magicdraw™

Architecture Made Simple

DoDAF PLUGIN

user guide

version 16.9

No Magic, Inc.

2010

CONTENTS

CONTENTS

1 INTRODUCTION

Abstract

This document gives descriptions for all DoDAF artifacts, their forms (diagram, report, matrix, and table), mapping to MagicDraw UML model, implementation details and useful tips. This information is described for every type of DoDAF product.

Overview

Section 2 describes the DoDAF implementation concept in MagicDraw.

Section 3 describes how “All Views” products are implemented in MagicDraw.

Section 4 describes how “Operational View” products are implemented in MagicDraw.

Section 5 describes how “Systems View” products are implemented in MagicDraw.

Section 6 describes how “Technical Standards View” products are implemented in MagicDraw.

Section 7 describes cross-cutting model elements.

2 DODAF CONCEPT IN MAGICDRAW

What is DoDAF

"The Department of Defense (DoD) Architecture Framework (DoDAF), Version 1.0, defines a common approach for DoD architecture description development, presentation, and integration. The Framework enables architecture descriptions to be compared and related across organizational boundaries, including Joint and multinational boundaries.

The Framework supports the development of interoperating and interacting architectures as referenced in DoD issuances." [*DoDAF, v1.0, Deskbook*] It defines three related views of architecture: Operational View (OV), Systems View (SV), and Technical Standards View (TV) as depicted in Figure 1.

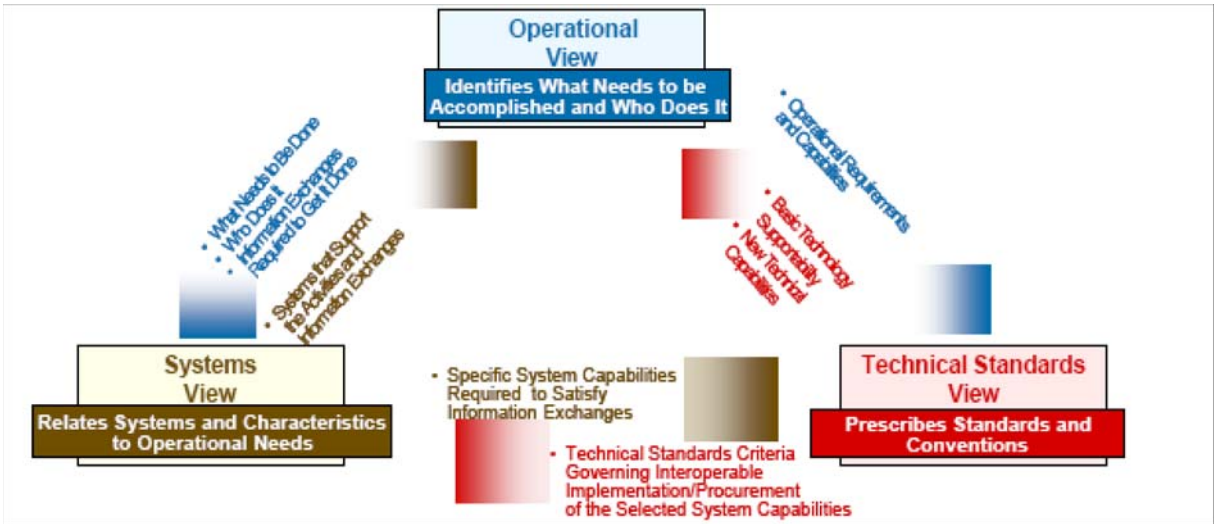

Figure 1 -- Linkages Among Views [DoDAF, v1.0, Deskbook]

DoDAF Architecture Views

"The term *integrated architecture* refers to an architecture description that has an Operational View (OV), Systems View (SV), and Technical Standards View (TV), and the views are integrated (i.e., there are common points of reference linking the OV and SV and also linking the SV and TV). The Operational Activity to Systems Functions Traceability Matrix (SV-5), for example, relates operational activities from the Operational Activity Model (OV-5) to system functions from the Systems Functionality Description (SV-4); the SV-4 system functions are related to systems in the Systems Interface Description (SV-1); thus bridging the OV and SV. An architecture is defined to be an *integrated architecture* when products and their constituent architecture data elements are developed such that architecture data elements defined in one view are the same (i.e., same names, definitions, and values) as architecture data elements referenced in another view." [*DoDAF, v.10, Vol. II*]

Definition of the Operational View

"The OV is a description of the tasks and activities, operational elements, and information exchanges required to accomplish DoD missions. DoD missions include both warfighting missions and business processes. The

OV contains graphical and textual products that comprise an identification of the operational nodes and elements, assigned tasks and activities, and information flows required between nodes. It defines the types of information exchanged, the frequency of exchange, which tasks and activities are supported by the information exchanges, and the nature of information exchanges." [*DoDAF, v.10, Vol. II*]

Definition of the Systems View

"The SV is a set of graphical and textual products that describes systems and interconnections providing for, or supporting, DoD functions. DoD functions include both warfighting and business functions. The SV associates system resources to the OV. These system resources support the operational activities and facilitate the exchange of information among operational nodes." [*DoDAF, v.10, Vol. II*]

Definition of the Technical Standards View

"The TV is the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements. Its purpose is to ensure that a system satisfies a specified set of operational requirements. The TV provides the technical systems implementation guidelines upon which engineering specifications are based, common building blocks are established, and product lines are developed. The TV includes a collection of the technical standards, implementation conventions, standards options, rules, and criteria organized into profile(s) that govern systems and system elements for a given architecture. " [*DoDAF, v.10, Vol. II*]

Definition of the All-Views

"There are some overarching aspects of an architecture that relate to all three views. These overarching aspects are captured in the All-Views (AV) products. The AV products provide information pertinent to the entire architecture but do not represent a distinct view of the architecture. AV products set the scope and context of the architecture. The scope includes the subject area and time frame for the architecture. The setting in which the architecture exists comprises the interrelated conditions that compose the context for the architecture. These conditions include doctrine; tactics, techniques, and procedures; relevant goals and vision statements; concepts of operations (CONOPS); scenarios; and environmental conditions." [*DoDAF, v.10, Vol. II*]

DoDAF Architecture Products

MagicDraw UML offers full support of all DoDAF products. It is easy to create integrated DoDAF products and maintain their inter-relationships. MagicDraw will create **integrated architecture** [*DoDAF, v1.0, Desktop, page 2-1*], supporting all types of relationships between elements in separate products or views. Modification of the model will result in the automatic updates of the referenced elements.

There are 4 major DoDAF product types distinguished in MagicDraw:

1. UML based **diagrams**. User is provided with a set of UML and SysML diagrams extended to meet DoDAF products. Diagrams will be equipped with DoDAF specific diagram toolbar for the usability sake.
   Any regular UML, SysML, or BPMN diagram can also be added to the DoDAF project as any DoDAF product.
2. Dependency **matrixes**[1]**.** Read-only tables with model elements as columns and rows showing if there is a relationship between them. A dependency matrix is a view showing existing model information.

1. Matrices are called diagrams in MagicDraw for simplicity sake.

3. Tables[2]. A writable table made specifically to meet some DoDAF products, TV-1 for example. Modifying the table will change the related model element, so model integrity will be maintained.
4. Generated **documents** (**reports**). MagicDraw will generate PDF or RTF documents from the DoDAF model. Go to Tool->Report and pick the report templates for the DoDAF products.

MagicDraw will have possibility to link any external document to the DoDAF model.

UML Profile for DoDAF

In order to meet specific DoDAF needs, a UML profile extending UML and SysML was created. The UML profile is explained in detail for every DoDAF product.

Notes

- Only DoDAF specific properties will be listed in the detailed description.
- All standard UML properties will accessible to the modeler, e.g. Name, Owner, etc.

Summary of DoDAF Products

The products of each of the views and their form in MagicDraw are listed below.

DoDAF Product	Name	Diagram (diagram type)	Matrix	Report	Table
AV-1	Overview and Summary Information	-	-	√	-
AV-2	Integrated Dictionary	-	-	√	
OV-1	High-Level Operational Concept Graphic	UML Class	-	-	-
OV-2	Operational Node Connectivity Description	UML Class UML Composite structure	-	-	-
OV-3	Operational Information Exchange Matrix	-	-	√	-
OV-4	Organizational Relationships Chart	UML Class	-	-	-
OV-5	Operational Activity Model	UML Class UML Activity	-	-	-
OV-6a	Operational Rules Model	-	-	√	-
OV-6b	Operational State Transition Description	UML State Machine	-	-	-
OV-6c	Operational Event-Trace Description	UML Sequence	-	-	-
OV-7	Logical Data Model	UML Class	-	-	-
SV-1	Systems Interface Description	UML Class UML Composite structure	-	-	-

2. Tables are called diagrams in MagicDraw for simplicity sake..

SV-2	Systems Communications Description	UML Class UML Composite structure	-	-	-
SV-3	Systems-Systems Matrix	-	√	-	-
SV-4	Systems Functionality Description	UML Class UML Activity	-	-	-
SV-5	Operational Activity to Systems Function Traceability Matrix	-	√	-	-
SV-6	Systems Data Exchange Matrix	-	-	√	-
SV-7	Systems Performance Parameters Matrix	-	-	-	√
SV-8	Systems Evolution Description	UML State Machine	-	-	-
SV-9	Systems Technology Forecast	-	-	-	√
SV-10a	Systems Rules Model	-	-	√	-
SV-10b	Systems State Transition Description	UML State Machine	-	-	-
SV-10c	Systems Event-Trace Description	UML Sequence	-	-	-
SV-11	Physical Schema	UML Class	-	-	-
TV-1	Technical Standards Profile	-	-	-	√
TV-2	Technical Standards Forecast	-	-	-	√

Product Taxonomy

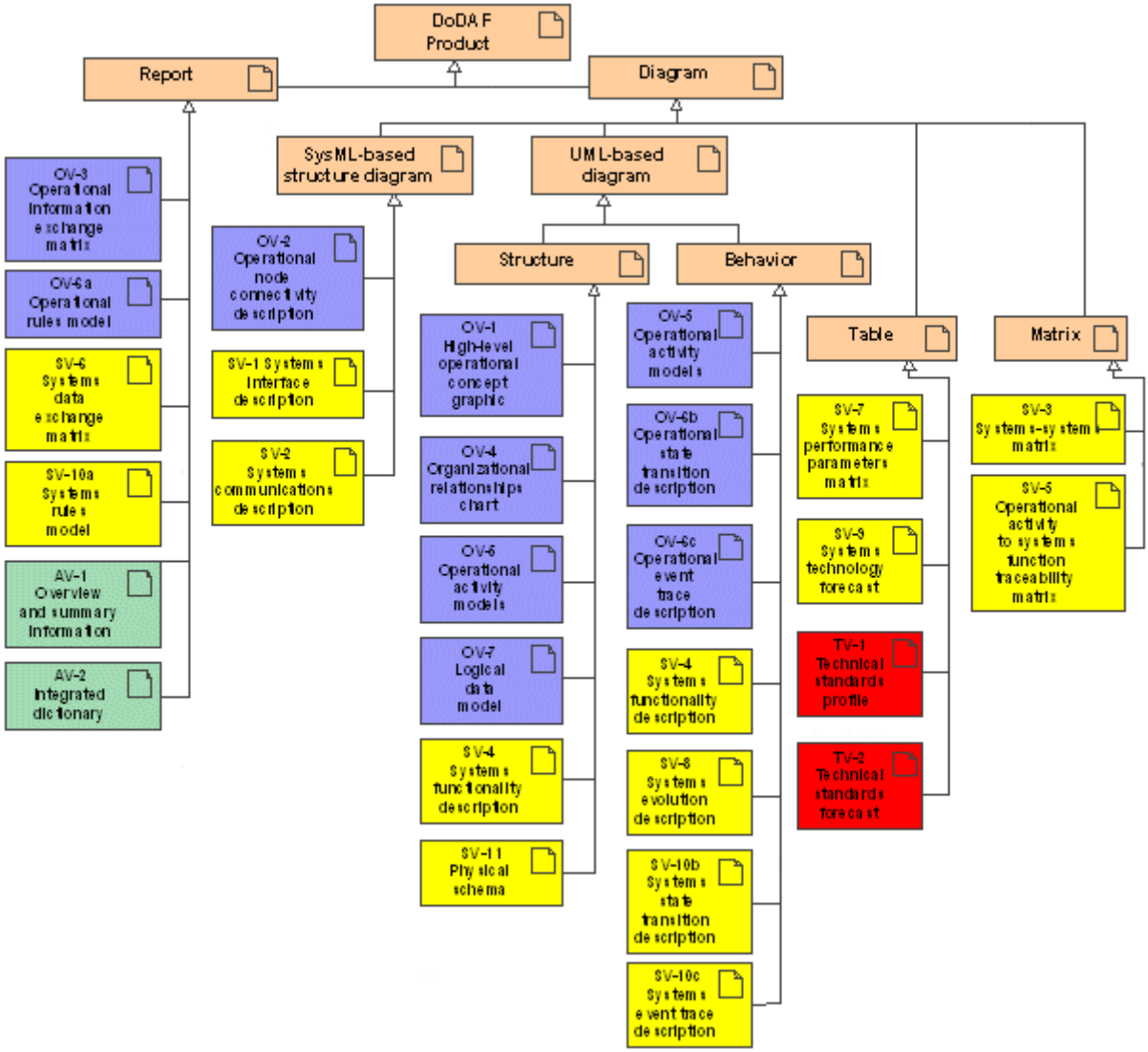

Figure 2 -- Product taxonomy

3 ALL VIEWS

AV-1. Overview and Summary Information

Description

"The Overview and Summary Information provides executive - level summary information in a consistent form that allows quick reference and comparison among architectures. AV-1 includes assumptions, constraints, and limitations that may affect high-level decision processes involving the architecture." [*DoDAF, v.10, Vol. II]*

Implementation in MagicDraw

AV-1 will be generated from the rest of the model. Naturally, one needs to define the AV-1 as start for the whole project. However we suggest sketching the AV-1 at the beginning and after the whole DoDAF project is described, generating a brand new from the model data. In such case, it you will be sure, that AV-1 has the latest model data.

Some of the AV-1 information does not exist in the model, so it will be necessary to add some details manually.

UML Profile for DoDAF

No UML extensions needed for AV-1.

Report template

Overview and Summary Information report provides summary information about architecture project. Thus generally information is generated for the whole project. Part of report may be generated using information about model elements collected from the model and rest part may be specified manually by specifying properties on Report dialog.

In general AV-1 provides summary information in the given below form:

Architecture Project Identification

- Name
- Architect
- Organization developing the architecture
- Assumptions and constraints
- Approval authority
- Date completed
- Level of effort and projected and actual costs to develop the architecture

Scope: Architecture View(s) and Products Identification

- Views and products developed
- Time frames addressed
- Organizations involved

Purpose and Viewpoint

- Purpose, analysis, questions to be answered by analysis of the architecture

- From whose viewpoint the architecture is developed

Context

- Mission
- Doctrine, goals, and vision
- Rules, criteria, and conventions followed
- Tasking for architecture project and linkages to other architectures

Tools and File Formats Used

Findings

- Analysis results
- Recommendations

User can customize report by selecting whether to include information from the model or not. Element documentation and empty sections may be also included optionally.

Relationship to other products

MagicDraw takes AV-1 data from the rest of the model.

Sample

1.Architecture Project Identification

1.1.Name
DoDAF sample

1.2.Architect
DoDAF sample team

1.3.Organization developing the architecture
No Magic, Inc.

1.4.Assumptions and constraints
This sample is intended for demonstration purposes only.
Most of diagrams are taken from DoDAF version 1.0 - Deskbook.

1.5.Approval authority
NA

1.6.Date completed
January 24, 2007

1.7.Level of effort and projected and actual costs to develop the architecture
NA

Figure 1 -- AV-1 sample

AV-2. Integrated Dictionary

Description

"The Integrated Dictionary contains definitions of terms used in the given architecture. It consists of textual definitions in the form of a glossary, a repository of architecture data, their taxonomies, and their metadata (i.e., data about architecture data), including metadata for tailored products, associated with the architecture products developed. Metadata are the architecture data types, possibly expressed in the form of a physical schema." [*DoDAF, v.10, Vol. II*]

Implementation in MagicDraw

One of main purposes of AV-2 is to let all elements be accessible and easy traceable. It should serve as table of contents for quick access to any element in the DoDAF model. In MagicDraw, there are two ways to get quick access to all DoDAF elements:

- MagicDraw model browser represents all DoDAF products and elements in tree structure grouped by product name. This is an integral part of MagicDraw application.
- MagicDraw generates AV-2 as a **report**. User may also narrow down the scope of the AV-2 if needed. This report should be created last, when the entire model is stable and will not be changed.

UML Profile for DoDAF

No UML extensions needed for AV-1.

Report template

Integrated Dictionary report provides the list of DoDAF model elements from selected scope. In the report model elements are grouped according to the first letter of DoDAF element names. In each group information about DoDAF element name, type, documentation and dependent elements is given.

User can customize report by selecting the type of DoDAF elements for which report should be generated. Element documentation and the list of dependent elements may be also included optionally.

Relationship to other products

MagicDraw takes AV-2 data from the rest of the model.

