from The Rational Edge: The second in a two-part series, this article describes the Systems View (SV) and Technical Standards View (TV) products of the Department of Defense (DoD) Architecture Framework (DoDAF). Part 1 of the article presents an overview of the DoDAF and describes the Operational View (OV) products.

This multipart article discusses an approach to modeling complex systems architectures in a manner that complies with the Department of Defense (DoD) Architectural Framework (DoDAF). It explains how you can leverage modeling best practices in conjunction with the Unified Modeling Language (UML) and IBM Rational tools to create model views that not only support DoDAF compliance, but also add significant value in the design and development of complex systems without diverting effort from primary systems development objectives.

In Part 1 of this article I presented an overview of the DoDAF specification and explored its Operational View (OV) products. These are the products of greatest interest to government agencies and other operational decision-makers tasked with comparing the architectures of alternative systems, and with managing their development.

Here in Part 2, I'll explain the Systems View (SV) products. These are the model views most relevant to DoD contractors and others whose job it is to design and implement these complex systems architectures. For completeness relative to the DoDAF specification, I'll also briefly cover the Technical Standards View (TV) products in Part 2.
The Systems View products

The systems that comprise the operational architecture must collaborate to implement the mission capability specified in the Operational View, which I covered in Part 1 of this article. The purpose of the Systems View (SV) products is to provide multiple perspectives of the system under consideration. These views describe the system’s structure and express how it interacts with other elements of the enterprise architecture.

The various SV products follow from a white-box expansion of the subject system architecture, which identifies the logical and physical components of the system that must interact in order to achieve the desired behavior. These systems (the logical components) and system nodes (the physical components) are stereotyped classes, and are represented in a System Context Diagram. Relationships between these elements are indicative of operations/request messages that are specified when creating the SV-10c sequence diagrams (see below). Other SV products provide further information related to the physical and logical system interfaces, the system interactions, and the planned evolution of the system in the context of the operational enterprise.

Table 1 lists and describes the Systems View products and recommends a rational order in which to create them. Subsequent sections describe each of the SV products in more detail.

<table>
<thead>
<tr>
<th>Product</th>
<th>Title</th>
<th>Description</th>
<th>Representation</th>
<th>Creation Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV-1</td>
<td>System Interface Description</td>
<td>Identifies systems and system components and their interfaces, within and between nodes. Models reconciliation of both logical and physical perspectives through realization of common interfaces.</td>
<td>Class diagram with classes, localities, and interfaces</td>
<td>3</td>
</tr>
<tr>
<td>SV-2</td>
<td>Systems Communications Description</td>
<td>Models physical nodes and their related communications infrastructure.</td>
<td>Composite Structure Diagram Deployment Diagram</td>
<td>6</td>
</tr>
<tr>
<td>SV-3</td>
<td>Systems Matrix</td>
<td>Models relationships between systems and subsystems in the context of the overall architecture of the enterprise.</td>
<td>Model resident text matrix Exportable to XML</td>
<td>5</td>
</tr>
<tr>
<td>SV-4</td>
<td>Systems Functionality Description</td>
<td>Identifies system behavior and the information flow related to that behavior.</td>
<td>Activity Diagram for each system use case</td>
<td>8</td>
</tr>
<tr>
<td>SV-5</td>
<td>Operational Activity to System Function Traceability Matrix</td>
<td>Maps system internal behavior (realizations) to operational external activities (specification).</td>
<td>Model resident text matrix Exportable to XML</td>
<td>9</td>
</tr>
<tr>
<td>SV-6</td>
<td>System Information Exchange Matrix</td>
<td>Details information exchanges between system elements, including applications and hardware allocated to those elements.</td>
<td>Model resident text matrix Exportable to XML</td>
<td>10</td>
</tr>
</tbody>
</table>
**SV-7**

**System Performance Parameters Matrix**

Describes performance characteristics of system elements.

Model resident text matrix Exportable to XML Joint Realization Table(s) 11

**SV-8**

**System Evolution Description**

Describes planned evolution increments toward a specific future implementation.

Schedule or project plan with timelines 12

**SV-9**

**System Technology Forecast**

Describes emerging technologies that are likely to impact the current or specified future state of the system(s).

