Embedded Systems INSTITUTE

STEVENS Institute of Technology

Architectural Descriptions and Models

White Paper Resulting from Architecture Forum Meeting

March 21-22, 2006 (Washington, USA)

Edited by:

Dr. Gerrit Muller, Embedded Systems Institute

Mr. Eirik Hole, Stevens Institute of Technology

Input was provided by the following participants in the Architecture Forum:

Name	Organization
John Bagley	Raytheon
Ari Herva	Nokia
Eirik Hole	Stevens Institute of Technology
Jouko Junkkari	Nokia
Eric Kreuwels	FEI Company
Bjørn V. Larsen	Kongsberg Defence & Aerospace
Hugo van Leeuwen	FEI Company
Robert L. McCaig	Asset Inc.

Name	Organization
Peter van de Meulen	Philips
Gerrit Muller	Embedded Systems Institute
Rolf Siegers	Raytheon
Lauri Ståhle	Nokia
Andy Turner	Nokia
Dinesh Verma	Stevens Institute of Technology
Charles Weber	Homeland Security Institute
Mark Weitekamp	ANSER

Published on August 1, 2006

1. Introduction

Seven companies and two institutes discussed the state-of-practice of systems architecting during a two-day forum. The objective of the forum has been formulated as follows:

The forum will have an emphasis on practical systems architecting and the application of architectural information and knowledge. The objective is to provide a venue for the exchange of practical experience in the realm of development, implementation and management of system and enterprise architectures. This shall in turn be a platform for the exchange of ideas for improved practices in the above areas as well as the goal-oriented use of architectural knowledge and information in various life cycle phases and enterprise functions.

Participants in the System Architecture Forum are selected to be non-competitive and from different domains. In this second meeting the following domains were present:

Defense, Government and Space systems, Healthcare equipment, Measurement equipment, Consumer electronics, Telecommunications and semiconductors. The representatives of the participating companies are either practitioners themselves or managers that have lots of practical experience.

While discussing architectural descriptions and models it becomes immediately clear that:

- A tremendous amount of architectural models *can* be made; DoDAF (DoD 2003) defines 26 artifacts, (MoDAF adds another 9 artifacts to these), an adapted RUP method used 11 artifacts. For both methods many missing views were identified that require still other artifacts.
- A core set of models is common across domains.
- A practical description focuses on about 10 to 12 artifacts
- It is a challenge to find the "right" level of detail for these artifacts

The forum participants have agreed that the research fellows from the initiating institutes will start the codification of systems architecting know-how by producing theme-based white papers and by capturing best practices and heuristics, based on the discussions during the forum.

2. Exploring architectural descriptions and models

The forum participants were initially asked to list the different models and artifacts that were needed to document an architecture based on their experience. Then three seed presentations presented three approaches to documentation and modeling from the perspectives of defense and aerospace systems, different architecture frameworks with emphasis on the DoDAF, and finally from the perspective of complex mechatronic systems (electron microscopes) in the commercial domain. With this background, groups split out in breakout-sessions to discuss this more in detail, and elaborate on these suggestions

Fact finding

Two different processes and the related artifacts are used as a starting point to explore architectural descriptions and models: REAP, the Raytheon Enterprise Architecture Process based on DoDAF, and a process from Assett based on RUP. Appendix C provides an overview of available frameworks. In addition the principal architect of FEI performed a bottom up analysis within FEI of available documents and the perceived architectural value of these documents by the engineering community.

The Assett process and artifacts

The process described by Bob McCaig (Assett) more or less extends the 4+1 Kruchten views (Kruchten 1995) with a number of artifacts that support the execution of the project. Most notably *cost*, *WBS* (Work Breakdown Structure), *schedule* and *test documentation* are added explicitly. All artifacts and a documentation diagram are shown in Appendix A. In the documentation diagram some more artifacts are shown, which are not part of the architectural description in Assett's process. Some architects view artifacts describing *mission area*, *goals* and *objectives* as essential part of architectural descriptions, because these aspects are the driving force for specification and design and provide the rationale and justification for most decisions.