Sample

A
ADSI (System)
AFATDS (System)
APS (System)
ASAS (System)

C
CIGSS (System)
Conduct Combat Assesment (SystemFunction)
CTAPS (System)

D
DCCC (OperationalNode)
DCCC (SystemsNode)
DIA (OperationalNode)
DIA (SystemsNode)
DJFCC (DOCC, 513ACE) (OperationalNode)
DJFLCC (SystemsNode)

Figure 2 -- AV-2 sample

4 OPERATIONAL VIEW PRODUCTS

OV-1. High-Level Operational Concept Graphic

Description

"The High-Level Operational Concept Graphic describes a mission and highlights main operational nodes (see OV-2 definition) and interesting or unique aspects of operations. It provides a description of the interactions between the subject architecture and its environment, and between the architecture and external systems.

OV-1 is usually a free-form graphic document that describes a mission. However a UML representation may also be added, so you can trace how your mission is implemented by lower abstraction level in OV or SV models. We recommend adding as much as possible information to the UML model as this data can be later reused by other artifacts.

OV-1 consists of a graphical executive summary for a given architecture with accompanying text. The content of OV-1 depends on the scope and intent of the architecture, but in general it describes the business processes or missions, high-level operations, organizations, and geographical distribution of assets.

OV-1 is the most general of the architecture products and the most flexible in format. The product usually consists of one or more graphics (or possibly a movie), as needed, as well as explanatory text." [*DoDAF, v.10, Vol. II*]

Implementation in MagicDraw

OV-1 is a product highlighting a core concepts or the scope of the subject system. In general a sketch can be used for OV-2, however, in order to get consistent model, it is recommended to draw a **diagram**.

So in MagicDraw OV-1 is implemented as diagram, which is based on **UML Use Case diagram**.

If an external document or an image is done for OV-1, a hyperlink to a free-form document describing OV-1 can be added to simplify the navigation.

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Mission	<<Mission>> [UseCase]	
Objective	<<Objective>> [UseCase]	
Line	<<Line>> [Association]	N/A
Target Area	<<TargetArea>> [Class]	
Asset	<<Asset>> [Class]	

Mission

A Mission is UML Use Case showing the mission of the subject system. There might be multiple missions defined in the same model.

Extends UML Use Case.

Objective

Describes the objective of the subject system. There might be multiple objectives defined in the same model.

Extends UML Use Case.

Line

Line is a generic relationship showing an abstract connection between the elements. It may represent actual relationship, defined in the other products.

Extends UML Association.

Attributes:

representationType: LineRepresentationKind	Defines what relationship type does this Line represents. Possible values are taken from LineRepresentationKind enumeration: Needline Flow Interface Communications Link
actualImplementation: DoDAFRelationship [0..1]	Defines what concrete relationship does this line represents

TargetArea

Defines target area.

Extends UML Class.

Attributes:

position: String	Textual description describing the position of the target area

Asset

Asset is an abstract entity, describing an asset of subject system. It may represent actual element, defined in the other products.

Extends UML Class.

Attributes:

position: String	Textual description describing the position of the asset
representationType: AssetRepresentationKind	Defines what entity type does this Asset represents. Possible values are taken from AssetRepresentationKind enumeration: Operational Node Operational Activity Organizational Resource Systems Node System System Function
actualImplementation: DoDAFEntity [0..1]	Defines what concrete entity does this asset represents

Profile diagram

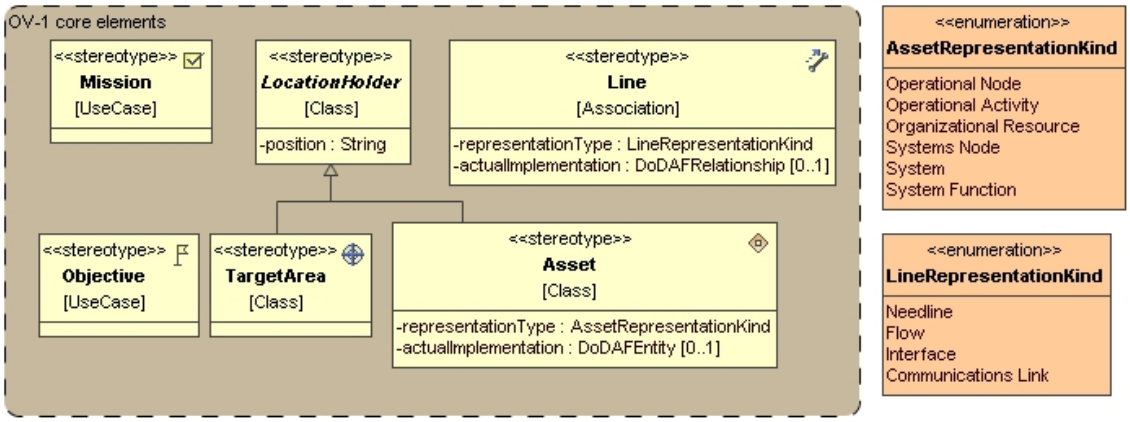

Figure 1 -- OV-1 profile diagram

Creating the product

OV-1 should be created as a high-level abstraction model, serving as a guideline for the rest of the DoDAF model. It should show only core objects and relationships. It is up to a modeler to choose the abstraction level. Modeler can choose from:

Show the missions, objectives, main assets (with types defined), and connecting lines (with types defined).

Show the missions, objectives, main assets (with actual entities defined), and connecting lines (with actual relationships defined). It is recommended to create Assets and Lines without showing their actual implementations (Nodes, Systems, Interfaces, etc). Later, when appropriate OV and SV products are done, user can link them the to the OV-1.

Show the missions, objectives, main entities, and main relationships. You can drag and drop existing elements to the OV-1.

Use the OV-1 specific diagram toolbar for fast creation of this product.

Relationship to other products

OV-1 can use various DoDAF elements as *actualImplementation* property for the Asset and various DoDAF paths as *actualImplementation* property for the Line.

If needed, concrete DoDAF element can be used in the OV-1, however, we recommend to use only core ones or none.

Sample

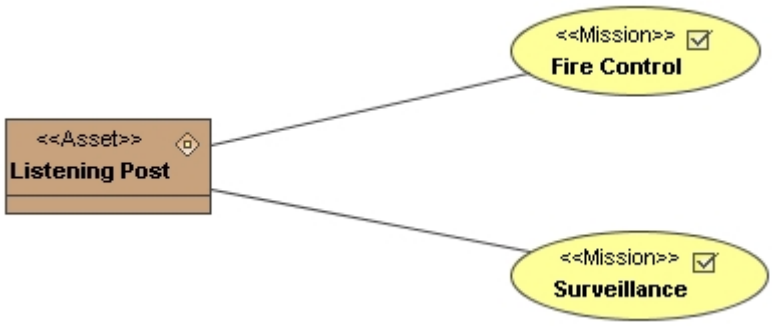

Figure 2 -- OV-1 sample 1

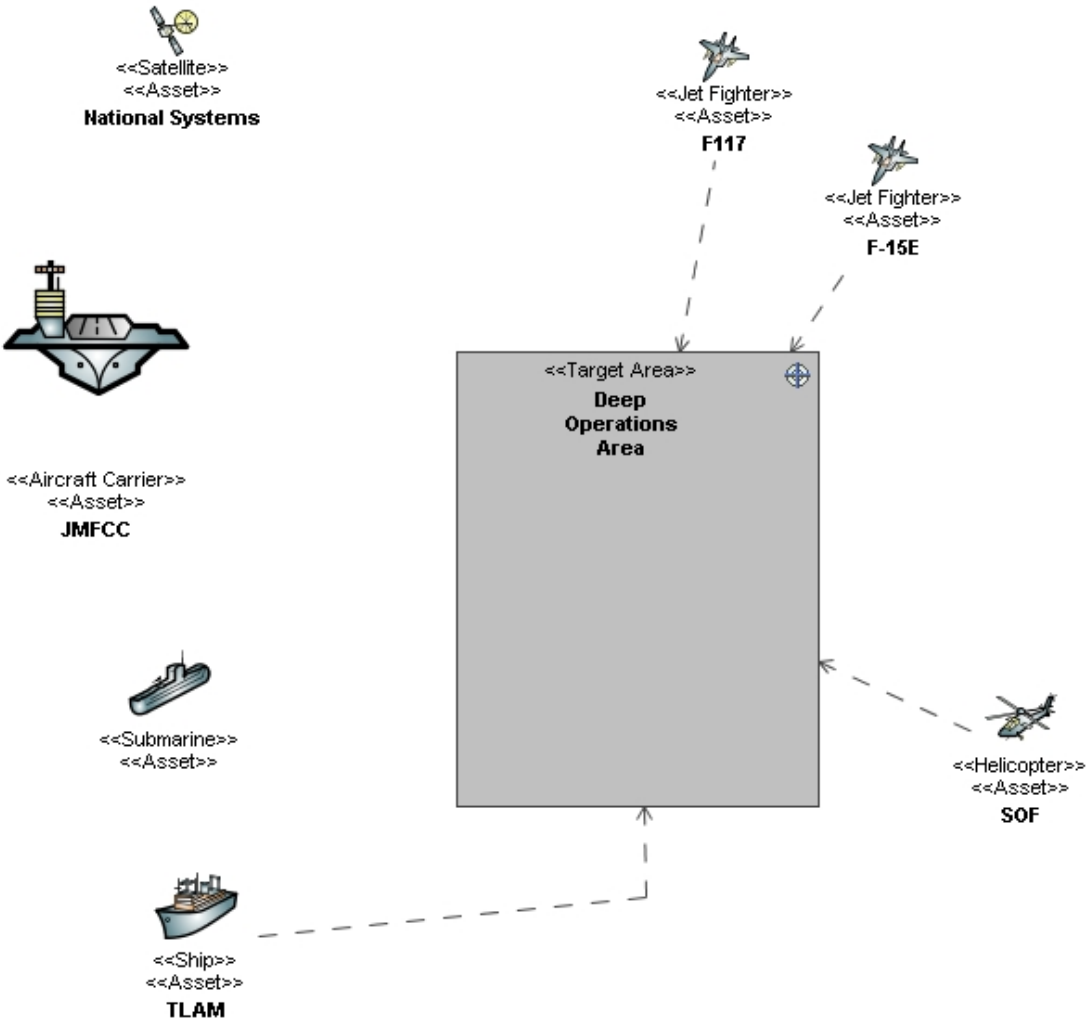

Figure 3 -- OV-1 sample 2

OV-2. Operational Node Connectivity Description

Description

“The Operational Node Connectivity Description graphically depicts the operational nodes (or organizations) with needlines between those nodes that indicate a need to exchange information. The graphic includes internal operational nodes (internal to the architecture) as well as external nodes.

OV-2 is intended to track the need to exchange information from specific operational nodes (that play a key role in the architecture) to others. OV-2 does not depict the connectivity between the nodes. “[*DoDAF, v.10, Vol. II*]

Implementation in MagicDraw

OV-2 is a **diagram**, which is based on either **UML Class Diagram** or **UML Composite Structure Diagram** depending on the level of details.

User also may use:

- Regular UML Class Diagram
- Regular UML Composite Structure Diagram
- SysML Block Definition Diagram
- SysML Internal Block Diagram
- Any other MagicDraw diagram

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Needline	<<Needline>> [Association, Connector]	N/A
Operational Node	<<OperationalNode>> [Class]	
Operational Node Usage	<<OperationalNodeUsage>> [Property]	
Organizational Resource Usage	<<OrganizationalResourceUsage>> [Property]	
Service Specification	<<ServiceSpecification>> [Class]	
Soa Service	<<SoaService>> [Port]	
Information Exchange	See OV-3 data element definitions	
Information Element	See OV-3 data element definitions	
Operational Activity	See OV-5 data element definitions	
Systems Node	See SV-1 data element definitions	
Interface	See SV-1 data element definitions	

Needline

A Needline is a relation between two Operational Nodes showing that these two nodes communicate to each other and exchange information.

Extends UML Association (in block definition diagrams), UML Connector (in internal block diagrams).

Attributes

identifier: String	Unique identifier of the Needline (may be a number)
implementingInterface: Interface [*]	Interface from SV-1 implementing this Needline

OperationalNode

An Operational Node is an element that produces, consumes, or manipulates information. It can also group Organizational Structure Elements from OV-4 and Operational Activities from OV-5.

Derives from SysML Block, Performer (**extends** UML Class).

Attributes

levelIdentifier: String	If using hierarchical decomposition of nodes: Identifier that corresponds to the node's place in the node hierarchy – should be unique
isExternal: Boolean = false	Shows if this is an external Operational Node. By default Operational Node is not external.
operationalRole: OperationalRole	Defines an operational role that is applied to Operational Node. Possible values are taken from OperationalRole enumeration: Operational Service Provider Operational Service Consumer Operational Unanticipated User
performers: Performer [*]	Organizational Structure elements grouped by this Operational Node **Note**: Performer may be any specific element of *Performer* class: Organization, Organization Type, Role, Person
activities: OperationalActivity [*]	Operational Activities grouped by this Operational Node
implementedBy: SystemsNode [*]	Systems Nodes from SV-1/SV-2 that support this Operational Node by automation resident at Systems Node
performs: OperationalActivity [*]	OperationalActivities performed by this Performer

OperationalNodeUsage

Operational Node Usage is a helper element that is needed to show an Operational Node inside other Operational Node. We are using SysML based approach here: parent Operational Node will have a Operational Node Usage as a SysML BlockProperty. The type of the Operational Node Usage will be child Operational Node.

Derives from SysML BlockProperty (**extends** Property).

Note. Semantically this is the same as Part in UML composite structure diagrams.

Attributes

operationalRole: OperationalRole	Defines an operational role that is applied to Operational Node Usage. Possible values are taken from OperationalRole enumeration: Operational Service Provider Operational Service Consumer Operational Unanticipated User

OrganizationalResourceUsage

Organizational Resource Usage is a helper element that is needed to show an Organizational Resource inside Operational Node. We are using SysML based approach here: parent Operational Node will have an Organizational Resource Usage as a SysML BlockProperty.

Derives from SysML BlockProperty (**extends** Property).

Note. Semantically this is the same as Part in UML composite structure diagrams.

ServiceSpecification

Service Specification is used to define Interfaces provided through the Soa Service and/or used by the Soa Service. Parts of Service Specification are typed by these Interfaces and represent potential users and providers of the Soa Service. Service Specification may have associated Behavior for denoting the order of invocation of operations on it.

Extends UML Class.

Attributes

serviceDescription: string[1]	The description of the Soa Service or a link to a formal Soa Service description

SoaService

The service model element provides the end-point for service interaction. Soa Service can be owned by a Service Provider or a Service Consumer. The type of Soa Service should be Interface or Service Specification. Interface should be chosen as a type of Soa Service if there is no protocol for using the Soa Service. Service Specification should be chosen as a type for Soa Service if there is protocol for using the Soa Service.

Extends UML Port.

Profile diagram

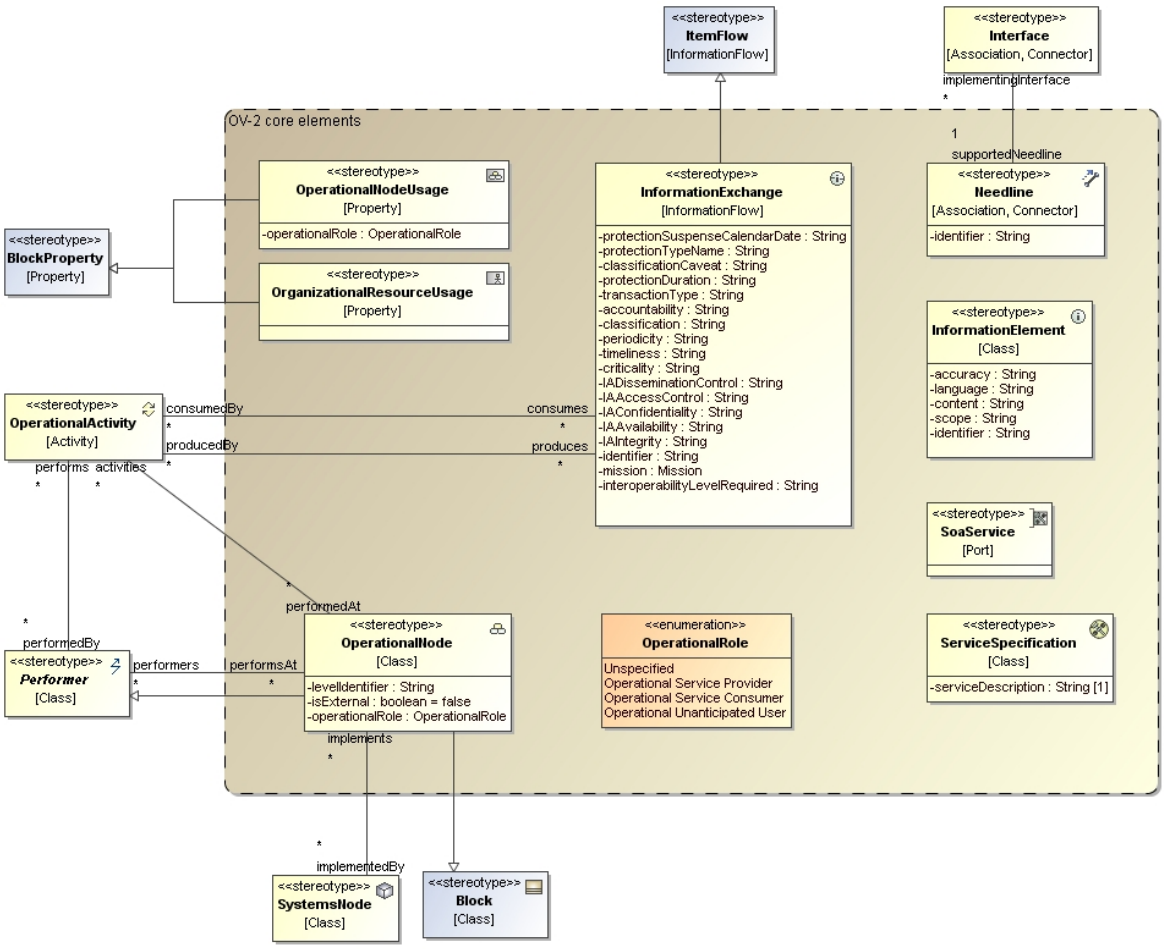

Figure 4 -- OV-2 profile diagram

Creating the product

It is recommended at first to create the **OV-2 Operational Node Connectivity Descriptions** with main Operational Nodes. You will create main Operational Nodes, Needlines, Information Exchanges, etc.

