happens, the modeler requires an additional skill: the ability to extract the right information from the expert (in a useful form).

Fortunately, we are finally starting to see some movement in the realm of making data modeling easier. Subject matter expertise is getting much easier to access since technology allows us to build a proxy subject matter expert with the right collection of documents. Ostensibly, variants of this technology will make it easy to transform our myriad documentation into viable models.

However, the problem of model mismatch remains.

Model Mismatch

When thinking about model mismatches, I am reminded of an ancient Buddhist parable. The story tells of a group of blind men who seek to learn more about this strange animal – an elephant – that has been brought to their town. Each man experiences a small part of the animal and forms his idea of the creature from his limited perspective. The trunk is mistaken for a snake. The tail is mistaken for a rope. The leg – a tree trunk. The side of the elephant – a wall. And the tusk – a spear.

While all the men accurately described the elephant, they did so from only a single perspective. When we attempt to merge data models, we tend to find very a similar challenge.

When a data model is documenting a C2 (command and control) standard, the description of an electro-optical camera (video camera) might seem trivial. There are a number of high-level features one might expect to find on a camera: focus, zoom, color mode, gain, and then all the pointing parameters.

What if you are modeling the internals of a camera system? Focal plane? Lens? CCD? Suddenly, there are countless more components (and substantially more technical complexity) to the data model.

This causes a problem when we attempt to integrate systems built with such drastically different models. When comparing two different models, we often look for similarities and commonality. When the level of detail varies (as with the example), there is a one-to-many relationship. That is, one part (in the less detailed model) may be broken into many (in the more detailed model). This may make it difficult to assert any level of equivalence. This problem is exacerbated when the difference is based on modeling approaches or philosophy.

Beyond the Wall

When running a marathon, it is possible to deplete your available energy and "hit the wall." This is a point at which you are completely exhausted, but your press forward to the finish line, struggling to make progress with each step. There are a lot of parallels to working on a standard. After expending all the energy on the release cycle, maintenance begins. And for a while, it's just a slow and steady beat.

Looking back, it's clear that the decisions we made – sometimes blindly, sometimes stubbornly – echo the themes explored in the *Sustained Innovation Through Composable Systems* paper. That paper makes the case that real, lasting innovation doesn't come from flashy breakthroughs, but from building systems that can evolve – piece by piece, function by function. It's comforting (and a little validating) to see that our slog through semantic clarity, standardized model formats, and testable conformance wasn't just good engineering – it was laying the foundation for exactly that kind of composability. They call it a strategy. We called it a survival tactic. But the end goal is the same: building systems that aren't just interoperable once – they stay that way, even as the world around them keeps changing.

Fortunately, there has been a lot of engagement with folks really digging in to move some significant initiatives forward. The team has caught their second wind and is making steady progress to move the standard forward!

Ongoing effort will make substantial improvements to the Query-Template language: decoupling the two from each other so we can better represent other standards like OMS. There is also an effort to relate disparate models (even when they are modeled at different levels of detail). And we are even exploring a possible way to model "observables" to make the Shared Data Model (a managed collection of data model "building blocks") extensible without requiring CCB involvement.

Beyond FUD – Focus. Usable. Drive.

Somedays, it seems like we are only scratching the surface of what is possible with the FACE Technical Standard. There are many technical challenges out there that threaten to pull our attention in another direction. To that end, we must stay focused. Focus will help us ensure the standard evolves and effectively solves the problems in its scope.

Usability will help dispel some of the naysayers. Some – not all. (Remember, naysayers say nay. They must say nay, for if they did not, they would not be naysayers. Also, naysayers are not to be confused with those who merely say, "nee.") Yes, data modeling is hard. Maybe it doesn't have to be. Let's tackle complexity with clarity and reject job security through obscurity. Making hard stuff look easy? That's real engineering.

And finally, let's drive. Let's push forward, not with hesitation, but with confidence. Let's deliver real progress and make this standard the standard – for data, for semantics, and for the future of interoperable systems.