Text document 13

**SV10a**

**Systems Rules Model**

Describes constraints imposed on system functionality by business needs or operational mission requirements.

- Architectural constraints that may or may not be incorporated in the model (OCL/ SysML)
- Model references functional and non-functional requirements in text document 1

**SV-10b**

**Systems State Transition Description**

Describes system’s response to events.

State Transition Diagram(s) **

**SV-10c**

**Systems Event/Trace Description**

Describes internal systems behavior in terms of operational sequences and actions that realize operational scenarios or critical activities that reflect behavior identified in OV-6c.

Sequence diagrams for both logical and physical realizations of behavior 2 (logical) 4 (physical)

**SV-11**

**Physical Data Model**

Describes physical implementation of data storage and movement.

Class diagram indicating schema relationships to the logical data elements in OV-7 7

**State Transition Diagrams are optionally used to model critical, real-time responses to complex events requiring special treatment.**

**SV-1: System interface description**

The SV-1 creates the foundation for the subject system’s internal architecture. It depicts systems, system nodes, and the interfaces that exist within and between them. In this way, the SV-1 provides the linkage between the Operational View and the Systems View. This entails working with both the logical decomposition of the system and the allocation of logical functionality to physical components. The classifiers in this view represent objects in both logical and physical versions of sequence diagrams for each system use-case flow or scenario (derived from operations/messages to the subject system) identified in the Operational View.

We start with identifying candidate logical elements comprising the subject system. The initial discovery process may be intuitive and based on domain experience. The focus, at this point, is to start thinking about the components likely to comprise the logical subsystems. These may eventually turn out to be subsystems or even primitives, but that distinction is not yet important.
Later, as a result of use-case flowdown and joint realization activities, we identify remaining localities (as well as additional logical elements as we discover a need for them) to which we allocate logical functionality in order to realize specified behavior. From that information, we can allocate operations indicated on sequence diagrams to interfaces, each of which is realized by both logical (class) and physical (locality) elements. The SV-1 diagram contains the classes, localities, interfaces, and connections between those systems and system nodes.

**SV-2: Systems communications description**

The SV-2 is referred to as the Systems Communications Description. Intended to reflect the physical nodes (localities) and their communications infrastructure, the SV-2 is represented using a composite structure diagram, an artifact new to UML 2.0. A composite structure diagram is represented as a container of roles or objects that are explicitly connected at ports associated with roles (see Figure 1). Due to the potential volume and variety of information associated with communications connectivity, it may be desirable to associate these model elements with entries in a requirements repository, such as IBM Rational RequisitePro®, to take advantage of attribute values as supporting information.
SV-3: Systems matrix
The SV-3 is a matrix view of the system-to-system relationships that exist at any specified level of system decomposition. At a minimum, the matrix should identify which systems have relationships with other systems. You can also include additional content regarding characteristics of those relationships, as necessary. You derive the information content to create the SV-3 from the relationships established in the logical and physical realizations of behavior present in the SV-10c sequence diagrams.

SV-4: Systems functionality description
The SV-4 describes the functionality and required data flows necessary to support required system behavior. It takes the form of an activity diagram with partitions allocated to system elements responsible for activities. You add object flows to the activity flow in order to indicate data object inputs and outputs necessary to specified activities. The SV-4’s information content provides an alternate perspective from the information of the SV-10c sequence diagrams with their messages and parameters.

SV-5: Operational activity to system function traceability matrix
The SV-5 provides traceability between operational activities (e.g., use-case flow, scenarios) and the system functionality (operations) that realize the required behavior. We produce this information in the form of a hierarchical listing of the operational nodes, the operations they must support, and realizations of those operations. Ideally, you would extend these to encompass those systems/subsystems that collaborate to affect the realization, as well as to include the messages/operations sent to those system/subsystems.

SV-6: System information exchange matrix
The SV-6 is a matrix of data exchanges, similar to the OV-3 described in Part 1 of this article, which represents the behavior-based interactions between component systems and subsystems of the subject system. You can generate the SV-6 automatically using the IBM Rational Eclipse-based modeling tool by sourcing the SV-10c content. Each matrix row represents a data exchange, which is composed of characteristics of the data transferred between roles/objects in an interaction from the SV-10c sequence diagrams. The matrix identifies a distinct data exchange for each pair of objects or roles that interact and exchange information. Specific data exchange characteristics are typically associated with non-functional requirements or design constraints. The content of each Information Exchange Requirement (IER) is representative of an instantiation of a data object, where the attributes represent the data characteristics required by the DoDAF.