If you have composite structure for the Operational Nodes, you need to create an **OV-2 Operational Node Internal Connectivity Descriptions** for the Operational Nodes that have sub nodes. Sub nodes will be created as Operational Node Usages in the parent Operational Nodes. Child Operational Node will be set as type for the Operational Node Usages.

Node's internal diagram can be easily created from an Operational Node (context menu -> New Diagram -> OV-2 Operational Node Internal Connectivity Description).

There are several ways to create the Information Exchanges:

1. Drag the Information Element onto the Needline
2. Right click on the Needline->Information Exchanges->New
3. Create new Information Exchange from the Needline's specification window, "Information Exchanges" node.

Information Elements (according to [*DoDAF, v.10, Vol. II*]) should be defined in the OV-3. However, because of implementation specifics, in MagicDraw Information Elements have to be defined in OV-2.

It is not necessary to define performed Operational Activities, however you can show them using the Note.

<<< PLEASE add instructions on adding "operational node usage" and "organizational node usage" >>>

<<< Please show examples of how usage looks. >>>

<<<Please add an example of Operational Activities in a Note >>>

Modeling service architectures

OV-2 product may be used for modeling service architectures. Operational Node and sub nodes (Operational Node Usages) have “Operational Role” property that allows you to specify the responsibility Operational Node plays: Operational Service Consumer, Operational Service Provider or Operational Unanticipated User.

Operational Service Consumer and Operational Service Provider can communicate only via Soa Service. Since Soa Service extends UML Port Operational Service Provider and Operational Service Consumer should be realizing classifiers of the same component. To model this on **OV-2 Operational Node Connectivity Description** diagram create Component and drag on it Operational Nodes. Operational Nodes will become realizing classifiers of the same Component and you will be able to connect their Soa Services using Needline [connector] relationship.

The type of Soa Service can be UML Interface or Service Specification. Use Service Specification when there is behavior for using Soa Service. After behavior addition you may detail it using any type of behavior diagram. After this double-click on Service Specification will open not Service Specification specification dialog but associated behavior diagram.

Relationship to other products

Organizational structure items from **OV-4** may be set as *performers* for Operational Node, showing that they are performing at this node.

Operational Activities from **OV-5** may be set as *activities* for Operational Node, showing that they are performed at this node.

Interface from **SV-1** may be set as *implementingInterface* for Needline, showing that this Needline is implemented by this interface.

Systems Node from **SV-1** may be set as *ImplementedBy* for Operational Node, showing that this Operational Node is implemented by Systems Node.

Information Exchange may have relations to Operational Activities from **OV-5** through *consumedBy* and *producedBy* properties, showing that it is consumed or produced by Operational Activities.

System Data Exchange from **SV** products is tied to Information Exchange using the *implementedBy* property.

Information Element caried by Information Exchange is assigned view *informationElement* property.

Sample

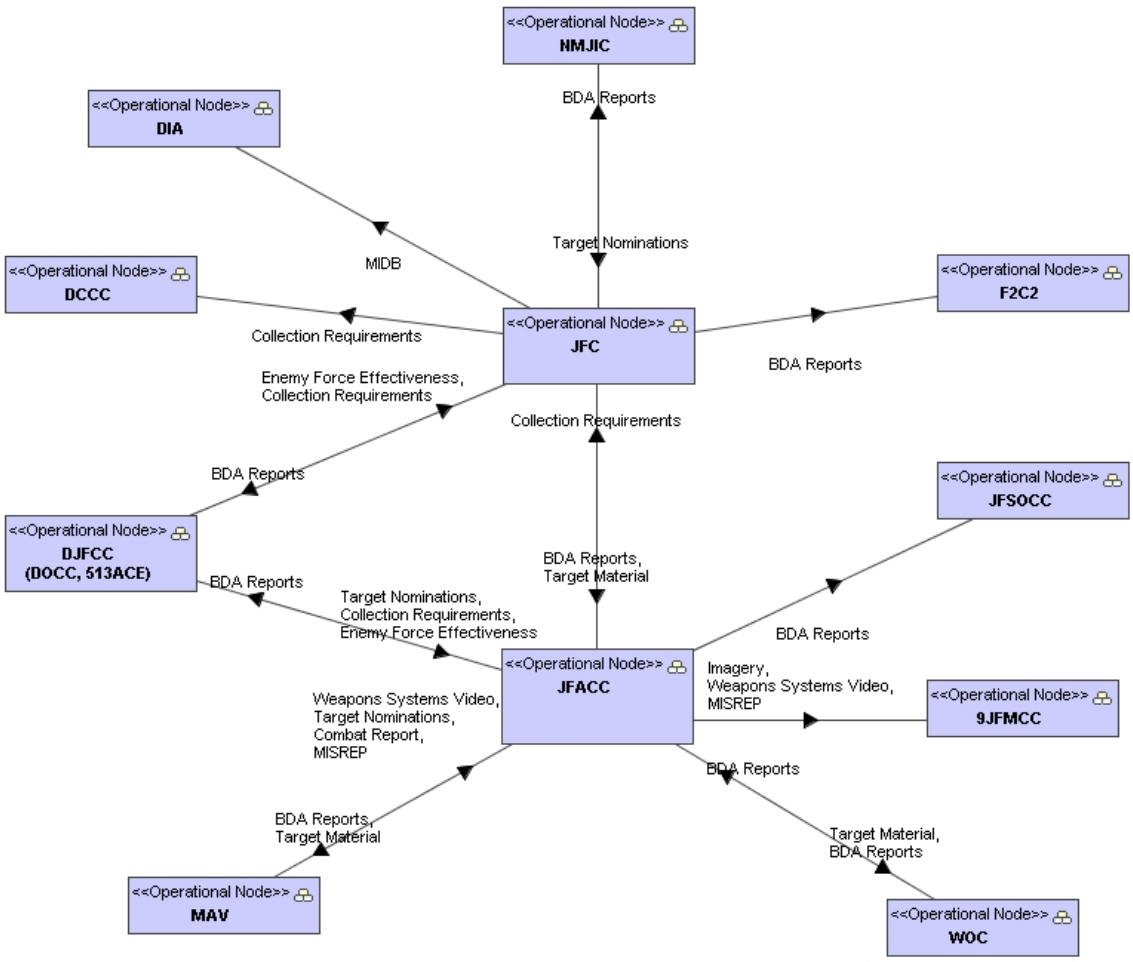

Figure 5 -- OV-2 sample

OV-3. Operational Information Exchange Matrix

Description

"The Operational Information Exchange Matrix details information exchanges and identifies "who exchanges what information, with whom, why the information is necessary, and how the information exchange must occur" [CJCSI 6212.01B, 2000]. There is not a one-to-one mapping of OV-3 information exchanges to OV-2 Needlines; rather, many individual information exchanges may be associated with one Needline. [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

OV-3 is a **document**, which is implemented as MagicDraw **report**, using model information mainly from OV-2. After OV-2 is done, user can generate a document for OV-3.

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Information Exchange	<<InformationExchanget>> [InformationFlow]	
Information Element	<<InformationElement>> [Class]	
System Data Element	See SV-6 data element definition	

InformationExchange

Information Exchange shows information that is exchanged by two Operational Nodes via the Needline. An Information Element will show the kind of information that is exchanged. According to [*DoDAF, v.10, Vol. II*], Information Exchange should be modeled in the OV-3. However in MagicDraw it is modeled in OV-2, while OV-3 is just **document** generated from OV-2.

Extends UML InformationFlow.

Attributes

protectionSuspenseCalendarDate: String	The calendar date on which the designated level of safeguarding discontinues
protectionTypeName: String	Name for the type of protection
classificationCaveat: String	A set of restrictions on information of a specific classification; supplements a security classification with information on access, dissemination, and other types of restrictions
protectionDuration: String	How long the information must be safeguarded
transactionType: String	Descriptive field that identifies the type of exchange
accountability: String	Security principle that ensures that responsibility for actions/events can be given to an organization willingly or by obligation
classification: String	Classification code for the information
periodicity: String	How often the Information Exchange occurs; may be an average or a worst case estimate and may include conditions (e.g., wartime or peacetime)
timeliness: String	Required maximum allowable time of exchange from node to node (in seconds)
criticality: String	The criticality assessment of the information being exchanged in relationship to the mission being performed
IADisseminationControl: String	The kind of restrictions on receivers of the information based on sensitivity of information
IAAccessControl: String	The class of mechanisms used to ensure only those authorized who can access information
IAConfidentiality: String	The kind of protection required for information to prevent unintended disclosure
IAAvailability: String	The relative level of effort required to be expended to ensure that the information can be accessed
IAIntegrity: String	The kind of requirements for checks that the content of the information has not been altered
identifier: String	Identifier for the Information Exchange – usually based on the relevant Needline identifier
mission: Mission	Joint Mission Area, cross-mission area domain, Universal Joint Task List activity, related specific scenario, or other task-related basis of the architecture
interoperabilityLevelRequired: String	Level of Information Systems Interoperability (LISI), or other interoperability measure, required
informationElement: InformationElement [1]	Information being carried by this Information Exchange
implementedBy: SystemDataExchange [*]	A list of System Data Exchanges that implement this Information Exchange

consumedBy: OperationalActivity [*]	A list of Operational Activities that consume information
producedBy: OperationalActivity [*]	A list of Operational Activities that produce information

InformationElement

Information Element describes information that is used in the DoDAF model.

Extends UML Class.

Attributes

identifier: String	Identifier for the Information Element – usually based on the relevant Information Exchange identifier; should be unique for the architecture
accuracy: String	Description of the degree to which the information conforms to actual fact as required by Operational Node
language: String	Identifier/name of codes of the natural language(s) involved in the information exchange; relevant for multinational operations
content: String	The content of the Information Element (i.e., actual information to be exchanged)
scope: String	Text description of the extent or range of the Information Element content
implementedBy: SystemDataElement [*]	A System Data Element that implements this Information Element
informationExchange: InformationExchange [*]	Information Exchanges via which this Information Element is exchanged

Profile diagram

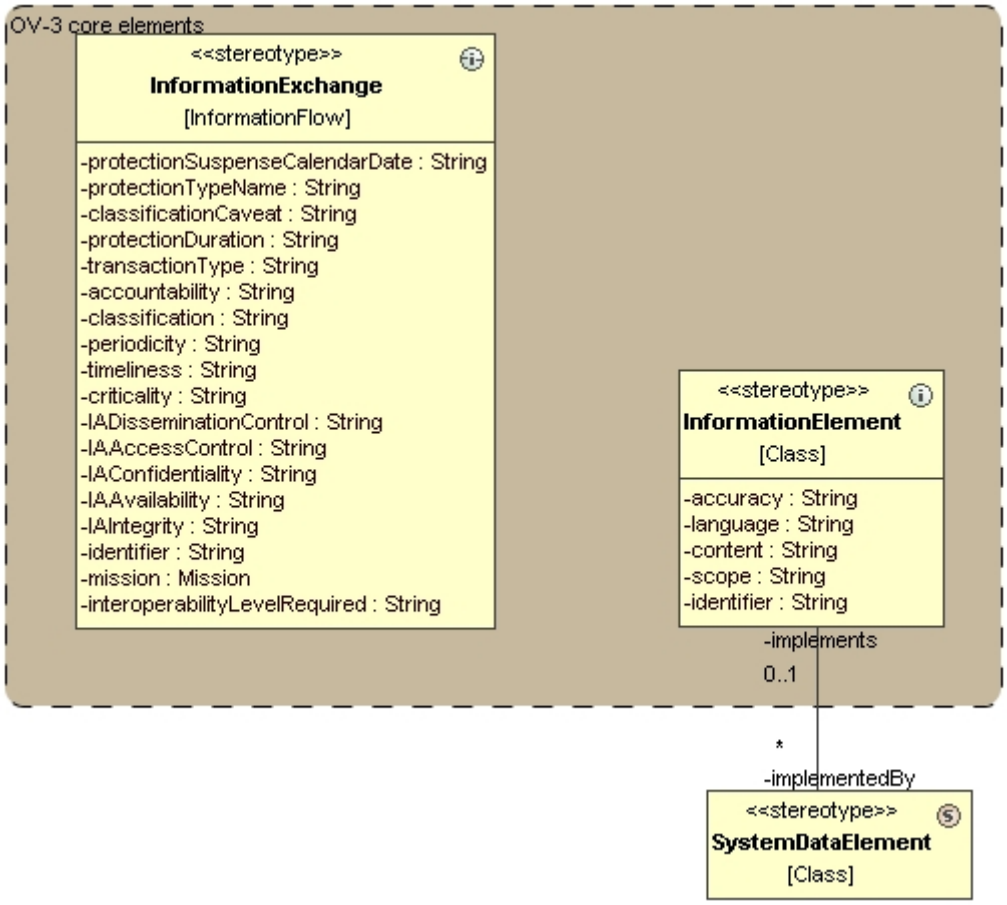

Figure 6 -- OV-3 profile diagram

Creating the product

OV-2 has to be created upfront in order to generate the OV-3.

If OV-2 is created according to MagicDraw's suggested way – Needlines, Information Exchanges and Information Elements are fully described, there is nothing more to model in order to get the OV-3. OV-3 just gathers information about Information Exchanges, Information Elements and attributes of Information Exchange and generates the document. Report may be generated for whole model or for selected scope.

Report template

In Operational Information Exchange Matrix report information about Information Exchange attributes are grouped according to Information Exchange attributes types:

- *Information Exchange Associations* section associates Information Exchange with producing and consuming Operational Nodes and Operational Activities.
- *Information Element Description* section provides information about specific attributes of Information Element: name, identifier, content, scope, accuracy, and language.
- *Nature of Transaction* section includes the list of Information Exchange attributes that specifies nature of transaction: mission/scenario UJTL or METL, transaction type, interoperability level required, and criticality.
- *Performance Attributes* section defines Information Exchange performance attributes: periodicity and timeliness.
- *Information Assurance* section lists Information Exchange information assurance attributes: access control, availability, confidentiality, dissemination control, and integrity.
- *Security* section presents Information Exchange security attributes: accountability, protection (type name, duration, date), classification, and classification caveat.

In each section items are sorted according to Information Exchange identifier.

User can customize report by selecting Information Exchange attributes sections that should be included to the report. Inside each section the list of Information Exchange attributes can also be customized.

Relationship to other products

OV-3 uses most of the **OV-2** information – Needlines, Operational Nodes, and Information Elements.

Sample

2. Information Exchange Associations

Needline Name and Identifier	Information Exchange Name and Identifier	Sending Op Node Name and Identifier	Sending Op Activity Name and Identifier	Receiving Op Node Name and Identifier	Receiving Op Activity Name and Identifier
name NA identifier id JFACC WOC	name flow for Target Material/Analysis identifier id Target2	name JFACC identifier NA operational role NA	name Conduct Battle Damage Assesment identifier 6.3	name WOC identifier NA operational role NA	name Conduct Munitions Effect Assesment identifier 6.2
name NA identifier id DJFCC JFACC	name flow for Target nominations identifier id Target1	name DJFCC (DOCC, 513ACE) identifier NA operational role NA	name Recommend Restrike identifier 6.1	name JFACC identifier NA operational role NA	name Request Target Materials

Figure 7 -- OV-3 sample

OV-4. Organizational Relationship Chart

Overview

"The Organizational Relationships Chart illustrates the command structure or relationships (as opposed to relationships with respect to a business process flow) among human roles, organizations, or organization types that are the key players in architecture. This product clarifies the various relationships that can exist between organizations and sub-organizations within the architecture and between internal and external organizations." *[DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

OV-4 is a diagram that is based on the **UML class diagram**.