REAP and DoDAF

The *Raytheon Enterprise Architecture Process* uses the *framework- products* defined in DoDAF. DoDAF is the (USA) Department of Defense Architecture Framework. Appendix D shows all 26 framework-products defined by DoDAF. The DoDAF standard is limited on purpose to a catalogue of framework-products. It does not impose any specific way-of-working, this standard is less heavy than many process based standards <any example of MIL or DoD standard that is all encompassing is welcomed>. The products are grouped in 4 main views, as shown in Figure 1.

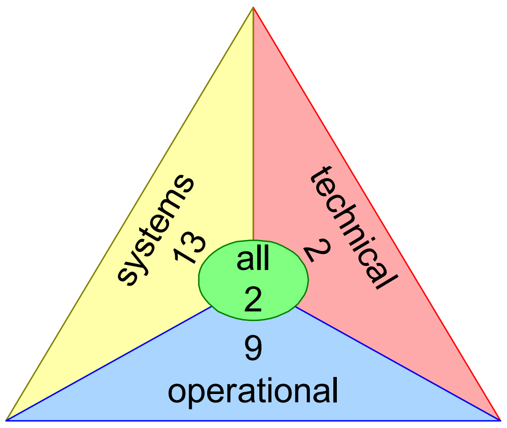


Figure 1. Main DoDAF views. The number indicates the amount of framework-products in this view.

Views and models at FEI

The principal architect at FEI sent out a questionnaire starting with the question: "What are the one or two most important architectural documents / descriptions / tooling that you have encountered?". About nine different types of models were identified as the most important, see appendix E. The principal architect also classified most important architectural documents in two dimensions: discipline (software, hardware, mechanics), and the degree of internal or external orientation, see Figure 2.

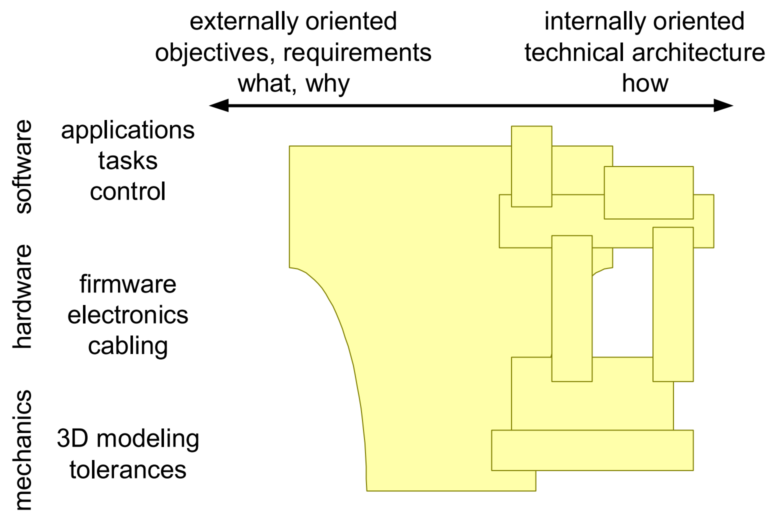


Figure 2. Classification of most important FEI architectural documents, as perceived by FEI engineering community

Discussion: What is a common and what is missing?

Two of the questions that were discussed in break-out sessions:

- What models did you miss?
- What models would you drop?

