THE Open GROUP

FACE™
Future Airborne Capability Environment

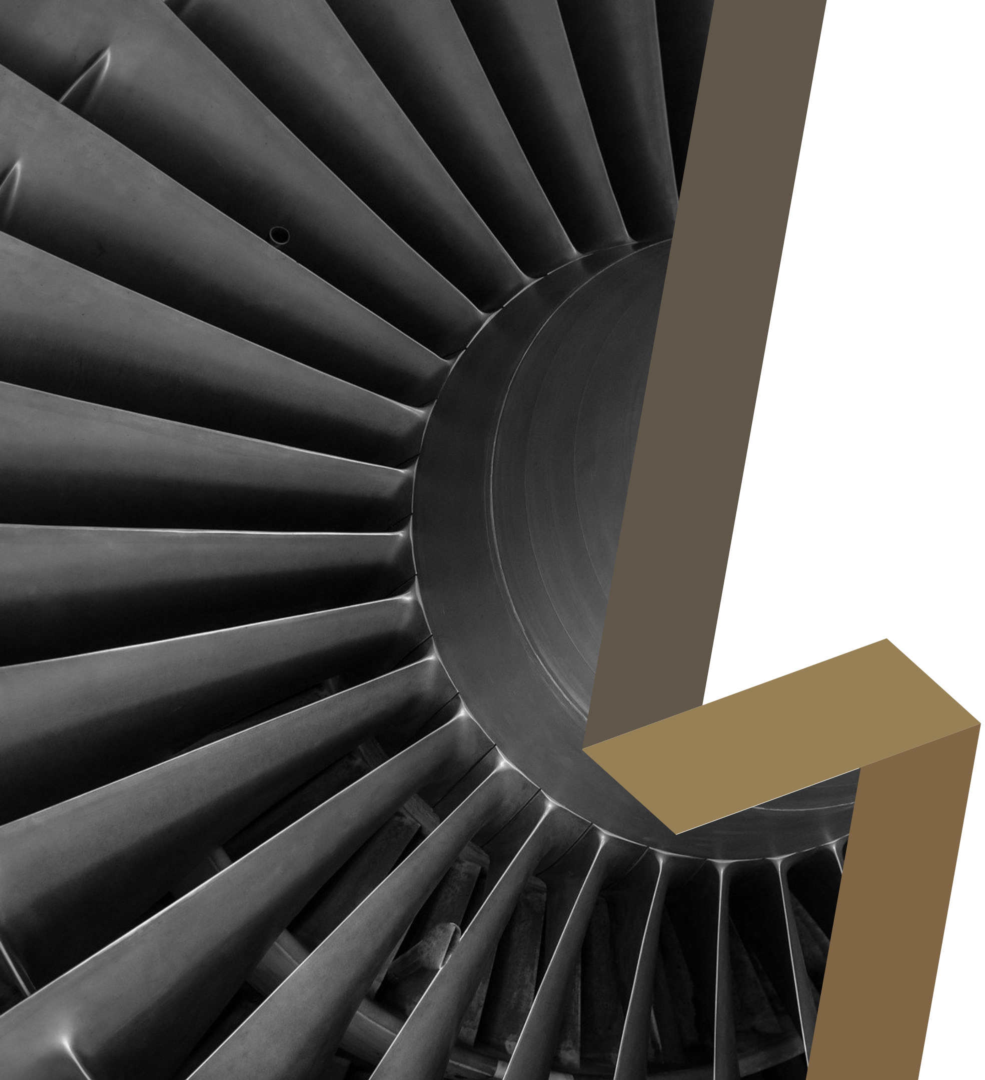
SOSA™
Sensor Open Systems Architecture

U.S. Army FACE™ & SOSA™ Technical Interchange Meeting

Huntsville Alabama
September 14, 2021

U.S.ARMY

PROGRAM EXECUTIVE OFFICE AVIATION



FACE Data Architecture Overview

James “Bubba” Davis, Ph.D.
L3Harris, FACE DIOG Chair

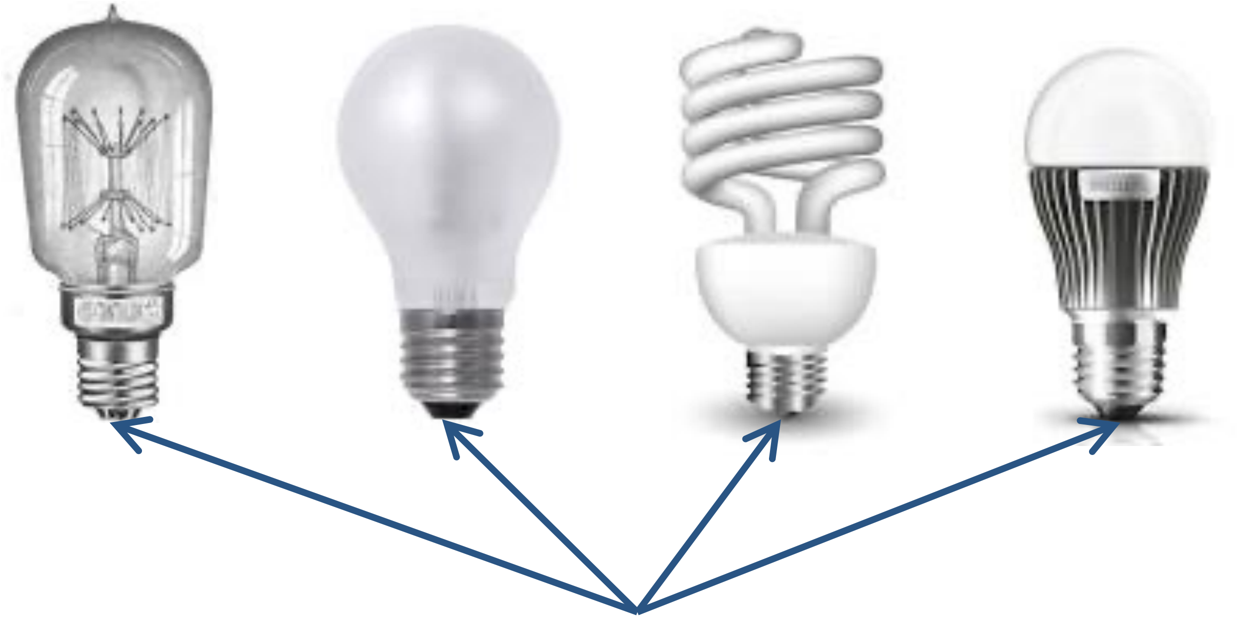
Gordon Hunt
Skayl, FACE DIOG vice-Chair

Warfighting capabilities are *increasingly* dependent complex software integrations that span multiple platforms, operators, and maintainers.

But… the rate at which we can deploy and maintain these systems safely and accurately is *decreasing*, and the complexity isn’t going away

We Need Open Interfaces to Support Large Scale Integrations

What is standardized on all of these devices?



The Interface!