User also may use:

- Regular UML Class Diagram
- SysML Block Definition Diagram
- Any other MagicDraw diagram

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Performer	<<Performer>> [Class]	N/A
Organization	<<Organization>> [Class]	
Organization Type	<<OrganizationType>> [Class]	
Role	<<Role>> [Class]	
Person	<<Person>> [Class]	
Responsibility	<<Responsibility>> [Class]	
Organizational Relationship	<<OrganizationalRelationship>> [Usage]	N/A
Direct Organizational Relationship	<<Direct>> [Usage]	N/A
Indirect Organizational Relationship	<<Indirect>> [Usage]	N/A
Coordination Organizational Relationship	<<Coordination >> [Usage]	N/A
Situation Dependent Organizational Relationship	<<SituationDependent>> [Usage]	N/A
Contributing Organizational Relationship	<<Contributing>> [Usage]	N/A
Backup Organizational Relationship	<<Backup>> [Usage]	N/A
Operational Node	See OV-2 data element definitions	
Operational Activity	See OV-5 data element definitions	

Performer

This is an **abstract** stereotype that is a parent for all Organizational Structure elements. Only elements derived from Performer are capable to perform Operational Activity.

Derives from SysML Block (**extends** UML Class).

Attributes

performsAt: OperationalNode [*]	Operational Nodes that this Performer is performing at
performs: OperationalActivity [*]	Operational Activities that are performed by this Performer

Organization

An Organization represents a group of people or organizations that are related for a particular purpose. An Organization element defines a concrete organization, not a generic type.

Derives from Performer.

Attributes

militaryServiceType: MilitaryServiceKind	Military service type. Possible values are defined in the MilitaryServiceKind enumeration: Army Navy Air Force Marine Corps Joint
code/symbol: String	Service office code or symbol
responsibility: Responsibility	Responsibility of the described Organization
type: OrganizationType [1]	Type of this Organization

OrganizationType

A type of the Organization.

Derives from Performer.

Role

A function within Organization or Organization Type.

Derives from Performer.

Attributes

responsibilities: Responsibility [1..*]	Responsibility of the described Role
persons: Person [*]	Persons who are assigned to perform this Role

Performer

An abstract stereotype for defining entities that are capable to perform the Operational Activity.

Derives from SysML Block (**extends** UML Class).

Attributes

performsAt: OperationalNode [*]	OperationalNodes hosting this Performer
performs: OperationalActivity [*]	OperationalActivities performed by this Performer

Person

An actual person within Organization.

Derives from Performer.

Attributes

responsibilities: Responsibility [*]	Responsibilities of the Person
roles: Role [*]	Roles that Person is assigned to perform

Responsibility

Responsibility assigned to an Organization, Role, or Person.

Extends UML Class.

Attributes

persons: Person [*]	Persons that are assigned to perform Responsibility
roles: Role [*]	Roles that are assigned to perform Responsibility

OrganizationalRelationship

Generic relationship used for connecting OV-4 elements. Possible types of organizational relationships:

- Direct Organizational Relationship
- Indirect Organizational Relationship
- Coordination Organizational Relationship
- Situation Dependent Organizational Relationship
- Contributing Organizational Relationship
- Backup Organizational Relationship

All organizational relationships **extend** UML Usage.

Profile diagram

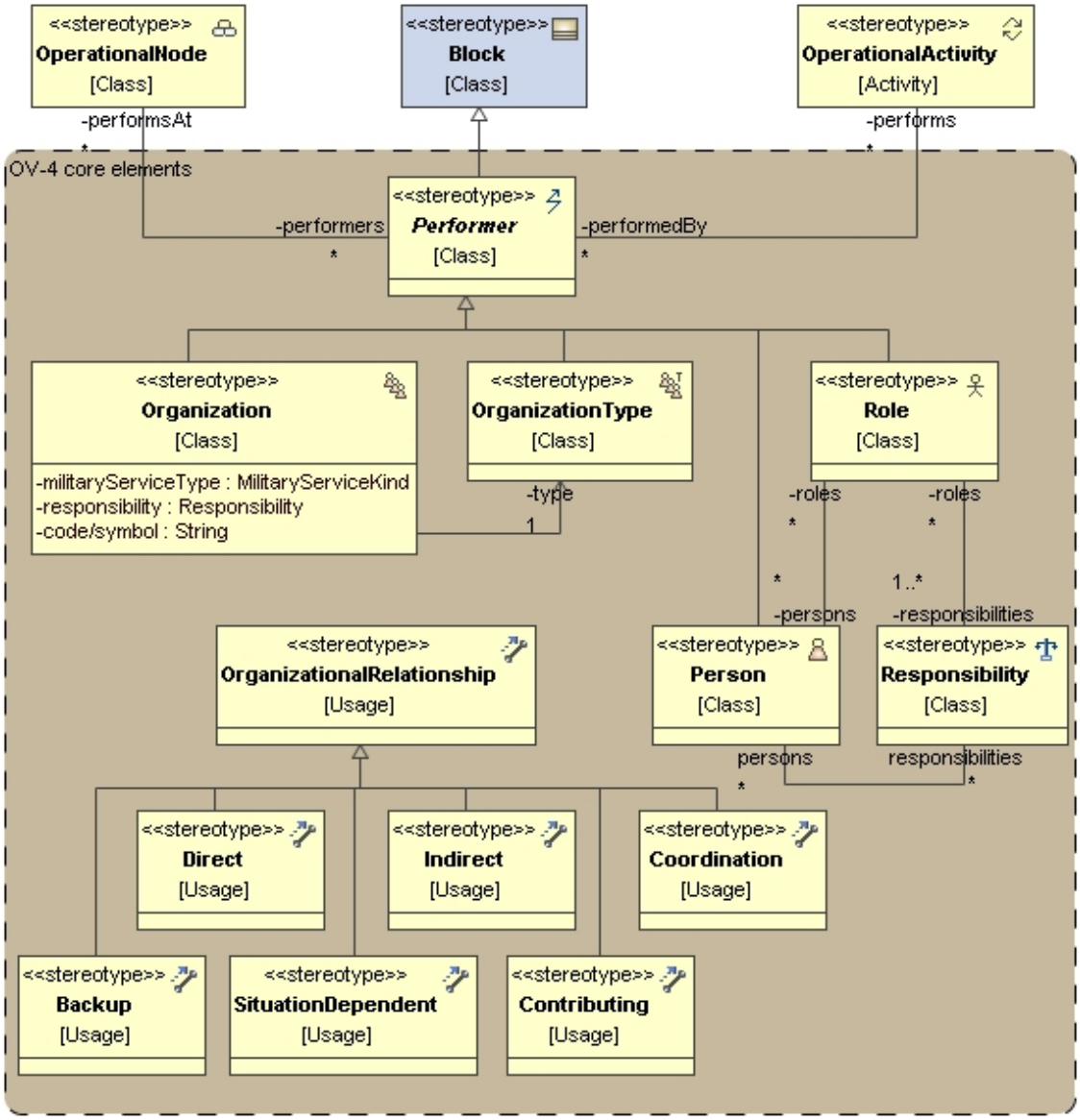

Figure 8 -- OV-4 profile diagram

Creating the product

There is nothing fancy with OV-4 creation. Use the OV-4 specific diagram toolbar when creating the Organizational Relationship Charts.

If you have created OV-4 elements in other products, for example as performers in OV-5, simply drag them into the OV-4 from model browser.

Relationship to other products

Every Performer in OV-4 may have relations to Operational Nodes from **OV-2**, showing what Operational Nodes it performs at (property *performsAt*).

Every Performer in OV-4 may have relations to Operational Activities from **OV-5**, showing what Operational Activities are performed by it (property *performs*).

Sample

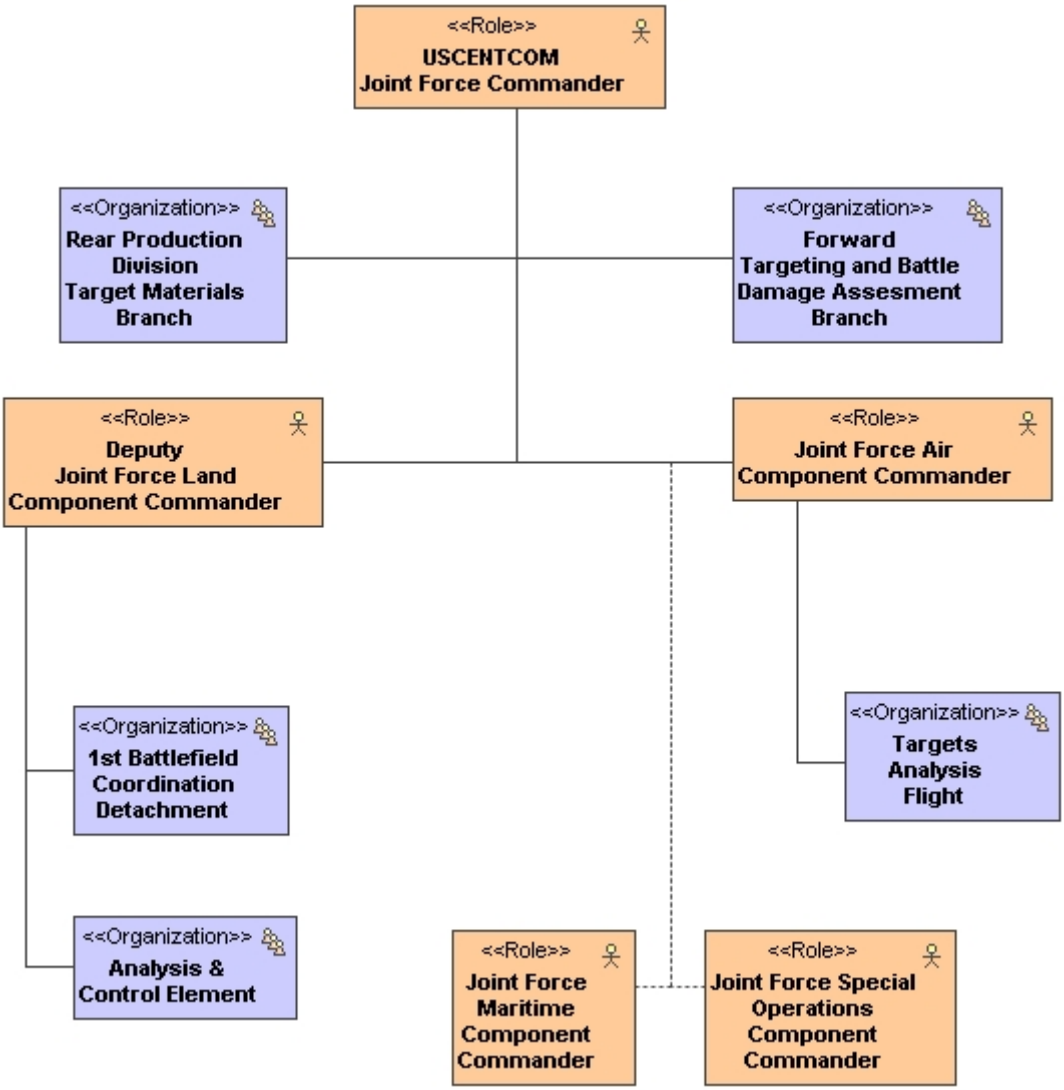

Figure 9 -- OV-4 sample

OV-5. Operational Activity Model

Overview

"The Operational Activity Model describes the operations that are normally conducted in the course of achieving a mission or a business goal. It describes capabilities, operational activities (or tasks), input and output (I/O) flow between activities, and I/O flows to/from activities that are outside the scope of the architecture. High-level operational activities should trace to (are decompositions of) a Business Area, an Internal Line of Business, and/or a Business Sub-Function as published in OMB's Business Reference Model [OMB, 2003]". [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

As OV-5 should show hierarchies and flows of the Operational Activities, there is no single UML/SysML diagram that is able to do so. Therefore we suggest having two different products of the OV-5:

- **OV-5 diagram** that is based on **UML Class diagrams** to define the hierarchy of Operational Activities.
- **OV-5 diagram** that is based on **UML Activity diagrams** to define the flow of the Operational Activities.

User also may use:

- Regular UML Class Diagram
- Regular UML Activity Diagram
- SysML Block Definition Diagram
- SysML Activity Diagram
- Any other MagicDraw diagram

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Operational Activity	<<OperationalActivity>> [Activity]	
Information Flow	<<InformationFlow>> [ControlFlow, ObjectFlow]	N/A
Capability	<<Capability>> [Use Case]	
Operational Activity Action	<<OperationalActivityAction>> [CallBehaviorAction]	N/A
Operational Node	See OV-2 data element definitions	
Information Exchange	See OV-3 data element definitions	
Information Element	See OV-3 data element definitions	
Performer	See OV-4 data element definitions	
System Function	See SV-4 data element definitions	

OperationalActivity

Operational Activity is an activity or process performed by Performer.

Extends UML Activity.Attributes

levelIndentifier: String	If using hierarchical decomposition of activities: Identifier that corresponds to the activity's place in the activity hierarchy – should be unique
cost: float	Cost for activity derived from or used in activity-based cost analysis
parent: OperationalActivity [0..1]	Parent Operational Activity
subactivity: OperationalActivity [*]	Child Operational Activities
consumes: InformationExchange [*]	Information Exchanges consumed by this Operational Activity

produces: InformationExchange [*]	Information Exchanges produced by this Operational Activity
performedBy: Performer [*]	Performers of this Operational Activity. **Note**: Performer may be any specific element of Performer class: Organization, Organization Type, Role, Person
performedAt: OperationalNode [*]	Operational Nodes at which this Operational Activity is performed
implementedBy: SystemFunction [*]	System Functions that implement this Operational Activity
capability: Capability [*]	Capabilities to which this Operational Activity belongs

InformationFlow

Relationship in OV-5 Operational Activity Flow used to show the Information Flows between Operational Activities.

Extends UML ControlFlow, UML ObjectFlow.

Attributes

type: InformationElement [1]	Information Element carried by flow.

Capability

The Capability of the subject system describes a function or action that the system can perform.

Extends UML UseCase.

Attributes

mission: Mission	Joint Mission Area, cross-mission area domain, Universal Joint Task List activity, related specific scenario, or other task-related basis of the architecture
operationalThread: OperationalActivity [*]	Operational Activities to which Capability is related

OperationalActivityAction

Call action that invokes Operational Activity directly.

Extends UML CallBehaviorAction.

Profile diagram

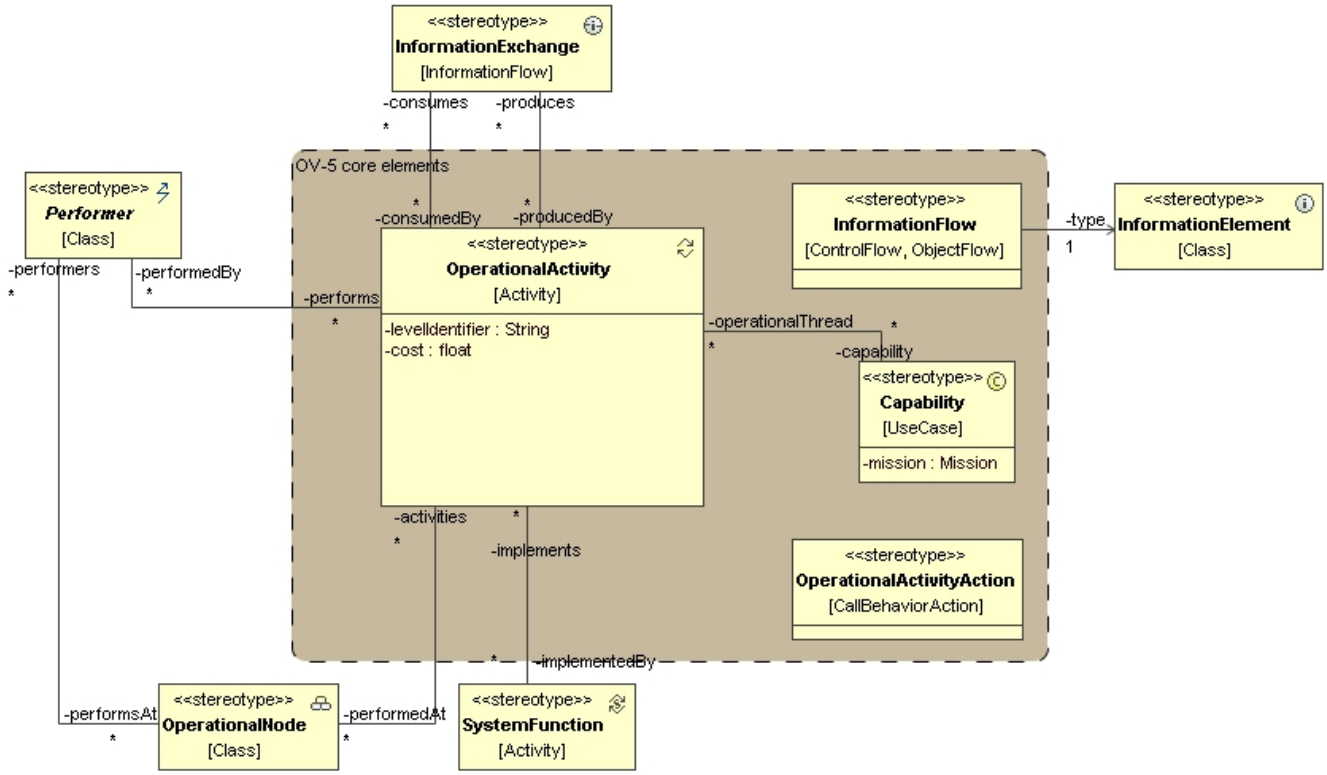

Figure 10 -- OV-5 profile diagram

Creating the product

We do recommend starting with the OV-5 Operational Activity Flow, because MagicDraw will be able to create OV-5 Operational Activity Hierarchy automatically from the data in the OV-5 flow definition.