The term model is used here in a broad sense with the emphasis on *what* need to be described, rather than *how* it is described. All discussions groups identified many models to be added. Many missing models are related to:

- Business (financial, barriers to entry, industry standards, business value, risks, roadmapping)
- Domain (customers, applications, products, infrastructure, customer goals and objectives, end-to-end mission thread, prototypes)
- Physical world (continuous system, time, physical behavior and performance)
- Qualities (security, safety, reliability, evolution, et cetera)
- Process and organization (verification and validation, risks, configuration management, iteration)

- Technical (interfaces, schematics, data, implementation, roadmapping)
- Architecture Evolution (strategy, road mapping, guiding principles)

At the same time the consensus in the discussions is that too many models/descriptions decrease the project focus and overview. From architecting perspective *breadth* of descriptions is more important than *depth*. The amount of detail in architectural descriptions must be limited and *analysis paralysis* should be avoided.

The forum tended towards the notion of a core set of views that should be addressed in some way or another in any system development, although we were not able to identify these views at this point. It was also discussed that this core set might be dependent on the type of system to be developed, the regulatory and business context and so on. This core set would then be augmented by additional views within individual industries, companies and programs according to their needs.

Tensions and conflicting needs

An additional question posed to the breakout teams was:

- Identify tensions and conflicting needs.

Stakeholder orientation

The preferred presentation model depends strongly on the stakeholder. An example that was mentioned was that you might have to communicate performance aspects of an architecture completely differently to a user/customer, than to the affected developers. However, adaptation of the presentation to different stakeholders will cause new problems related to miscommunication and transformation losses. A general guideline is that the creator ("producer") of the model must adapt to the needs of the users ("consumers") of the model.

Size versus completeness

The users of the architecture description will always be able to ask more than has been described. At the same time users always complain about the status of the documentation and the lack of updates. There is a clear tension between the time and effort needed to create and maintain an architectural description, and the completeness and level of detail of the description. As indicated in the previous section breadth is more important for an architectural description than depth. The real challenge for the architect is to "right-size" the architectural description.

At product family or portfolio level additional means can be applied to get manageable architectural descriptions at product level. For example reference architectures described at such a higher level set the scope for product level descriptions. Roadmapping can also help, for instance to avoid pile-ups for the next release(s). Roadmapping is also a very powerful tool for (architecture) evolution.

Consistency of descriptions

The consistency of the different models was also the subject of a heated debate. An explanation of IEEE1471 (IEEE 2000) provided at the WICSA 2001 (Hilliard 2001) emphasized that consistency of all views and models is not always possible and desirable. Each view is an abstraction designed to emphasize a certain aspect of the system and thereby lacking in other aspects. From the standpoint of one part of the system it is therefore unlikely that the different views will align and integrate perfectly.

Inconsistencies in the descriptions are of course not desirable, but they do not necessarily mean that the integrity of the system is endangered. It does raise concerns however. How do we know what inconsistencies are a real issue versus just a result of omissions or ambiguities in the descriptions? How can we deal with this in a way that reduces the probability of real issues falling through the cracks? Currently it is left to the experience and intuition of the architect, as well as the individual engineers, to make sure that inconsistencies, ambiguities and omissions are handled effectively.

[Frank 2006] points out that "tolerance for ambiguity" is a desirable competence of system engineers. System architects and engineers operate in an environment full of uncertainties, unknowns, conflicts and inconsistencies. Inconsistencies that threaten the project success must be resolved as early as possible. While minor inconsistencies between the different views may be tolerated, one should at least seek to keep each individual view internally consistent.

Responsibilities of the architect

The architect responsibilities overlap with many project members. Especially the boundary between project leader responsibilities and architect and between product manager and architect is less sharp than most people realize. For instance cost, effort, WBS, and schedules are all somehow driving and/or being driven by architectural decisions, although the formal responsibility in most organizations resides with the project leader. Many of the missing models discussed in the previous section are part of this gray-area of overlapping responsibilities.

Standard versus need driven

The participants come from different domains, ranging from defense to consumer electronics. These different domains have a different approaches: standards or process driven, often mandated by customers such as the government or more pragmatic risk or need driven in case of companies that make catalogue type products.