Referenced Documents

S251	Sustained Innovation Through Composable Systems, Guertin, Hunt, Matthews, May 2025, S251, https://event.nps.edu/conf/app/researchsymposium/unsecured/file/1156/SYM_Panel%20%232_Guertin%20Paper_SYM-AM-25-301.pdf
C231	Open Universal Domain Description Language, Edition 1.1 (C231), published by The Open Group, March 2023; refer to: www.opengroup.org/library/c231
C232	FACE® Technical Standard, Edition 3.2 (C232), published by The Open Group, August 2023; refer to: www.opengroup.org/library/c232
NS4	NATO Standardisation Agreement (STANAG) 4586: Standard Interfaces of a UA Control System (UCS) for NATO US Interoperability. Brussels: NATO Standardization Agency. April 2017. https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=275263
AS4	Unmanned System (UxS) Control Segment (UCS) Architecture. SAE International. March 2022. https://www.sae.org/standards/content/as6512/

About the Author(s)

Chris Allport is a co-founder of Skayl, a Maryland-based firm dedicated to solving the toughest interoperability challenges in aerospace, defense, and beyond. With nearly three decades of experience in software engineering, system integration, and standards development, he is best known for turning complex concepts into real, working prototypes. As chair of the Shared Data Model subcommittee, Allport has spent much of his recent career tackling the nuanced and often painful problems of semantics, structure, and system boundaries. In his early years, he attacked integration problems with point solutions and sheer optimism. Now, apparently tempered by time, he approaches them with a growing appreciation for scalable architecture.

Gordon Hunt Gordon Hunt is a co-founder of Skayl. His focus is on system of systems integration and semantic data architectures which enable increased systems flexibility, interoperability, and cyber security. Gordon's experience in building real systems with current and emerging infrastructure technologies is extensive. His technical expertise spans embedded systems, distributed real-time systems, data modeling and data science, system architecture, and robotics & controls. He is a recognized expert in industry on Open Architecture and data-centric systems and is Skayl's primary consultant for distributed real-time and service-oriented architectures and implementations. As a regular author and presenter, he speaks frequently on modern combat-system and command and control architectures and is an active member in numerous professional communities.

David Lombardi is a Principal Data Architect for Skayl. Renowned for his expertise in advanced data modeling for the past 14+ years, Dave has successfully delivered mission-critical connectivity solutions in some of the most demanding environments, including NATO, the military, and global tier-one aerospace and defense companies. Dave excels in forging enduring, trust-based partnerships with clients and stakeholders, while optimizing internal business processes to achieve top results. A stellar technical communicator, Dave presents to and trains engineers and architects across the globe, empowering them to lead in the extraction of complex business needs and transforming them into innovative data solutions.

About The Open Group FACE® Consortium

The Open Group Future Airborne Capability Environment® Consortium (the FACE® Consortium), was formed as a government and industry partnership to define an open avionics environment for all military airborne platform types. Today, it is an aviation-focused professional group made up of industry suppliers, customers, academia, and users. The FACE Consortium provides a vendor-neutral forum for industry and government to work together to develop and consolidate the open standards, best practices, guidance documents, and business strategy necessary for acquisition of affordable software systems that promote innovation and rapid integration of portable capabilities across global defense programs.

Further information on the FACE Consortium is available at www.opengroup.org/face.

About The Open Group SOSA® Consortium

The Open Group SOSA® Consortium enables government and industry to collaboratively develop open standards and best practices to enable, enhance, and accelerate the deployment of affordable, capable, interoperable sensor systems. The SOSA Consortium is creating open system reference architectures applicable to military and commercial sensor systems and a business model that balances stakeholder interests. The architectures employ modular design and use widely supported, consensus-based, nonproprietary standards for key interfaces.

Further information on the SOSA Consortium is available at www.opengroup.org/sosa.

About The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through technology standards and open source initiatives by fostering a culture of collaboration, inclusivity, and mutual respect among our diverse group of 900+ memberships. Our membership includes customers, systems and solutions suppliers, tool vendors, integrators, academics, and consultants across multiple industries.

The mission of The Open Group is to drive the creation of Boundaryless Information Flow™ achieved by:

- Working with customers to capture, understand, and address current and emerging requirements, establish policies, and share best practices
- Working with suppliers, consortia, and standards bodies to develop consensus and facilitate interoperability, to evolve and integrate specifications and open source technologies
- Offering a comprehensive set of services to enhance the operational efficiency of consortia
- Developing and operating the industry's premier certification service and encouraging procurement of certified products

Further information on The Open Group is available at www.opengroup.org.