OV-5 Operational Activity Hierarchy

Use OV-5 specific toolbar to create hierarchies of the **Operational Activities**. Use **Composition (Associaton)** as main path between **Operational Activities**. As **UML Activity** is a **UML Classifier**, it can be used in the structure diagrams.

Capabilities may also be modeled in the OV-5 Operational Activity Hierarchy.

OV-5 Operational Activity Flow

This is very similar to **UML Activity diagram**. Drawing **Operational Activities** will result in creation of **Call Behavior Actions** with **Operational Activity** assigned as behavior.

Relationship to other products

Operational Activities may have consumed or produced Information Exchanges from **OV-2** (properties *consumes* and *produces*).

Operational Activities are performed at Operational Nodes from **OV-2** (property *performedAt*)

System functions from **SV-4** are implementing the Operational Activities (property *implementedBy*)

Operational Activity may be performed by Organizational structure elements in **OV-4** (property *performedBy*)

Sample

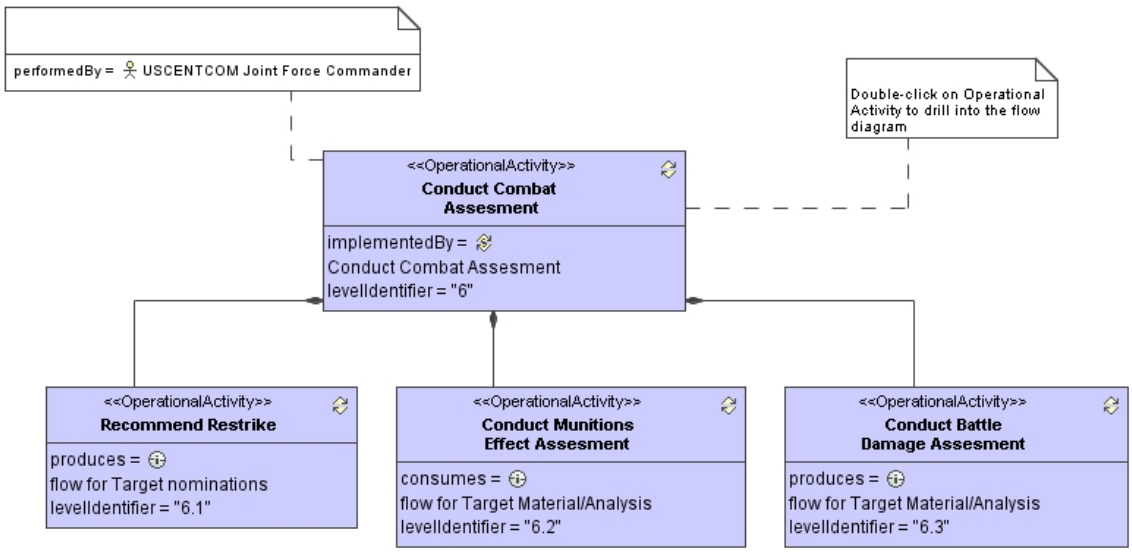

Figure 11 -- OV-5 sample (hierarchy)

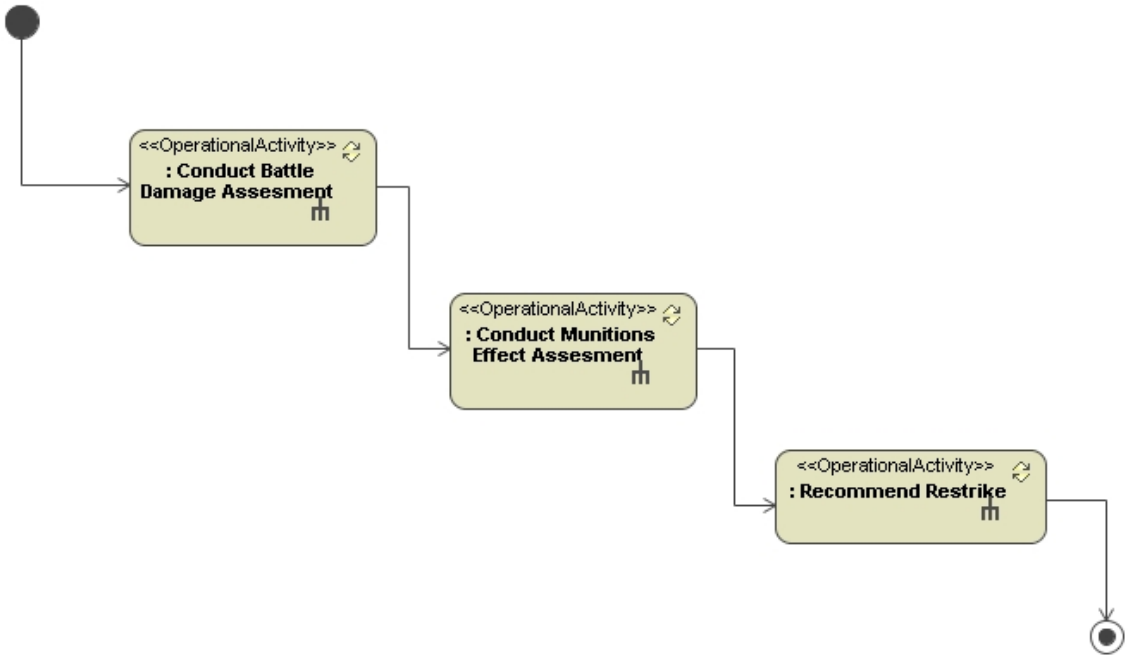

Figure 12 -- OV-5 sample (flow)

OV-6a. Operational Rules Model

Description

"The Operational Rules Model specifies operational or business rules on an enterprise, a mission, operation, business, or architecture. While other OV products (e.g., OV-1, OV-2, and OV-5) describe the structure of a business—what the business can do—for the most part, they do not describe what the business must do, or what it cannot do." [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

Operational Rule is implemented as **UML Constraint** in MagicDraw, so every DoDAF element in MagicDraw project can have an Operational Rule assigned.

OV-6a is a **document**, which is implemented as MagicDraw **report**, using model information mainly from OV products.

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Operational Rule	<<OperationalRule>> [Constraint]	{o}

Operational Rule

Rules are statements that define or constrain some aspect of the mission, or the architecture. It is intended to assert operational structure or to control or influence the mission thread.

Extends UML Constraint.

Attributes

type: RuleType [1] = Structural Assertion	Operational Rule type. Possible values are defined in the RuleType enumeration: Structural Assertion Action Assertion Derivation

Profile diagram

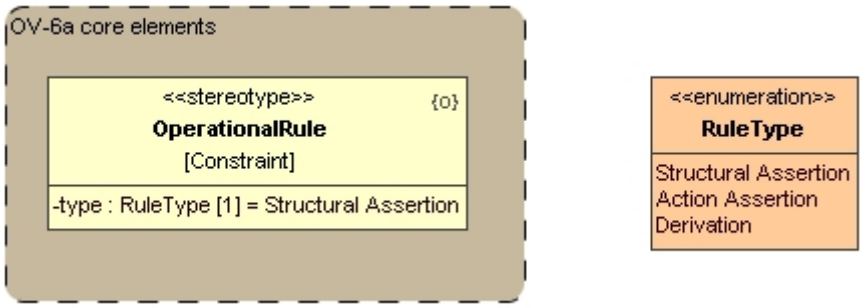

Figure 13 -- OV-6 profile diagram

Creating the product

In general OV-6a is just a collection of **Operational Rules** (**UML Constraints**) from the model compiled in a document. Report may be generated for whole project or for selected scope.

An Operational Rule can be applied to **any** element in the DoDAF model.

Report template

In Operational Rules Model report Operational Rules are grouped according to type:

- *Structural assertions* are added to “Structural Assertions” section of the report.
- *Action assertions* are added to “Action Assertions” section of the report.

- *Derivations* are added to “Derivations” section of the report.

Inside each group Operational Rules are listed in numbered manner and sorted according to Operational Rule name. Below each Operational Rule name constraint specification and information about constrained elements is provided.

User can customize report by selecting the type of Operational Rules, for which report should be generated. The list of constrained elements may also be included optionally.

Relationship to other products

The OV-6a is a collection of Operational Rules from the rest of the model.

Sample

1.Structural Assertions

1. ATO development

ATOs are developed on 24-96 hour planning cycle.

2. DBA development

BDA are developed for every target that has an air sortie directed against it.

3. MEA development

MEAs are developed to identify deficiencies in weapon system/munitions performance, tactics, or aim point selection

4. Threshold Nonfullfilment

Targets for which the BDA/MEA falls below a specified threshold (as developed by the Joint Force Commander [JFC]/Joint Force Air Component Commander [JFACC] in accordance with the Commander’s intent) are recommended for re-strike

Figure 14 -- OV-6a sample

OV-6b. Operational State Transition Description

Description

“The Operational State Transition Description is a graphical method of describing how an operational node or activity responds to various events by changing its state. The diagram represents the sets of events to which the architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

An Operational State Transition Description can be used to describe the explicit sequencing of the operational activities. Alternatively, OV-6b can be used to reflect the explicit sequencing of actions internal to a single operational activity or the sequencing of operational activities with respect to a specific operational node”. [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

OV-6b is a diagram that is based on **UML State Machine diagram**.

UML Profile for DoDAF

No additional mapping to UML is needed for producing OV-6b.

Creating the product

This product is created as a UML State Machine diagram.

Relationship to other products

Operational Activities from **OV-5** may be set as activities for the state's Entry, Do Activity and Exit Actions.

Operational Nodes from **OV-2** may have state machines as UML property O*wned Behavior*.

Information Elements from **OV-3** may have state machines as UML property O*wned Behavior*.

Organizational structure elements from **OV-4** may have state machines as UML property O*wned Behavior*.

Operational Activities from **OV-5** may have state machines as UML property O*wned Behavior*.

Sample

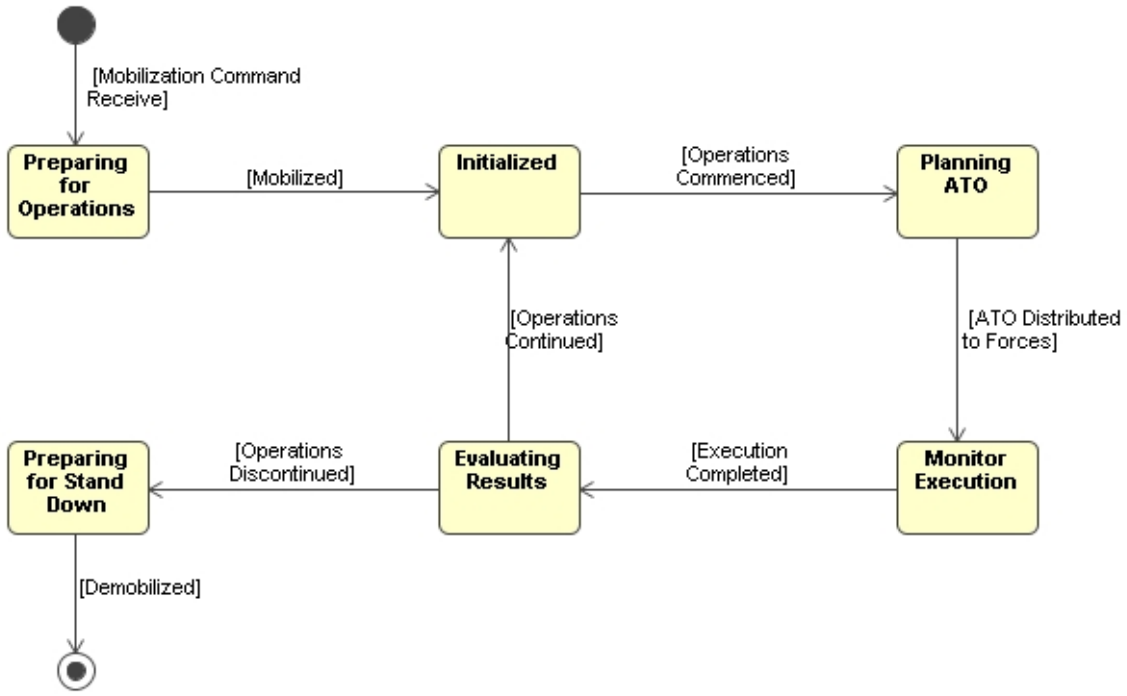

Figure 15 -- OV-6b sample

OV-6c. Operational Event-Trace Description

Description

"The Operational Event-Trace Description provides a time-ordered examination of the information exchanges between participating operational nodes as a result of a particular scenario. Each event-trace diagram should have an accompanying description that defines the particular scenario or situation.

OV-6c is valuable for moving to the next level of detail from the initial operational concepts. The product helps define node interactions and operational threads. The OV-6c can also help ensure that each participating operational node has the necessary information it needs at the right time in order to perform its assigned operational activity.

OV-6c can be used by itself or in conjunction with OV-6b to describe the dynamic behavior of business processes or a mission/operational thread." [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

OV-6c is a **diagram** that is based on **UML Sequence diagram**.

UML Profile for DoDAF

No additional mapping to UML is needed for producing OV-6c.

Creating the product

This is a regular UML sequence diagram. In order to create lifelines you may drag objects from other DoDAF products from the data browser to this diagram.

Relationship with other products

Operational Nodes from **OV-2** may be used as types for the interaction attributes that will be set as *represents* property for the lifelines in OV-6c.

Sample

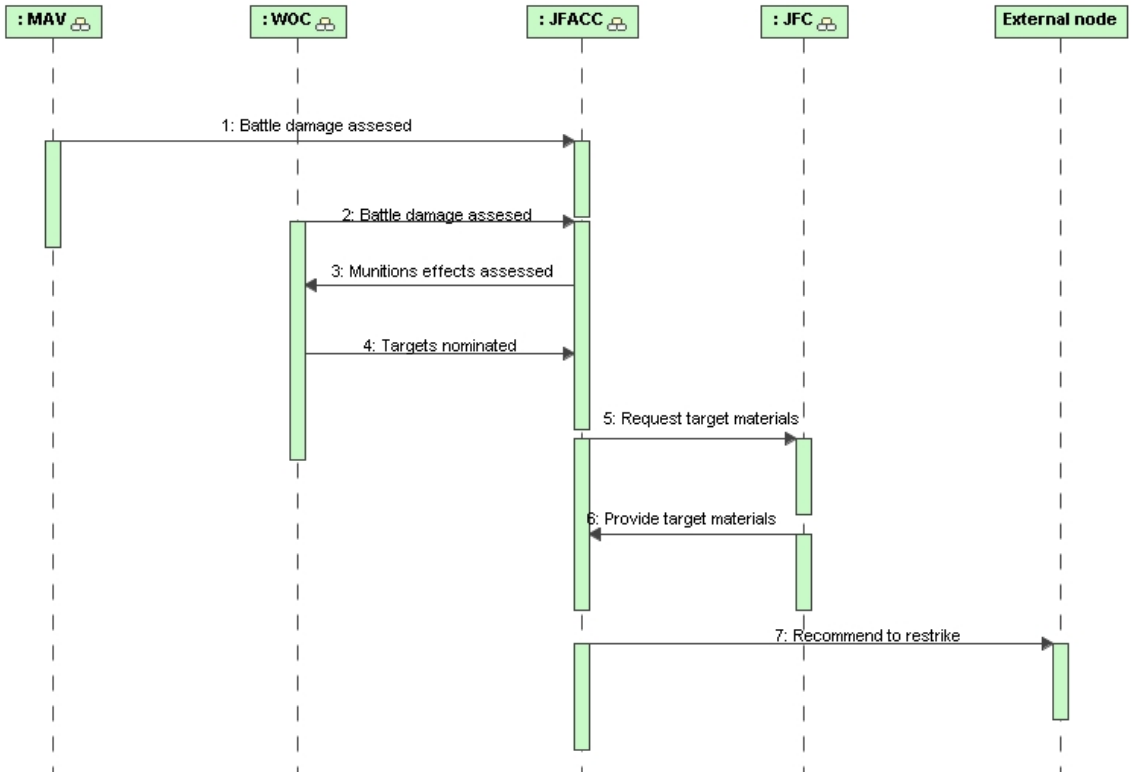

Figure 16 -- OV-6c sample

OV-7. Logical Data Model

Description

"The Logical Data Model describes the structure of an architecture domain's system data types and the structural business process rules (defined in the architecture's Operational View) that govern the system data. It provides a definition of architecture domain data types, their attributes or characteristics, and their interrelationships. OV-7 is an architecture product and describes information about a specific architecture domain. The architecture data elements for OV-7 include descriptions of entity, attribute, and relationship types. Attributes can be associated with entities and with relationships, depending on the purposes of the architecture." [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

OV-7 is a **diagram** that is based **UML Class Diagram**.