Level of formalism

The architectural descriptions used in practice range from highly informal to formal. The level of formality depends on the stakeholders involved, the homogeneity of the subject described and the distance to the actual implementation. Some stakeholders are incapable of understanding formal specifications, for example end-users, while other stakeholders, such as expert engineers, need a high level of formality. Well-defined and homogeneous problem areas, for instance protocol stacks, lend themselves for formal descriptions. Inhomogeneous or less well-defined subjects require less formal, more creative, description strategies. In general, if a description is closer to implementation then this description will be more formal.

The use of frameworks

It was mentioned that the use of frameworks easily degrades from a benefit into an impediment, see Section 3.

3. Analysis and recommendations

Number and type of architectural models needed

The discussions throughout the meeting confirmed to a large extent the findings of the first seed presentation. The critical aspects of the architecture of a system can be described in about 10 different architectural views. There seems to be a sweet-spot around 10 views where the need for breadth is balanced by the need for focus and the human capability of relating multiple heterogeneous views. The actual views chosen might be dependent on the type of system to be developed as well as the domain and environment it is developed in.

See http://www.architectingforum.org/whitepapers/SAF_WhitePaper_2005_1.pdf for a short summary of views and viewpoints according to IEEE 1471.

Principle 1:

The essence of a system can be captured in about 10 models/views

The challenge of the right level of detail

The forum concluded that the level of detail that should be documented was dependent on many factors. One major driver would be risk, depending on the maturity of technologies used as well as the novelty of the solution in general. This risk driven approach seemed to be prevailing among the commercial participants. The more government oriented participants tended to find themselves in a situation where the customer would have quite stringent requirements on what should be documented. This also made these descriptions part of the program deliverables to their customers, whereas the commercial industry would chose to document and model architectures for mainly internal purposes and benefits.

The rightsizing of an architecture description is being perceived as one of the core challenges for system architects at this moment. The forum scheduled this subject for the October 2006 meeting.

Diversity of architecture descriptions and models

Choosing the right "notation" or the right means to consolidate architecture information is a critical challenge. What is the right level of formality? Are well-defined formalisms available that fit the stakeholder needs? What language should be used, plain informal English, more formalized natural language, or a rigid formal description language? What kind of schemata fit the description needs?

The conclusion of the discussion is that a diversity of architecture descriptions and models is needed. The effectiveness of architectural descriptions may not be hampered by dogmatic unification or standardization. No small unified description and model formalism exists that covers the wide variety of needs and situations. It is observed that the level of formalism increases when we move closer to the implementation.

Principle 2:

A diversity of architecture descriptions and models is needed: languages, schemata and the degree of formalism.

Principle 3:

The level of formality increases as we move closer to the implementation level.

Frameworks and standards

The core of the discussion was focused on practical use and needs. What needs to be described, what kind of model or representation to use? Nevertheless, many participants did have some hesitance to start the discussion about architectural descriptions. The reluctance is caused by traumatic experiences with extensive architecture frameworks and standards that have been imposed in the past promising to solve all architecture needs. In practice these extensive frameworks turn into a burden: creating a lot of work for architects without offering much support for their actual task.

Root cause of failing frameworks is the diversity and heterogeneity of architecting work. The single unifying architecting framework does not exist, each architecting challenge needs its own tuned approach. Does this mean that architecting experiences and approaches cannot be shared and trained? No, we can share and train in a valuable way. However, architects always need to adapt approaches from other domains to fit in their own particular domain. The real value for architects is to share several different approaches in different domains. In this way architects develop a rich collection of approaches as reference for future work. This discussion lead to the following principle:

Principle 4:

Architecting education must be framework and standard agnostic, but architects must have seen or used multiple frameworks and standards.

Most frameworks and related standards suffer from overweight to cover a wide area of applications. They can therefore easily become an overkill and impediment if not used with care. Most frameworks therefore provide for, and encourage tailoring. Programs that are not required to adhere to specific architecture frameworks often have a leaner, more pragmatic approach to architectural descriptions driven by need and risk-level. One of our challenges is to find common essential elements to incorporate these in architect training programs.