The SV-6 emphasizes the logical and operational characteristics of the information exchanged. This product is not intended to exhaustively capture all the details of the information exchanged within the architecture, but rather to help us understand the most important aspects of significant exchanges. Tables 2 and 3 show an example of the relevant information content, taken from the DoDAF specification. This content would typically trace to supplemental or non-functional requirements.
Table 2: SV-6 data description, etc., from the DoDAF specification

<table>
<thead>
<tr>
<th>Interface Identifier</th>
<th>Data Exchange Identifier</th>
<th>Data Description</th>
<th>Producer</th>
<th>Consumer</th>
<th>Nature of Transaction</th>
</tr>
</thead>
</table>
| System Interface Name and Identifier | System Data Exchange Name and Identifier | • Data Element Name and Identifier  
• Content  
• Format Type  
• Media Type  
• Accuracy  
• Units of Measurement  
• Data Standard | • Sending System Name and Identifier  
• Sending System Function Name and Identifier | • Receiving System Name and Identifier  
• Receiving System Function Name and Identifier | • Transaction Type  
• Triggering Event  
• Interoperability Level Achieved  
• Criticality |

Table 3: SV-6 performance attributes, etc., from the DoDAF specification

<table>
<thead>
<tr>
<th>Interface Identifier</th>
<th>Data Exchange Identifier</th>
<th>Performance Attributes</th>
<th>Information Assurance</th>
<th>Security</th>
</tr>
</thead>
</table>
| System Interface Name and Identifier | System Data Exchange Name and Identifier | • Periodicity  
• Timeliness  
• Throughput  
• Size | • Access Control  
• Availability  
• Confidentiality  
• Dissemination Control  
• Integrity  
• Non-Repudiation Consumer | • Protection (Type Name, Duration, Date)  
• Classification  
• Classification Caveat  
• Releasability  
• Security Standard |

SV-7: System performance parameters matrix

The SV-7 describes characteristics considered critical to effectively attaining mission objectives assigned to the subject system. This information can best be presented as a form, table, or matrix. The application domain determines the specific content of this view. A notional example is available for reference in the DoDAF specification. A Joint Realization Form specifically designed for this purpose, called a System Operation Specification, is also available through IBM Rational Software Services. When completed, you should store the SV-7 in the Documents folder associated with the model or as a traceable requirements document within IBM Rational RequisitePro.

Figure 2 illustrates a sample System Operation Specification form.
**Figure 2: System Operation Specification form (SV-7)**

**SV-8: System evolution description**

The SV-8 is a plan/schedule for the system's evolution, in the context of the evolving enterprise. The SV-8 is typically captured in a scheduling tool, such as Microsoft Project. Key milestones are associated with incremental implementations of changes to the structure and/or behavior of the system. We recommend storing the file associated with the schedule in the Documents folder associated with the Eclipse-based model.

**SV-9: System technology forecast**

The SV-9 identifies emerging technology that is likely to impact the structure or behavior of the system in its enterprise context. Ideally, you would correlate incremental changes in technology with the milestones in the SV-8 to facilitate overall decision-making and enterprise management.

**SV-10a: Systems rules model**

The SV-10a captures constraints restricting behavior of the systems/subsystems involved in satisfying operational objectives. Information is captured in text and produced in document
form. You would typically capture this information using a template tailored to the organizational audience.

It can be challenging to make the distinction between business rules/constraints and requirements. In this regard, we should keep in mind that decision points in the activity diagrams should reflect the instantiation of those rules. Some of this content may lend itself to being expressed using SysML or Object Constraint Language (OCL), and used to validate architectural artifacts within the modeling tool. However, the primary product for this view is a document. The SV-10a is analogous to the OV-6a (described in Part 1 of this article), but reflects a lower level of systems decomposition. As with the OV-6a, I recommend you use a document and an associated requirements management tool like IBM Rational RequisitePro.