User also may use:

- Regular UML Class Diagram
- SysML Block Definition Diagram
- Any other MagicDraw diagram

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Information Element	See OV-3 data element definitions	
System Data Element	See SV-6 data element definitions	

Profile diagram

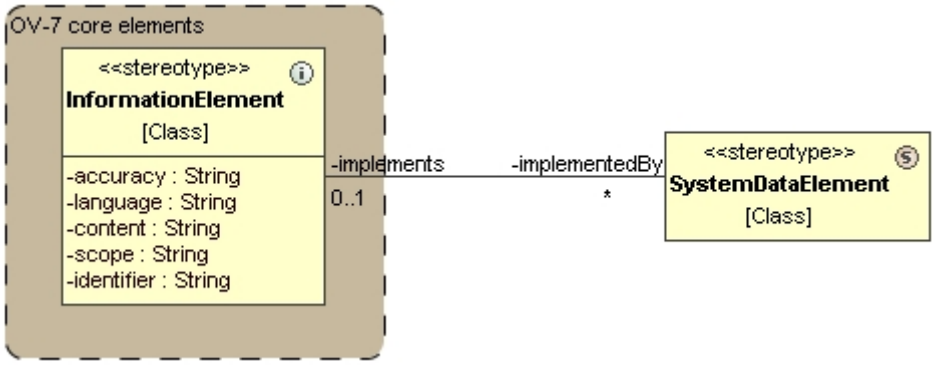

Figure 17 -- OV-7 profile diagram

Creating the product

OV-7 is a regular class diagram with **Information Element** used instead of a **UML Class**.

It is likely that a lot of **Information Elements** will be created by this diagram creation, so user can just drag those elements to the diagram from the browser tree.

Relationship with other products

Information Elements defined in **OV-2/OV-3** are reused in OV-7.

In **SV-6** System Data Element may implement Information Element (property *implementedBy*).

Sample

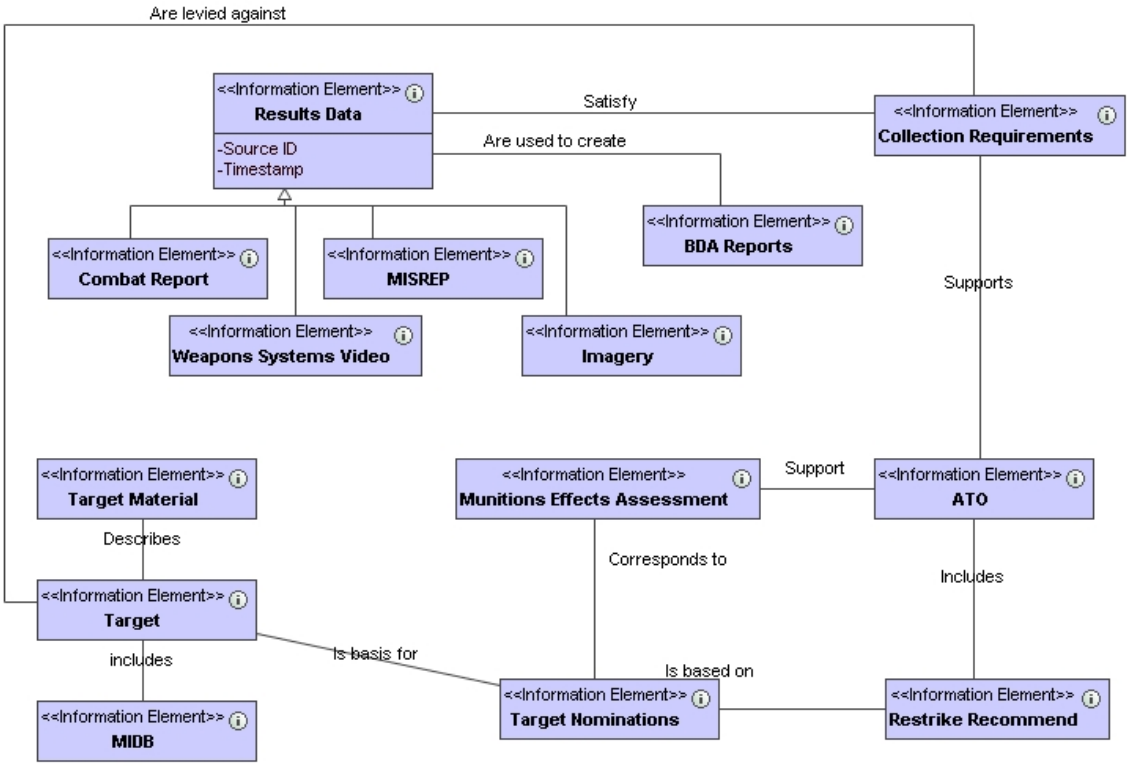

Figure 18 -- Figure 22 OV-7 sample

5 SYSTEMS VIEWS PRODUCTS

SV-1. Systems Interface Description

Description

"The Systems Interface Description depicts systems nodes and the systems resident at these nodes to support organizations/human roles represented by operational nodes of the Operational Node Connectivity Description (OV-2). SV-1 also identifies the interfaces between systems and systems nodes. OV-2 depicts the operational nodes representing organizations, organization types, and/or human roles, while SV-1 depicts the systems nodes that house operational nodes (e.g., platforms, units, facilities, and locations) and the corresponding systems resident at these systems nodes and which support the operational nodes. In addition to depicting systems nodes and systems, SV-1 addresses system interfaces. An interface, as depicted in SV-1, is a simplified, abstract representation of one or more communications paths between systems nodes or between systems." [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

SV-1 is a **diagram**, which is based on either **UML Class Diagram** or **UML Composite Structure Diagram** depending on the level of details.

User also may use:

- Regular UML Class Diagram
- Regular UML Composite Structure Diagram
- UML Implementation Diagram
- SysML Block Definition Diagram
- SysML Internal Block Diagram
- Any other MagicDraw diagram

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
System	<<System>> [Class]	
Systems Node	<<SystemsNode>> [Class]	
Interface	<<Interface>> [Association, Connector]	N/A
System Usage	<<SystemUsage>> [Property]	
Systems Node Usage	<<SystemsNodeUsage>> [Property]	
Hardware/Software Item	<<Hardware/SoftwareItem>> [Class]	
Organizational Resource Usage	See OV-2 data element definitions	
Operational Node	See OV-2 data element definitions	
Needline	See OV-2 data element definitions	
ServiceSpecification	See OV-2 data element definitions	
SoaService	See OV-2 data element definitions	
Interface Implementer	See SV-2 data element definitions	
System Function	See SV-4 data element definitions	
System Data Exchange	See SV-6 data element definitions	
System Data Element	See SV-6 data element definitions	

System

Description

Any organized assembly of resources and procedures united and regulated by interaction or interdependence to accomplish a set of specific functions. A term *system* in the framework is used to denote a family of systems (FoS), system of systems (SoS), nomenclature system, or a subsystem.

Derives from SysML Block, **extends** UML Class.

Attributes

participant: Participant	Defines participant role that is applied to System. Possible values are taken from Participant enumeration: System Service Provider System Service Consumer
performedFunctions: SystemFunction [*]	System functions performed by system
performanceParameterSet: PerformanceParameterSet [*]	Performance Parameter Sets specified for System
measurements: PerformanceMeasurementSet [*]	A list of Performance Measured Sets with measured Performance Parameter Set values

SystemsNode

Description

A node with the identification and allocation of resources (e.g., platforms, units, facilities, and locations) required implementing specific roles and missions. Usually denotes a facility where an Operational Node is located.

Derives from SysML Block, **extends** UML Class.

Attributes

implements: OperationalNode [*]	Operational Nodes housed by Systems Node
allocatedFunctions: SystemFunction [*]	System Functions allocated at Systems Node

Interface

Description

An interface is the abstract representation of one or more Communication Paths between Systems Nodes or between Systems. An SV-1 Interface is the systems representation of OV-2 Needline.

Extends UML Association (in block definition diagrams), UML Connector (in internal block diagrams)

Attributes

supportedNeedline: Needline [1]	Needline depicted by Interface
implementedBy: InterfaceImplementer [*]	A set of Communications Links and Communications Paths that implement Interface

SystemUsage

Description

System Usage is a helper element that is needed to show a System inside other System. We are using SysML based approach here: parent System will have a System Usage as a SysML BlockProperty. The type of the System Usage will be child Systems Node.

Derives from SysML BlockProperty (**extends** Property).

Note. This is the same as Part in UML composite structure diagrams.

participant: Participant	Defines participant role that is applied to System Usage. Possible values are taken from Participant enumeration: System Service Provider System Service Consumer

SystemsNodeUsage

Description

This is a helper element similar to System Usage.

Derives from SysML BlockProperty (**extends** Property).

Note. This is the same as Part in UML composite structure diagrams.

OrganizationalResourceUsage

Description

Organizational Resource Usage is a helper element that is needed to show an Organizational Resource inside System or Systems Node. We are using SysML based approach here; The parent System or Systems Node will have a Organizational Resource Usage described as a SysML BlockProperty.

Derives from SysML BlockProperty (**extends** Property).

Note. This is the same as Part in UML composite structure diagrams.

Hardware/SoftwareItem

Description

Describes the Software and/or hardware items of a system (or subsystem). Represents a software application or hardware equipment that has a serial number (or other identifier).

Derives from System.

Attributes

vendors/source: String	Source of system Software or Hardware Item

Profile diagram

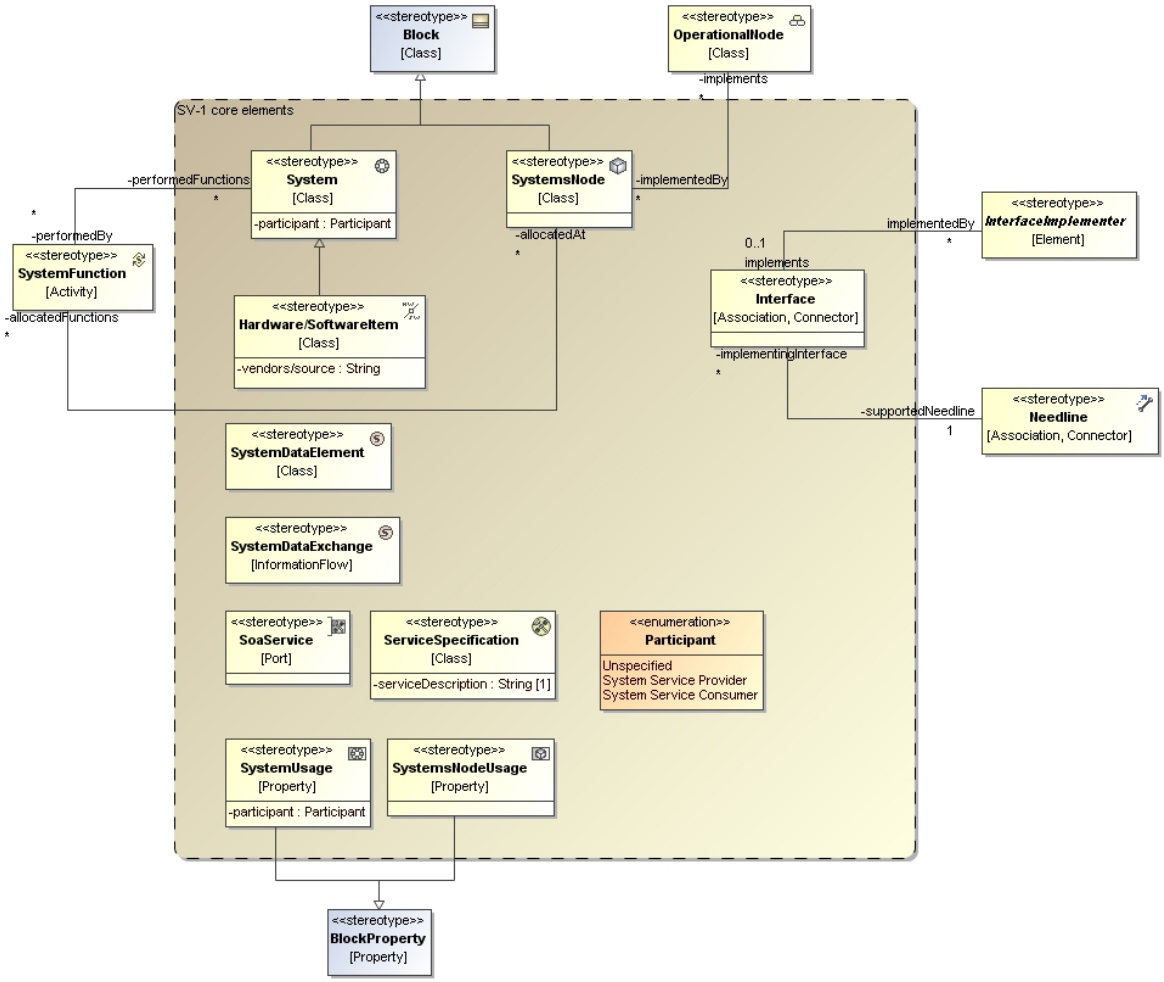

Figure 1 -- SV-1 profile diagram

Creating the product

It is recommended at first to create the **SV-1 Systems Interface Descriptions** with main Systems and System Nodes.

If you have composite structure for the Systems or System Nodes, you need to create an **SV-1 Systems Internal Interface Description** for the Systems/System Nodes that have sub-elements. Sub-elements will be created as System Usages and Systems Node Usages in the parent System/Systems Node. Child elements will be set as type for the Operational Node Usages.

Node's internal diagram can be easily created from a System/Systems Node (context menu -> New Diagram -> SV-1 Systems Internal Interface Description).

Modeling service architectures

SV-1 product may be used for modeling service architectures. System and sub systems (System Usages) have "Participant" property that allows you to specify the responsibility System plays: System Service Consumer or System Service Provider.

System Service Consumer and System Service Provider can communicate only via Soa Service. Since Soa Service extends UML Port System Service Provider and System Service Consumer should be realizing classifiers of the same component. To model this on **SV-1 Systems Interface Description** diagram create Component and drag on it Systems. Systems will become realizing classifiers of the same Component and you will be able to connect their Soa Services using Interface [connector] relationship.

The type of Soa Service can be UML Interface or Service Specification. Use Service Specification when there is behavior for using Soa Service. After behavior addition you may detail it using any type of behavior diagram. After this double-click on Service Specification will open not Service Specification specification dialog but associated behavior diagram.

Relationships to other products

Systems Node may implement the Operational Node from **OV-2** (property *Implements*).

Systems Node may show allocated System Functions from **SV-4** (property *allocatedFunctions*)

Interface may carry System Data Exchanges from **SV-6** (property *dataExchange*).

Interface may support Needline from **OV-2** (property *supportedNeedline*).

System and Hardware/Software Items may show performed System Functions (property *performedFunctions*)

Interface may show implementing Communication Links (property *implementedBy*)

Sample

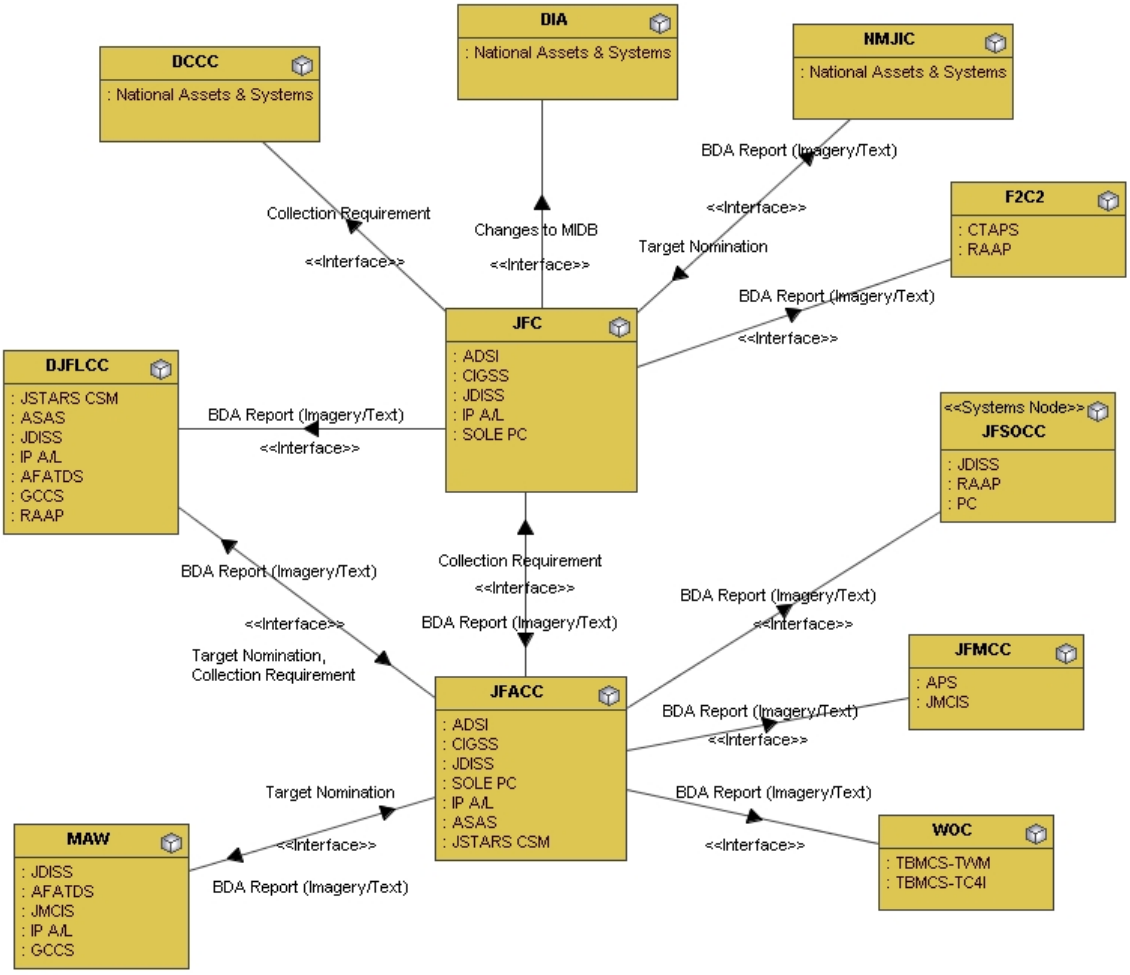

Figure 2 -- SV-1 sample (Systems Nodes)