4. Conclusion

Architectural descriptions require balancing acts in many directions:

- Depth versus breadth
- Stakeholder interests, from technical expert to (naïve) consumer
- Degree of formalism, from controllable and verifiable to understandable and usable
- Pragmatic emphasis on describing and communicating essential architectural principles and choices versus consistency and completeness of an all-encompassing description

These balancing acts are driven by the situational context. Nevertheless, in all cases an optimum of 10 to 12 architecting views is perceived as optimal. More views create too much chaos, less views oversimplifies the situation.

The next meeting of the forum will therefore go deeper into the challenge of right-sizing the architectural descriptions according to the characteristics of the system to be developed and the environment it is being developed under.

Literature

[Frank 2006] "Knowledge, Abilities, Cognitive Characteristics and Behavioral Competences of Engineers with High Capacity for Engineering Systems Thinking (CEST)"; by Moti Frank, *Systems Engineering*, Volume 9 Number 2, Summer 2009

[Hilliard 2001] "IEEE Std 1471 and Beyond"; by Rich Hilliard, SEI's First Architecture Representation workshop 2001 http://www.enterprise-architecture.info/Images/Documents/IEEE 1471- Beyond.pdf

[DoD 2003] "DoD Architecture Framework, Volume 1: Definitions and Guidelines", Version 1 US Dept. of Defence, 2003.

[IEEE 2000] "IEEE Recommended practice for architectural description of software-intensive systems," IEEE Std 1471, The Institute of Electrical and Electronics Engineers, Inc., 2000.

[Kruchten 1995] "The 4+1 View Model of Architecture," by Kruchten, P., IEEE Software, vol. 12, pp. 42 - 50, 1995.

Appendix A. 11 Types of Descriptions to Fully Document the Architecture (Bob McCaig)

Operational

Dynamic View (Use Cases)

OMI

Functional

Requirements

Static View (Landscapes)