**SV-10b: Systems state transition description**

When the behavior of one or more key architectural elements is event-driven, modeling with State diagrams can be especially useful in understanding that behavior. Where this approach is warranted, the SV-10b is produced.

**SV-10c: Systems event/trace description**

The SV-10c describes internal behavior of the subject system for each operation identified in the OV6c. We use sequence diagrams that focus on systems/subsystems and system nodes that interact using messages. These messages represent requests made to a system/subsystem/system node by associated systems, subsystems, or system nodes. The operation specification exists at the level of the Operational View and is realized in the Systems View. You create the structure for the realization by selecting the class owning the operation, clicking the right mouse button, and then selecting *DoDAF > Create Operation Realizations*. Any information exchanged as part of those requests (e.g., parameters) is represented by instances of an IO Entity class. Each message interaction also represents a data exchange and is used to populate the SV-6 matrix. You create this content by selecting *DoDAF > Create SV-6*. The matrix is displayed in the SV-6 tab.

**SV-11: Physical data model**

The SV-11 is the complement to the OV-7 (described in Part 1). We use a class diagram to represent database schema relationships necessary to host the informational content represented by the OV-7 Logical Data Model and the data objects of the SV-4.

**The Technical Standards View products**

The Technical Standards View provides the guidance that directs or constrains the implementation of the systems described in the Systems View. The TV reflects standards and limiting factors upon which design decisions are made, in the context of incrementally developing the system(s) to meet the mission objectives specified in the Operational View.

The TV addresses standards applicable to the current architecture (TV-1) and the evolution of that architecture (TV-2), as described in Table 4.
Table 4: Technical Standards View products and descriptions

<table>
<thead>
<tr>
<th>Product</th>
<th>Title</th>
<th>Description</th>
<th>Representation</th>
<th>Creation Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV-1</td>
<td>Technical Architecture Profile</td>
<td>Extraction of standards that apply to the specified architecture</td>
<td>Model references standards and constraints in text document. Consider use of IBM Rational RequisitePro or equivalent requirements tool.</td>
<td>1</td>
</tr>
<tr>
<td>TV-2</td>
<td>Standards Technology Forecast</td>
<td>Description of emerging standards that are expected to apply to the architecture, in specified timeframes</td>
<td>Model references standards and constraints (with time/milestone criteria) in text document. Consider use of IBM Rational RequisitePro or equivalent requirements tool.</td>
<td>2</td>
</tr>
</tbody>
</table>

**TV-1: Technical architecture profile**

The TV-1 describes existing standards and operational constraints that will likely affect the operational enterprise. The DoDAF specification provides a sample template suggesting that this information would be best captured using a text-based document. I recommend that you further incorporate relationships between specific standards and the architectural elements they affect, using a requirements management tools like IBM Rational RequisitePro. You can store specific characteristics of a standard as attributes of that standard so that establishing traceability becomes a relatively simple process.

**TV-2: Standards technology forecast**

The TV-2 describes potential and emerging standards and operational constraints that may affect the operational enterprise and its architecture as it, and its component systems, evolve. There are two categories of information captured in this product:

- Expected changes to standards or constraints referenced in the TV-1
- Changes to standards or new standards associated with the evolution of the enterprise to accommodate new systems and capabilities

The approach to capturing this information is the same as with the TV-1, except that traceability is also necessary to the SV-8 and SV-9 for entries that fall into the latter category above.

**Conclusion**

In this second part of my article, I've described the DoDAF Systems View (SV) and Technical Standards View (TV) products that extend and complement the information captured in the Operational View (OV) products described in Part 1. I've further shown how systems engineering teams can leverage the content of DoDAF products as we incrementally elaborate enterprise architecture from abstract capabilities to concrete logical and physical representations.

A robust, scalable process coupled with appropriate automation serves to drive development of consistent architectural content in a centralized model repository. Such a repository provides
enablement essential to the larger development organization and to key decision-makers in the operational enterprise. IBM Rational supports DoDAF compliance by integrating a proven process for systems engineering with a powerful, integrated tool suite to automate the creation of DoDAF-compliant products in the context of a well-formed systems architecture model.

Notes