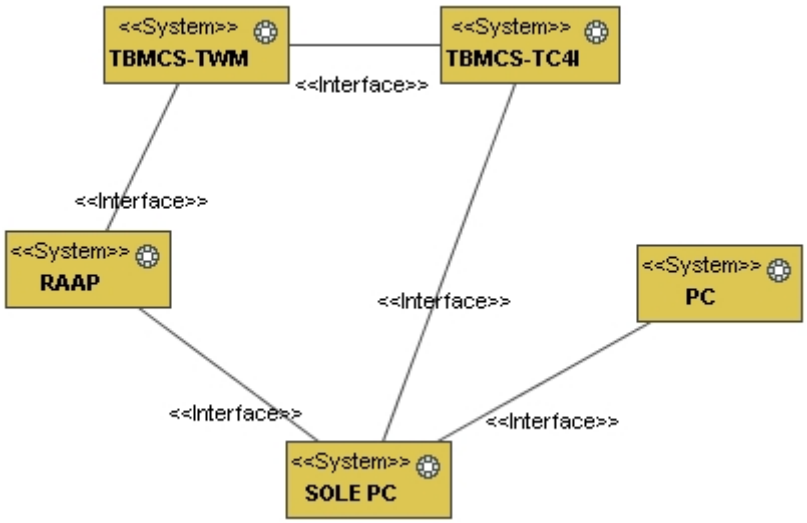

Figure 3 -- SV-1 sample (Systems)

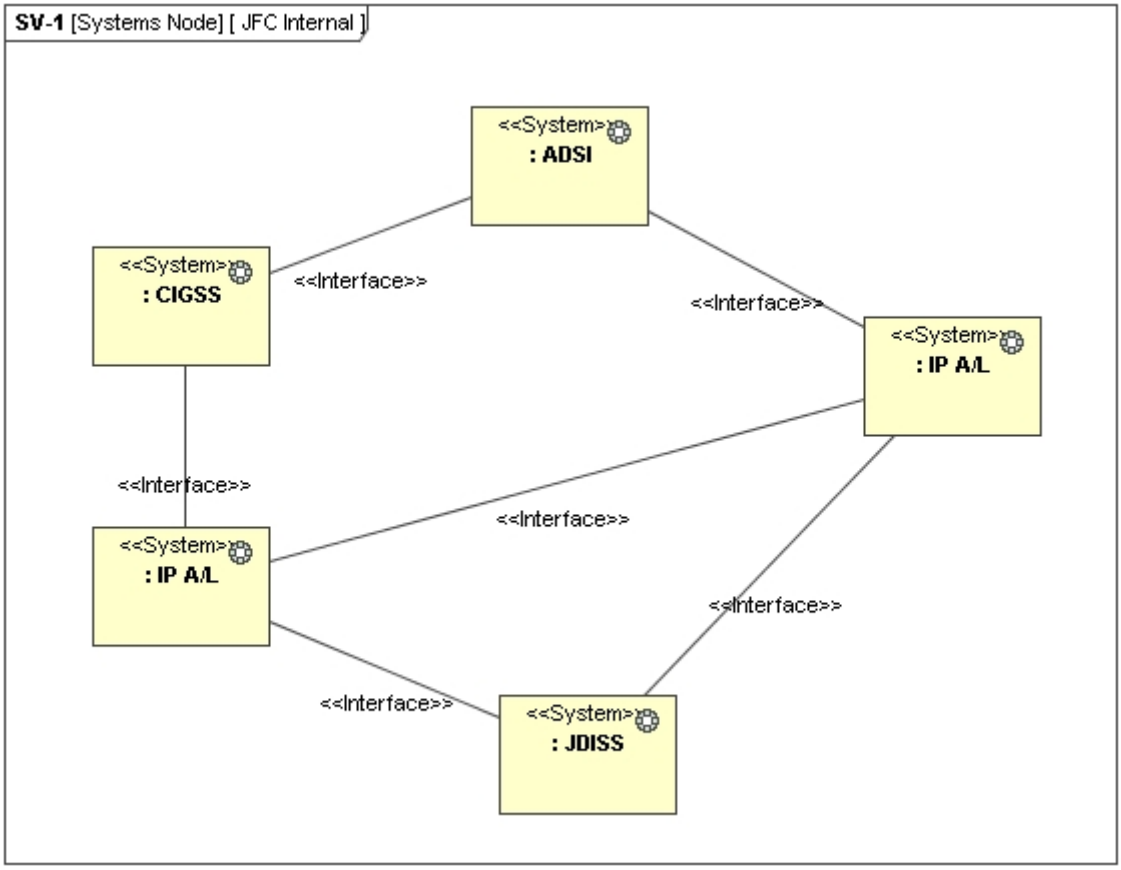

Figure 4 -- SV-1 sample (Systems Node internal)

SV-2. Systems Communications Description

Description

"The Systems Communications Description depicts pertinent information about communications systems, communications links, and communications networks. SV-2 documents the kinds of communications media that support the systems and implement their interfaces as described in SV-1. Thus, SV-2 shows the communications details of SV-1 interfaces that automate aspects of the *Needlines* represented in OV-2." [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

SV-2 is a **diagram**, which is based on either **UML Class Diagram** or **UML Composite Structure Diagram** depending on the level of details.

User also may use:

- Regular UML Class Diagram
- Regular UML Composite Structure Diagram
- UML Implementation Diagram
- SysML Block Definition Diagram
- SysML Internal Block Diagram

- Any other MagicDraw diagram

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Interface Implementer	<<InterfaceImplementer>> [Element]	N/A
Communications System	<<CommunicationsSystem>> [Class]	
Communications Link	<<CommunicationsLink>> [Association, Connector]	N/A
Communications Path	<<CommunicationsPath>> [Class]	
Communications Network	<<CommunicationsNetwork>> [Class]	
LAN	<<LAN>> [Class]	
WAN	<<WAN>> [Class]	
MAN	<<MAN>> [Class]	
Interface	See SV-1 data element definitions	
System Usage	See SV-1 data element definitions	

InterfaceImplementer

This is an **abstract** stereotype that is a parent for Communications Link and Communications Path elements. Only elements derived from Interface Implementer are capable to implement Interface.

Extends UML Element.

Attributes

Implements: Interface [0..1]	Interface implemented by Interface Implementer

CommunicationsSystem

Description

A Communications System is a System whose primary function is to control the transfer and movement of system data as opposed to performing mission application processing. Examples include switches, routers, and communications satellites.

Derives from System.

CommunicationsLink

Description

A single physical connection from one System (or Systems Node) to another.

Derives from InterfaceImplementer, **extends** UML Association (in block definition diagrams), UML Connector (in internal block diagrams

Attributes

capacity: String	Channel capacity
infrastructureTechnology: String	Infrastructure technology supporting Communications Link
communicationPath: CommunicationsPath [*]	Communications Path containing Communications Link
performanceParameterSet: PerformanceParameterSet [*]	Performance Parameter Set specified for Communications Link
measurements: PerformanceMeasurementSet [*]	A list of Performance Measured Sets with measured Performance Parameter Set values

CommunicationsPath

Description

A (connected) sequence of Communications Systems and Communications Links originating from one System (or Systems Node) and terminating at another System (or Systems Node).

Communications Path will contain an ordered list of Communications Links.

Derives from InterfaceImplementer, **extends** UML Class.

Attributes

communicationLinks: CommunicationsLink [1..*]	Communications Links contained by Communications Path

CommunicationsNetwork

Description

Communications Network is a set of Systems and Communications Links.

Derives from System.

Attributes

securityClassification: String	Classification of System Data that Communications Network is allowed to carry

LAN

Description

A subtype of Communications Network defining local area network (LAN).

Derived from CommunicationsNetwork.

WAN

Description

A subtype of Communications Network defining wide area network (WAN).

Derived from CommunicationsNetwork.

MAN

Description

A subtype of Communications Network defining metropolitan area network (MAN).

Derived from CommunicationsNetwork.

Profile diagram

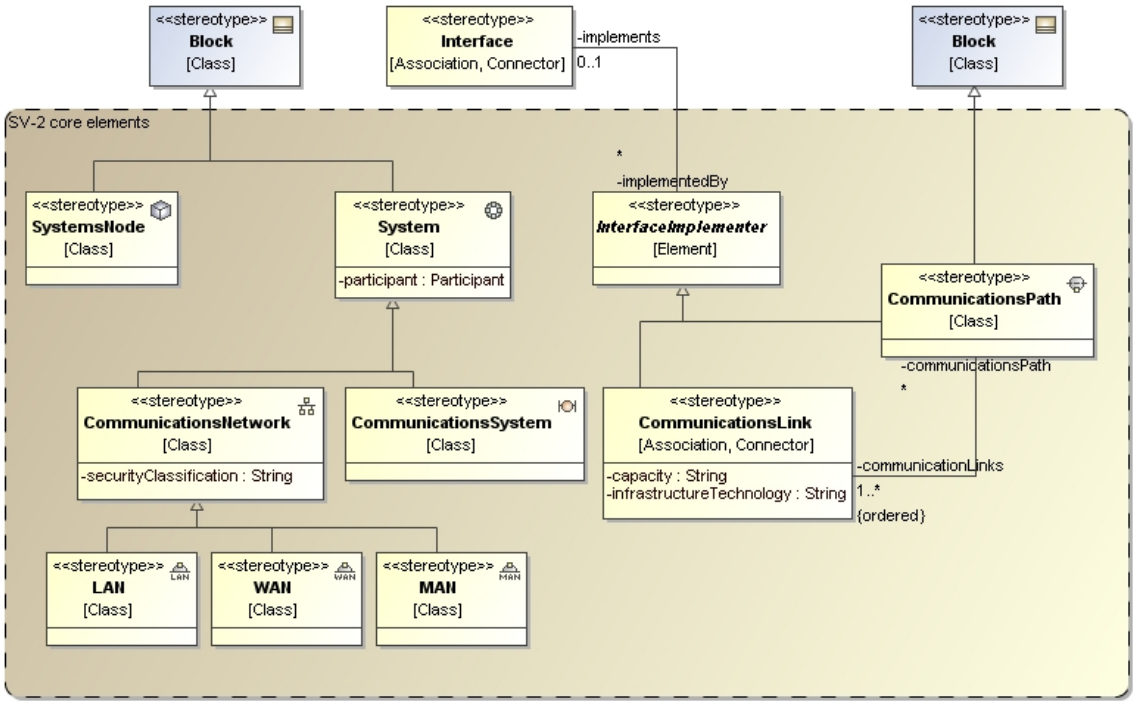

Figure 5 -- SV-2 profile diagram

Creating the product

Creation of SV-2 is very similar to SV-1.

SV-2 is a refinement of SV-1 that shows how Interfaces are implemented with actual Communication Links. So most of the System elements have to be created to this point. In MagicDraw simply drag those elements from the Data Browser to the SV-2 diagram.

Relationships to other products

Interfaces from **SV-1** are implemented by Communication Links in SV-2 (property *implements*).

Sample

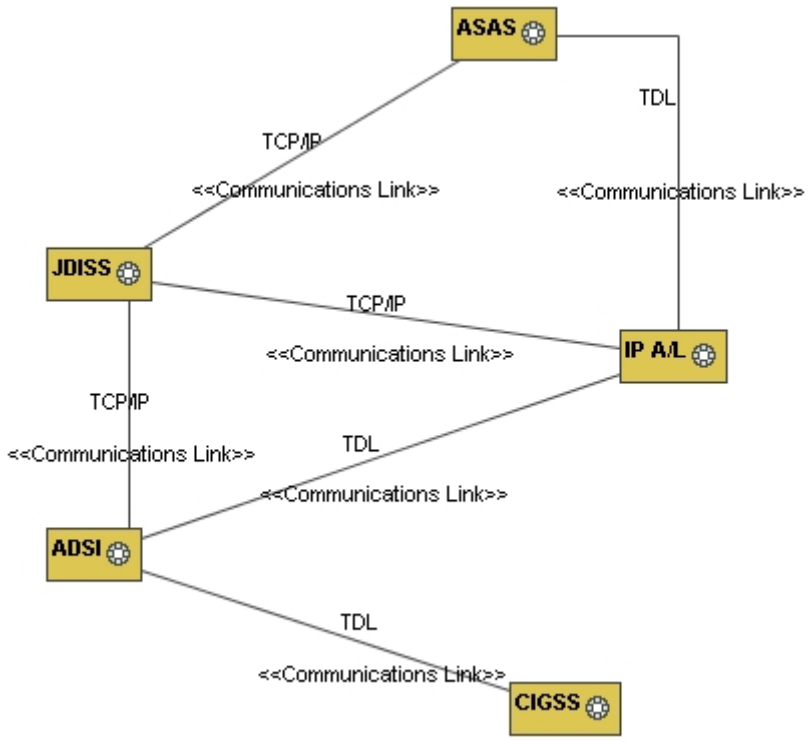

Figure 6 -- SV-2 sample

SV-3. Systems-Systems Matrix

Description

"The Systems-Systems Matrix provides detail on the interface characteristics described in SV-1 for the architecture, arranged in matrix form. It allows a quick overview of all interface characteristics presented in multiple SV-1 diagrams. The matrix form can support a rapid assessment of potential commonalities and redundancies (or, if fault-tolerance is desired, the lack of redundancies).

SV-3 is similar to an N2-type matrix, where the systems are listed in the rows and columns of the matrix, and each cell indicates a system pair interface, if one exists." [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

SV-3 is a **matrix** based on **dependency matrix** in MagicDraw.

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
System	See SV-1 data element definition	
Interface	See SV-1 data element definition	

Profile Diagram

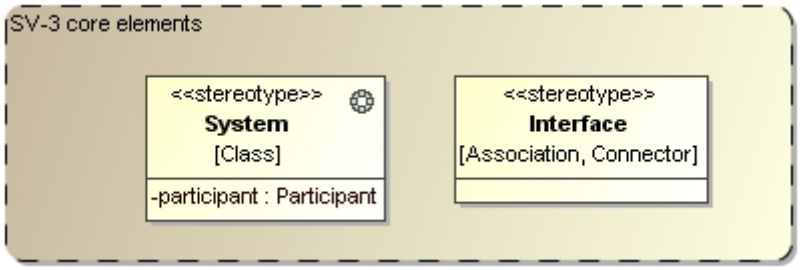

Figure 7 -- SV-3 profile diagram

Creating the product

When user will create SV-3 in the row scope he will be provided with a scope selection (root package by default) for the SV-3. MagicDraw will find all Systems in the given scope will put them as column and row elements. The matrix cells will be filled according to the interfaces between the Systems.

Relationships to other products

SV-3 uses Systems and Interfaces from **SV-1** and **SV-2**.

Sample

	PC [Systems View...	RAAP [Systems Vi...	SOLE PC [System...	TBMCS-TC4I [Sys...	TBMCS-TWM [Sys...
SV-1 [Systems View]	1	2	3	2	2
PC [Systems View::SV-1]			/		
RAAP [Systems View::SV-1]			/		/
SOLE PC [Systems View::S...	/	/		/	
TBMCS-TC4I [Systems Vie...			/		/
TBMCS-TWM [Systems Vie...		/		/	

Figure 8 -- SV-3 sample

SV-4. Systems Functionality Description

Description

”SV-4 describes system functions and the flow of system data among system functions. It is the SV counterpart to OV-5. The scope of this product may be enterprise wide, without regard to which systems perform which functions, or it may be system specific. Variations may focus on intra-nodal system data flow, inter-nodal system data flow, and system data flow without node considerations, function to system allocations, and function to node allocations. Like OV-5, SV-4 may be hierarchical in nature and may have both a hierarchy or decomposition model and a system data flow model”. [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

The implementation of SV-4 is based on the same principles as OV-5.

As SV-4 should show hierarchies and flows of the System Functions, there is no single UML/SysML diagram that is able to do so. Therefore we suggest having two different products of the SV-4:

- **SV-4 diagram** that is based on **UML Class diagrams** to define the hierarchy of System Functions.
- **SV-4 diagram** that is based on **UML Activity diagrams** to define the flow of the System Functions.

User also may use:

- Regular UML Class Diagram
- Regular UML Activity Diagram
- SysML Block Definition Diagram
- SysML Activity Diagram
- Any other MagicDraw diagram

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
System Function	<<SystemFunction>> [Activity]	
System Data Flow	<<SystemDataFlow>> [ControlFlow, ObjectFlow]	N/A
System Data Repository	<<SystemDataRepository>> [DataStoreNode]	
System Function Action	<<SystemFunctionAction>> [CallBehaviorAction]	N/A
Operational Activity	See OV-5 data element definition	
System	See SV-1 data element definition	
Systems Node	See SV-1 data element definition	
System Data Exchange	See SV-6 data element definition	
System Data Element	See SV-6 data element definition	

SystemFunction

Description

A data transform that supports the automation of activities or information elements exchange. The system functions documented in SV-4 may be identified using the Service Component Reference Model (SRM), or some other system function taxonomy, and correlated to SV-1 and SV-2 systems.