Application to Application Messages

Data

1. Data Requirements Matrix

- *Physical*
  - IT design

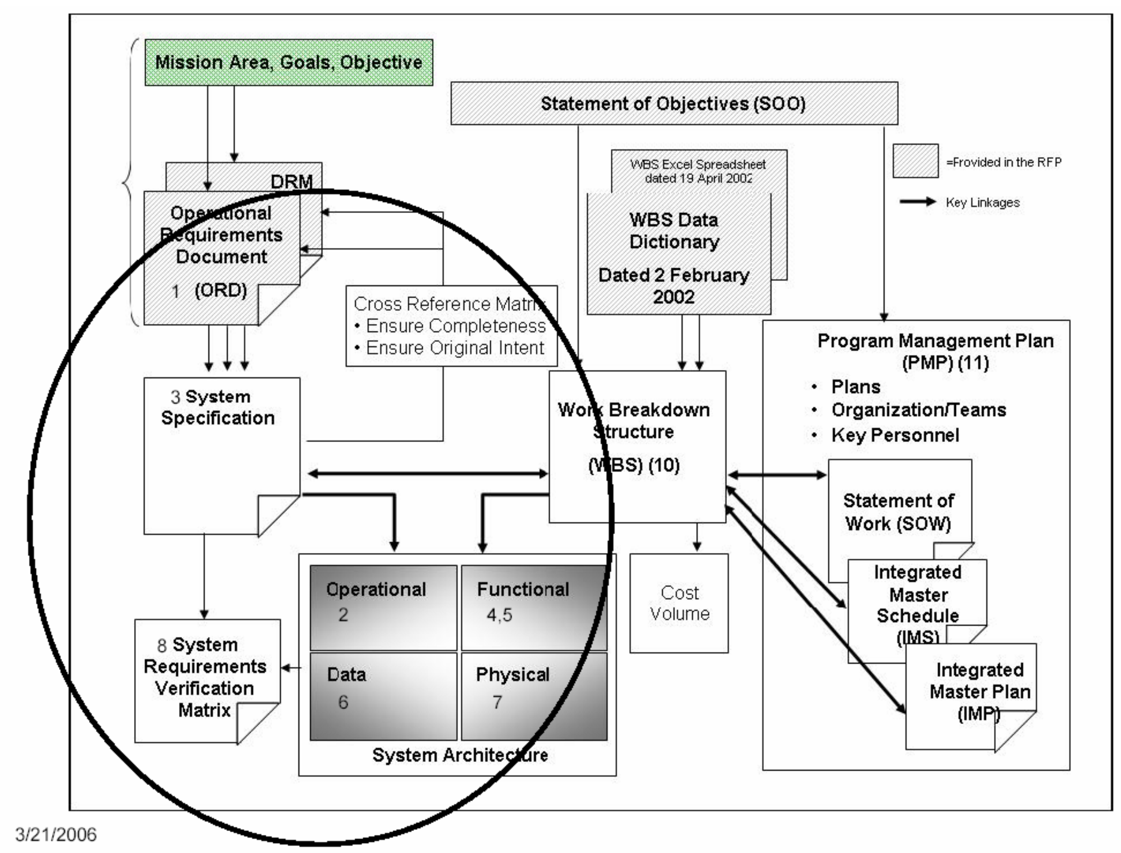
2. Processors, Storage, Links

  - Test documentation

3. Requirements Verification Matrix
4. Test Scenarios, Test Cases

- *Program*

5. Cost (WBS)
6. Schedule (Integrated Master Schedule)

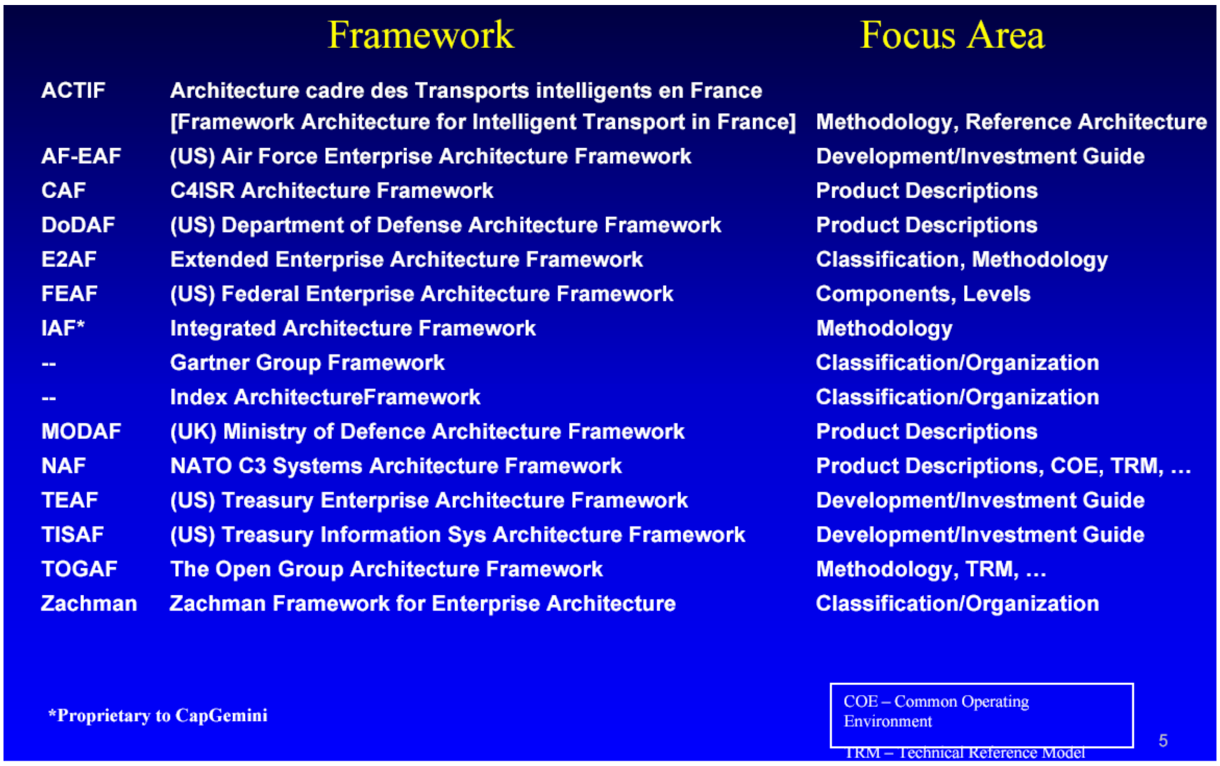

Mission Area, Goals, Objective
Statement of Objectives (SOO)
DRM
Operational Requirements Document 1 (ORD)
WBS Excel Spreadsheet dated 19 April 2002
WBS Data Dictionary Dated 2 February 2002
=Provided in the RFP
Key Linkages
Cross Reference Matrix
• Ensure Completeness
• Ensure Original Intent
3 System Specification
Work Breakdown Structure (WBS) (10)
Program Management Plan (PMP) (11)
• Plans
• Organization/Teams
• Key Personnel
Statement of Work (SOW)
Integrated Master Schedule (IMS)
Integrated Master Plan (IMP)
Cost Volume
Operational 2
Functional 4,5
Data 6
Physical 7
System Architecture
8 System Requirements Verification Matrix

3/21/2006

Appendix B. Some definitions (Rolf Siegers)

Architecture	"The fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution." (ANSI/IEEE 1471-2000)
Architecture Description	"A collection of products to document an architecture." (ANSI/IEEE 1471-2000)
Architecture Description Language (ADL)	A modeling notation to support architecture-based development. Usually describes components, structures, interfaces/interactions, data/control flow, etc.
Architecture Interchange Language (AIL)	A detailed format/schema to capture primitive architecture information to be exchanged between architecture modeling tools
Executable Architecture [Definition/ Description] Language (EADL)	A modeling notation for describing architectures of time-sensitive concurrent and distributed systems