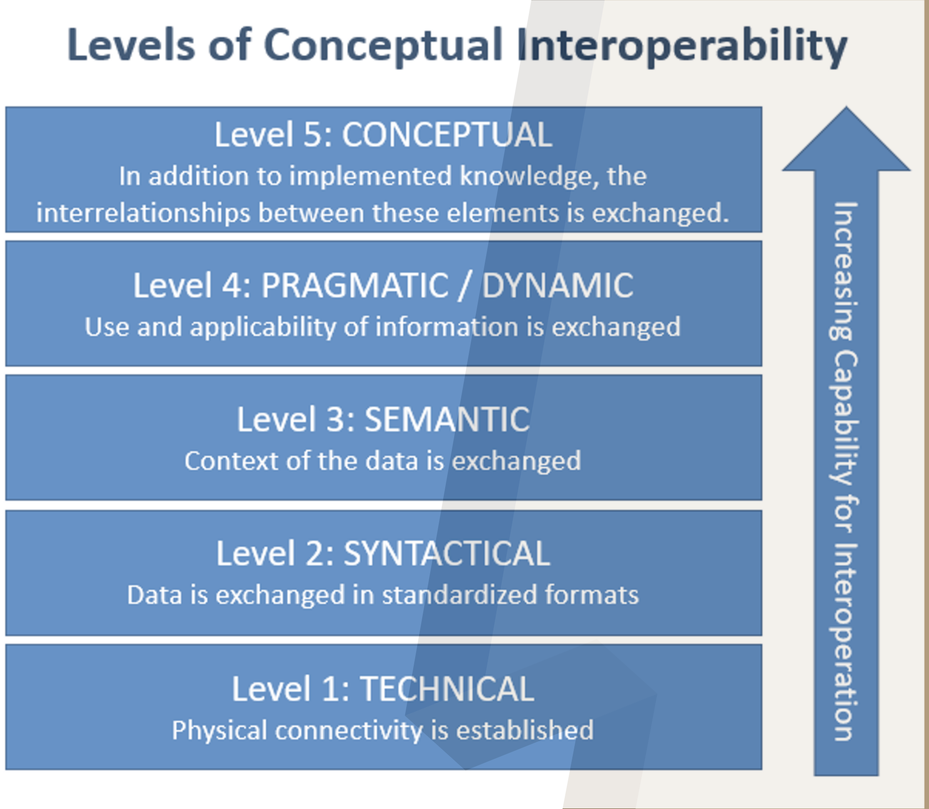
ANSI standard C81.67 and IEC standard 60061-1

What Enables and Supports Integration and Interoperability

Current Integration Practice

- Most programs/platforms today at Level 1 or 2
- Current commonality-based mandates address syntax in automatable forms
- Gaps
  - Where and how are semantics documented?
  - How are behavioral expectations captured?

Can we scale these approaches?



Levels of Conceptual Interoperability

Level 5: CONCEPTUAL
In addition to implemented knowledge, the interrelationships between these elements is exchanged.

Level 4: PRAGMATIC / DYNAMIC
Use and applicability of information is exchanged

Level 3: SEMANTIC
Context of the data is exchanged

Level 2: SYNTACTICAL
Data is exchanged in standardized formats

Level 1: TECHNICAL
Physical connectivity is established

Increasing Capability for Interoperation

(Tolk, 2004)

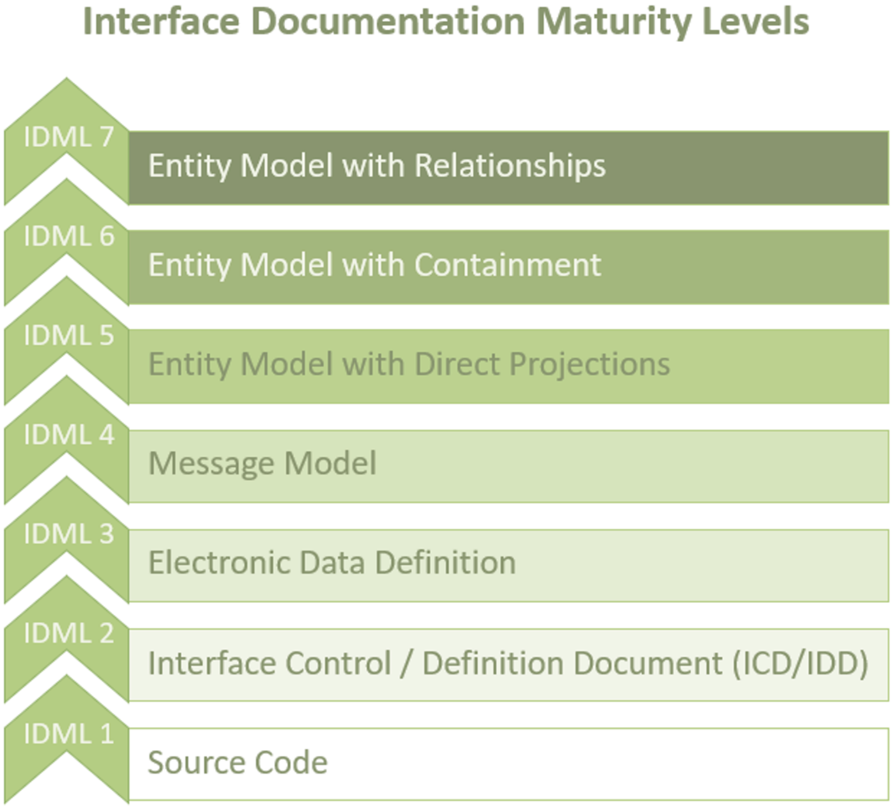
Towards Automating Implementing Interoperability

What is it?

- Additive model to describe the fidelity of interface documentation for automation
- Documenting semantics
- Most programs today between 2 and 4

What can you do with it?

- The rigor and specificity of interface documentation enables management, scalability and extensibility of the integration infrastructure.
  - Code generation from syntax
  - Mediation of format and representation
  - Traceability of ‘like’ concepts across interfaces
  - Determination of semantic equivalence



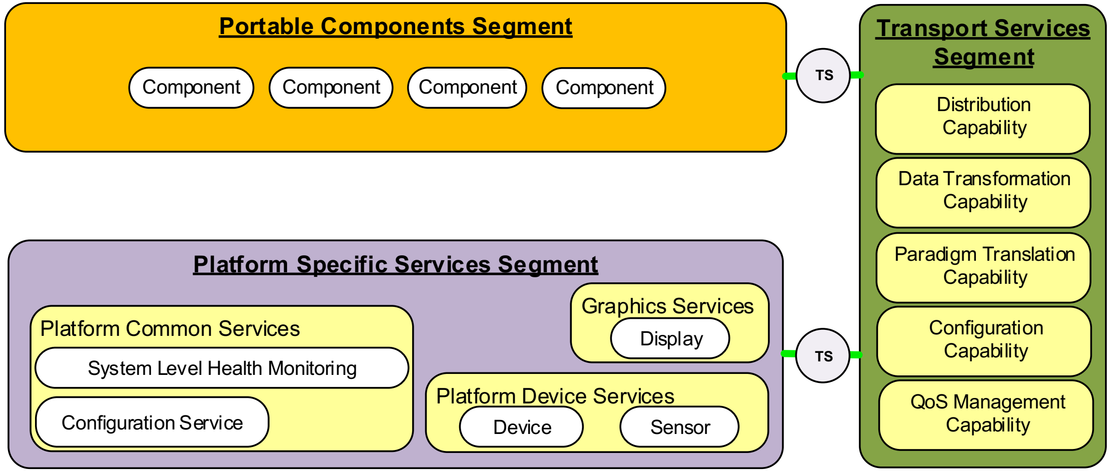
Interface Documentation Maturity Levels

IDML 7	Entity Model with Relationships
IDML 6	Entity Model with Containment
IDML 5	Entity Model with Direct Projections
IDML 4	Message Model
IDML 3	Electronic Data Definition
IDML 2	Interface Control / Definition Document (ICD/IDD)
IDML 1	Source Code

(Hand, Lombardi, Hunt, & Allport, 2018)

What is the FACE Data Architecture?