Reference Architecture	A high-level system design free of implementation details; an architecture template
Reference Model	A set of architectural guidelines focused on a specific aspect of an architecture (e.g., technical, business, services, data, performance)
Technical Architecture	A perspective of the overall architecture reflecting the enterprise's data, applications and technical components.
View	"A representation of a whole system from the perspective of a related set of concerns." (ANSI/IEEE 1471-2000)
Viewpoint	"A specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis." (ANSI/IEEE 1471-2000)

Appendix C. A sampling of Frameworks



	Framework	Focus Area
ACTIF	Architecture cadre des Transports intelligents en France [Framework Architecture for Intelligent Transport in France]	Methodology, Reference Architecture
AF-EAF	(US) Air Force Enterprise Architecture Framework	Development/Investment Guide
CAF	C4ISR Architecture Framework	Product Descriptions
DoDAF	(US) Department of Defense Architecture Framework	Product Descriptions
E2AF	Extended Enterprise Architecture Framework	Classification, Methodology
FEAF	(US) Federal Enterprise Architecture Framework	Components, Levels
IAF*	Integrated Architecture Framework	Methodology
--	Gartner Group Framework	Classification/Organization
--	Index ArchitectureFramework	Classification/Organization
MODAF	(UK) Ministry of Defence Architecture Framework	Product Descriptions
NAF	NATO C3 Systems Architecture Framework	Product Descriptions, COE, TRM, …
TEAF	(US) Treasury Enterprise Architecture Framework	Development/Investment Guide
TISAF	(US) Treasury Information Sys Architecture Framework	Development/Investment Guide
TOGAF	The Open Group Architecture Framework	Methodology, TRM, …
Zachman	Zachman Framework for Enterprise Architecture	Classification/Organization