Relevant FACE Architectural Segments

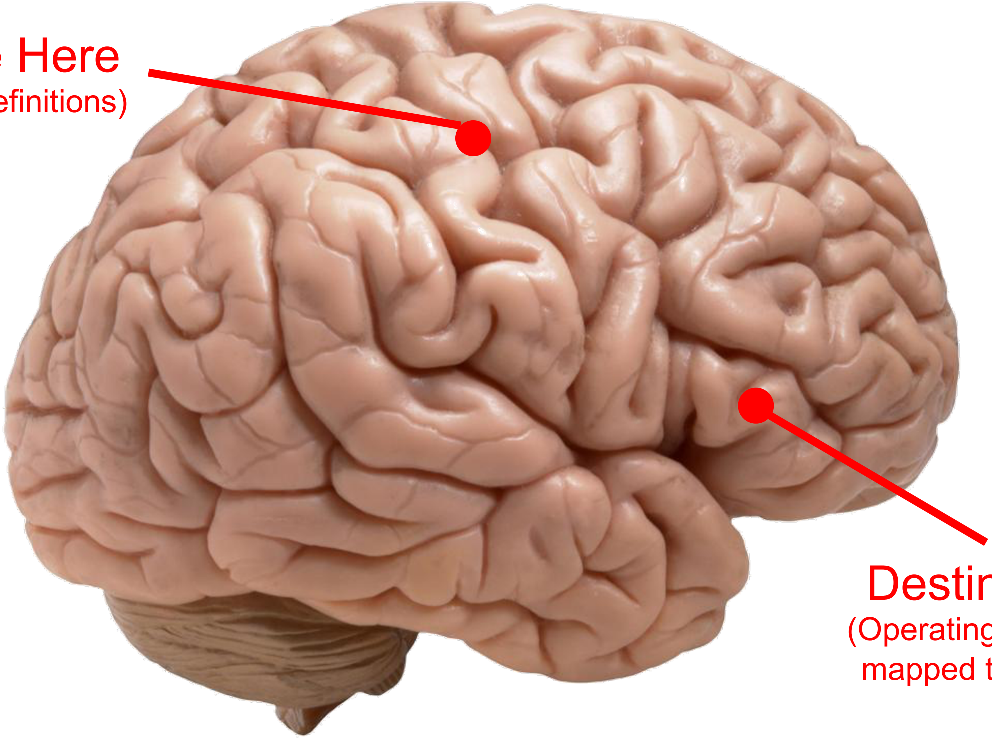

Portable Components Segment
Component
Component
Component
Component
TS
Platform Specific Services Segment
Platform Common Services
System Level Health Monitoring
Configuration Service
Graphics Services
Display
Platform Device Services
Device
Sensor
TS
Transport Services Segment
Distribution Capability
Data Transformation Capability
Paradigm Translation Capability
Configuration Capability
QoS Management Capability

All data exchanged via the TS interface must be properly data modeled using the FACE Data Architecture

- **Portable Components Segment (PCS)**
  - Portable Applications
  - Portable Common Services
- **Transport Services Segment (TSS)**
- **Platform Specific Services Segment (PSSS)**
  - Platform Device Services
  - Platform Common Services
  - Graphics Services

FACE Data Modeling Requires a Different Perspective

You Are Here
(data type definitions)



Destination
(Operating Context mapped to views)

FACE Data Architecture Objective

- **Describe the data** going into or coming out of a software component, in the context of the **entities of concern to the software component,** to **enable an integrator to combine software components** to provide a larger capability
  - In laymen's terms: describe concepts we want to communicate about well enough for everyone to clearly understand what we mean.

- Capture the semantics of data exchanged in a rigorous, machine processable format

FACE Data Architecture Elements

- The FACE Data Architecture consists of:
  - Data Model Language
  - Rules for the construction of UoP Supplied Models (USM) and Domain Specific Data Models (DSDM)
  - A set of Data Model Language bindings that map Data Model Language elements to each of the supported programming languages (C, C++, Ada, & Java)
  - The Shared Data Model (SDM)

Benefits of the FACE Data Architecture

- Supports Open Interfaces
  - Rigorously documents the semantics of data exchanged
  - Based on open standards
    - FACE Technical Standard defines the data centric APIs
- Provides for vendor independence
- Aids in breaking vendor lock

Who Benefits from the FACE Data Architecture?

- Programs benefit via open, data centric interfaces which aid in addressing vendor lock
- Industry benefits by additional avenues for competition
- System integrators benefit from rigorous, well-defined, standards-based software interfaces
- Software developers benefit during maintenance and sustainment
  - Initial software development requires an investment in data modeling

FACE™
Future Airborne Capability Environment

Why should I adopt the FACE Data Architecture?

- Are you concerned with integration and sustainment?
- Are your data semantics important?
- Do you have a MOSA requirement?
- Do you want open interfaces for your software components?
- Do you want standards-based, instead of proprietary, mechanisms for documenting your data?
- Do you need to mitigate vendor lock?

Future Airborne Capability Environment

Thanks!

Any questions?

You can find us at:

bubba.davis@l3harris.com
gordon@skayl.com