*Proprietary to CapGemini

COE – Common Operating Environment

TRM – Technical Reference Model

5

Appendix D. DoDAF

Applicable View	Framework Product	Framework Product Name	General Description
All Views	AV-1	Overview and Summary Information	Scope, purpose, intended users, environment depicted, analytical findings
All Views	AV-2	Integrated Dictionary	Data repository with definitions of all terms used in all products
Operational	OV-1	High-Level Operational Concept Graphic	High-level graphical/ textual description of operational concept
Operational	OV-2	Operational Node Connectivity Description	Operational nodes, operational activities performed at each node, connectivity and information exchange needlines between nodes
Operational	OV-3	Operational Information Exchange Matrix	Information exchanged between nodes and the relevant attributes of that exchange
Operational	OV-4	Organizational Relationships Chart	Organizational, role, or other relationships among organizations
Operational	OV-5	Operational Activity Model	Operational Activities, relationships among activities, inputs and outputs. Overlays can show cost, performing nodes, or other pertinent information
Operational	OV-6a	Operational Rules Model	One of the three products used to describe operational activity sequence and timing - identifies business rules that constrain operation
Operational	OV-6b	Operational State Transition Description	One of three products used to describe operational activity sequence and timing - identifies business process responses to events
Operational	OV-6c	Operational Event-Trace Description	One of three products used to describe operational activity sequence and timing - traces actions in a scenario or sequence of events and specifies timing of events
Operational	OV-7	Logical Data Model	Documentation of the data requirements and structural business process rules of the Operational View.
Systems	SV-1	Systems Interface Description	Identification of systems and system components and their interconnections, within and between nodes
Systems	SV-2	Systems Communications Description	Systems nodes and their related communications lay-downs
Systems	SV-3	Systems-Systems Matrix	Relationships among systems in a given architecture; can be designed to show relationships of interest, e.g., system-type interfaces, planned vs. existing interfaces, etc.
Systems	SV-4	Systems Functionality Description	Functions performed by systems and the information flow among system functions
Systems	SV-5	Operational Activity to Systems Function Traceability Matrix	Mapping of systems back to operational capabilities or of system functions back to operational activities
Systems	SV-6	Systems Data Exchange Matrix	Provides details of systems data being exchanged between systems
Systems	SV-7	Systems Performance Parameters Matrix	Performance characteristics of each system(s) hardware and software elements, for the appropriate timeframe(s)
Systems	SV-8	Systems Evolution Description	Planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation

Systems	SV-9	Systems Technology Forecast	Emerging technologies and software/hardware products that are expected to be available in a given set of timeframes, and that will affect future development of the architecture
Systems	SV-10a	Systems Rules Model	One of three products used to describe systems activity sequence and timing—Constraints that are imposed on systems functionality due to some aspect of systems design or implementation
Systems	SV-10b	Systems State Transition Description	One of three products used to describe systems activity sequence and timing—Responses of a system to events
Systems	SV-10c	Systems Event-Trace Description	One of three products used to describe systems activity sequence and timing -- System-specific refinements of critical sequences of events and the timing of these events
Systems	SV-11	Physical Schema	Physical implementation of the information of the Logical Data Model, e.g., message formats, file structures, physical schema

Technical	TV-1	Technical Standards Profile	Extraction of standards that apply to the given architecture
Technical	TV-2	Technical Standards Forecast	Description of emerging standards that are expected to apply to the given architecture, within an appropriate set of timeframes

Appendix E. Architectural Modeling and Tooling used at FEI

The inventory at FEI resulted in the following types of models and tools:

Block diagram

State transition diagram

Hierarchical tree structure

Relationships table

2D-space exploration

Simulators

Real-world inspectors

Budgets (tolerances, CPU, cost, …)

Layout, schematics