Extends UML Activity.

Attributes

implements: OperationalActivity [*]	A set of Operational Activities that are implemented by System Function
performedBy: System [*]	Systems to which System Function is allocated
allocatedAt: SystemsNode [*]	Systems Nodes at which System Function is allocated
produces: SystemDataExchange [*]	System data produced by System Function
consumes: SystemDataExchange [*]	System data consumed by System Function
subfunction: SystemFunction [*]	Sub-functions into which the System Function is decomposed
parent: SystemFunction [0..1]	System Function that is decomposed
performanceParameterSet: PerformanceParameterSet [*]	Performance Parameter Set specified for System Function
measurements: PerformanceMeasurementSet [*]	A list of Performance Measured Sets with measured Performance Parameter Set values

SystemDataFlow

Description

The SystemDataFlow of a system describes the data transferred between system Functions, external system data sources, or internal system data repositories.

Extends UML ControlFlow, UML ObjectFlow.

Attributes

type: SystemDataElement [*]	A list of System Data Elements that need to be exchanged

SystemDataRepository

Description

System data store.

Extends UML DataStoreNode.

SystemFunctionAction

Call action that invokes System Function directly.

Extends UML CallBehaviorAction.

Profile diagram

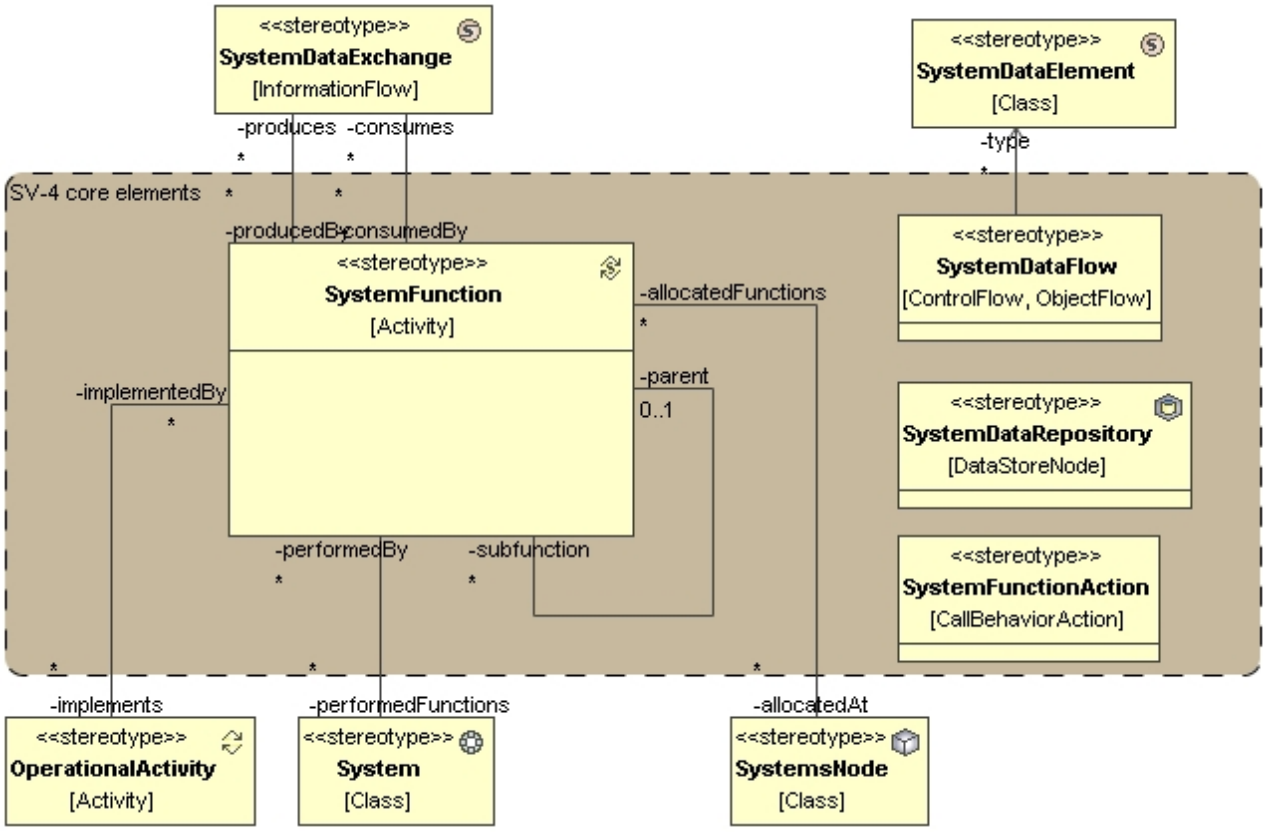

Figure 9 -- SV-4 profile diagram

Creating the product

Creation of SV-4 is very similar to OV-5.

We do recommend starting with the SV-4 System Function Flow, because MagicDraw will be able to create SV-4 System Function Hierarchy automatically from the data in the SV-4 flow definition.

SV-4 System Function Hierarchy

Use SV-4 specific toolbar to create hierarchies of the **System Functions**. Use **Composition (Associaton)** as main path between **System Functions**. As **UML Activity** is a **UML Classifier**, it can be used in the structure diagrams.

SV-4 System Function Flow

This is very similar to **UML Activity diagram**. Drawing **System Functions** will result in creation of **Call Behavior Actions** with **System Function** assigned as behavior.

Relationship to other products

Operational Activities from **OV-5** are implemented by System Functions (property *implements*).

Systems from **SV-1/SV-2** perform the System Functions (property *performedBy*).

Systems Nodes from **SV-1/SV-2** group or host the System Functions (property *allocatedAt*).

System Functions produce and consume the System Data Exchange from **SV-6** (properties *produces* and *consumes*).

System Data Flow has the System Data Element from **SV-6** as type (property *type*).

Sample

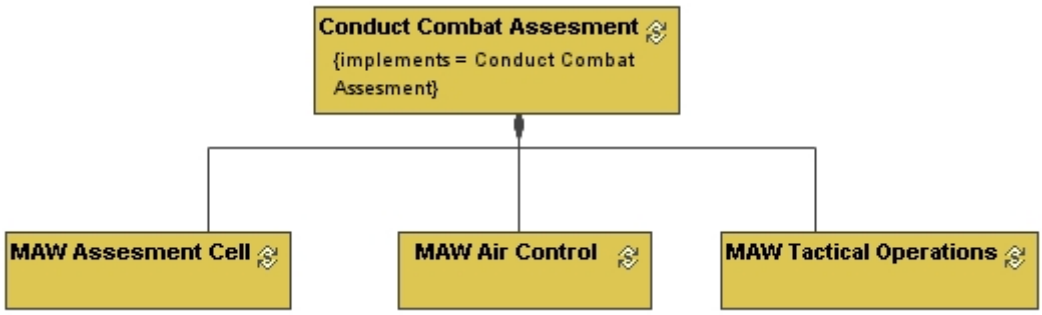

Figure 10 -- SV-4 sample (hierarchy)

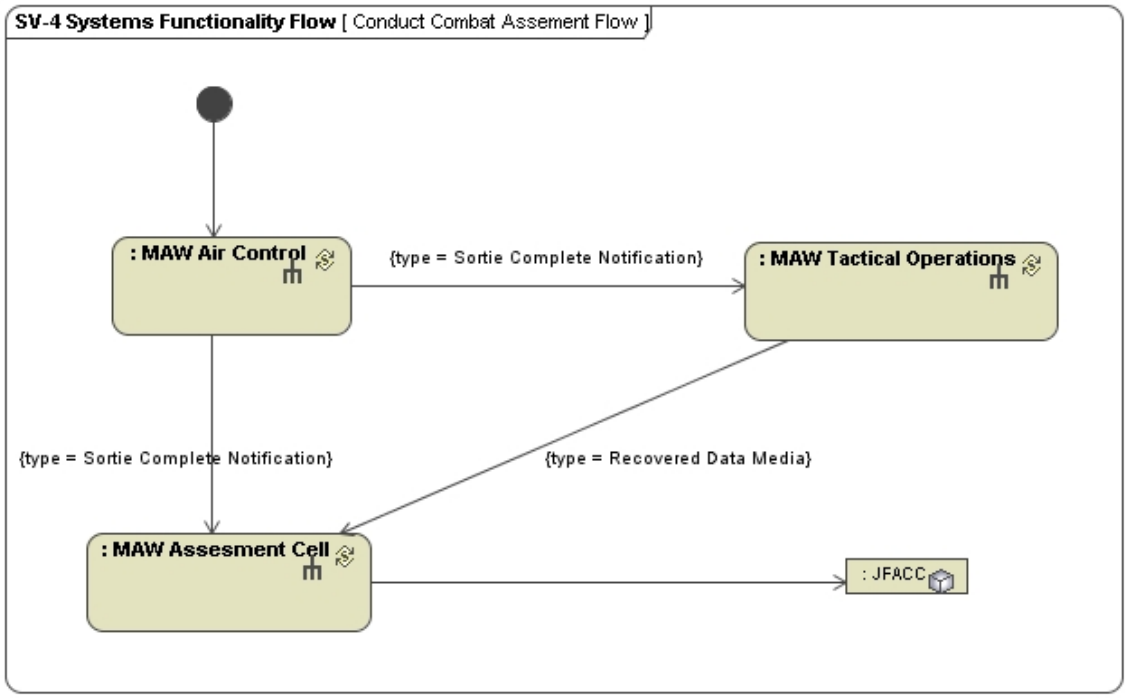

Figure 11 -- SV-4 sample (flow)

SV-5. Operational Activity to Systems Functions Traceability Matrix

Description

"Operational Activity to Systems Function Traceability Matrix is a specification of the relationships between the set of operational activities applicable to architecture and the set of system functions applicable to that architecture. The Framework uses the terms *activity* in the OVs and *function* in the SVs to refer to essentially the same kind of thing—both activities and functions are tasks that are performed, accept inputs, and develop outputs." [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

SV-5 is a **matrix** based on **dependency matrix** in MagicDraw.

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Operational Activity	See OV-5 data element definition	
System Function	See SV-4 data element definition	

Profile diagram

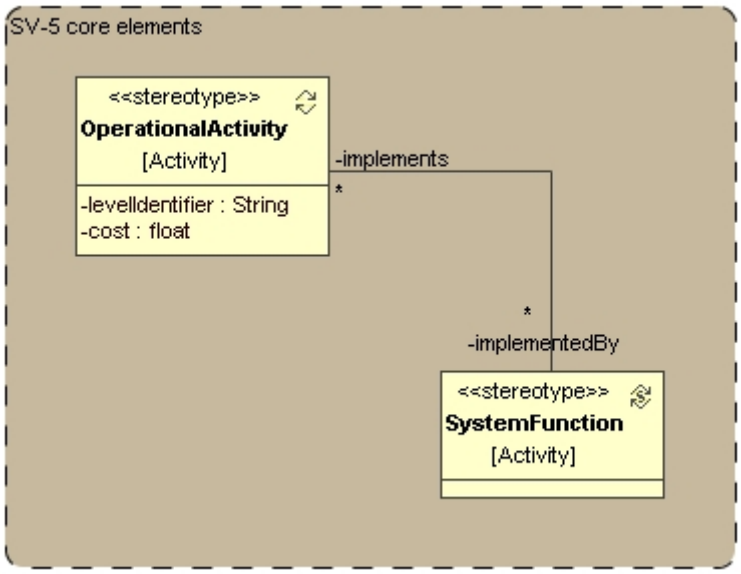

Figure 12 -- SV-5 profile diagram

Creating the product

When user will create SV-5 matrix in the row scope he will be provided with a scope selection for the System Functions. In the column scope he will be provided with a scope selection for the Operational Activities. MagicDraw will find all System Functions and Operational Activities in the given scope and put them as column and row elements. The matrix cells will be filled according to relations between System Functions and Operational Activities (calculated from *implementedBy* property for Operational Activity and *implements* property for System Function).

Relationships to other products

Operational Activities from **OV-5** are used in SV-5.

System Functions from **SV-4** are used in SV-5.

Sample

	Conduct Battle D...	Conduct Combat ...	Conduct Joint For...	Conduct Munition...	Develop Targets ...	Establish Guidanc...	Manage Targets [...	Prioritize Targets ...	Publish Air Taskin...	Recommend Rest...	Request Target ...
SV-4 [Systems View]		2									
Conduct Combat Assesme...		X									
MAW Air Control [Systems ...											
MAW Assesment Cell [Syst...											
MAW Tactical Operations [...											

Figure 13 -- SV-5 sample

SV-6. Systems Data Exchange Matrix

Description

“The Systems Data Exchange Matrix specifies the characteristics of the system data exchanged between systems. This product focuses on automated information exchanges (from OV-3) that are implemented in systems. Non-automated information exchanges, such as verbal orders, are captured in the OV products only. System data exchanges express the relationship across the three basic architecture data elements of an SV (systems, system functions, and system data flows) and focus on the specific aspects of the system data flow and the system data content.

The focus of SV-6 is on how the system data exchange is implemented, in system-specific details covering periodicity, timeliness, throughput, size, information assurance, and security characteristics of the exchange. In addition, the system data elements, their format and media type, accuracy, units of measurement, and system data standard are also described in the matrix.“ [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

SV-6 is a **document**, which is implemented as MagicDraw **report**, using model information mainly from SV-1.

Note. SV-6 is a product similar to OV-3. Therefore OV-3 implementation concepts will be reused.

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
System Data Exchange	<<SystemDataExchange>> [InformationFlow]	
System Data Element	<<SystemDataElement>> [Class]	
Information Exchange	See OV-3 data element definition	
Information Element	See OV-3 data element definitions	
Interface	See SV-1 data element definition	
System Function	See SV-4 data element definition	
Standard	See TV-1 data element definition	

SystemDataExchange

Description

An act of exchanging system data between two distinct Systems and the characteristics of the act, including System Data Element that needs to be exchanged and the attributes associated with System Data Element (e.g., content) as well as attributes associated with System Data Exchange such as timeliness.

Extends UML Information Flow.

Attributes

identifier: String	Identifier of System Data Exchange – usually based on the relevant Needline, Interface, and Information Exchange
consumedBy: SystemFunction [*]	A list of System Functions that consume system data
producedBy: SystemFunction [*]	A list of System Functions that produce system data
transactionType: String	Type of exchange
interoperabilityLevelAchievable: String	Level of Information Systems Interoperability (LISI) achieved or achievable through the exchange
criticality: String	The criticality assessment of the information being exchanged in relationship to the mission being performed
periodicity: String	Frequency of System Data Exchange transmission – may be expressed in terms of worst case or average frequency
timeliness: String	Allowable time of delay this system data can tolerate and still be relevant to the receiving system
throughput: String	Bits or bytes per time period – may be expressed in terms of maximum or average throughput required
size: String	Size of system data
IAAccessControl: String	The class of mechanisms used to ensure only those authorized that can access a specific System Data Element
IAAvailability: String	The relative level of effort required to be expended to ensure that the system data can be accessed
IAConfidentiality: String	The kind of protection required for system data to prevent unintended disclosure
IADisseminationControl: String	The kind of restrictions on receivers of system data based on sensitivity of system data
IAIntegrity: String	The kind of requirements for checks that the content of the system data element has not been altered
IANonRepudiationConsumer: String	The requirements for unassailable knowledge that the system data sent was consumed by the intended recipient
IANonRepudiationProducer: String	The requirements for unassailable knowledge that the system data received was produced by the stated source
protectionTypeName: String	The name for the type of protection

protectionDurationCode: String	The code that represents how long the system data must be safeguarded
protectionSuspenseCalendarDate: String	The calendar date on which the designated level of safeguarding discontinues for a specific system data element
classification: String	Classification code for the System Data Element
classificationCaveat: String	A set of restrictions on system data of a specific classification. Supplements a security classification with system data on access, dissemination, and other types of restrictions
releasability: String	The code that represents the kind of controls required for further dissemination of system data
securityStandard: Standard	System Data Exchange security standard
automates: InformationExchange [*]	Information Exchanges automated by System Data Exchange
dataElement: SystemDataElement [1]	System Data Element that is exchanged via System Data Exchange
performanceParameterSet: PerformanceParameterSet [*]	Performance Parameter Set specified for System Data Exchange
measurements: PerformanceMeasurementSet [*]	A list of Performance Measured Sets with measured Performance Parameter Set values

SystemDataElement

Description

The architecture data element or type that stores data from the architecture domain (i.e., it has a value) that is produced or consumed by a System Function and that has System Data Exchange attributes.

Extends UML Class

Attributes

identifier: String	Identifier of System Data Element – based on the relevant System Data Flow. System Data Element identifier correlates with OV-3 Information Element
content: String	The system data that is carried by the exchange
formatType: String	Application level format with parameters and options used, or other relevant protocol
mediaType: String	Type of media
accuracy: String	Description of the degree to which the system data conforms to actual fact as required by the System or System Function
unitOfMeasurement: String	Units used for system data
systemDataStandard: Standard	System Data Standard such as DoD XML Registry
dataExchange: SystemDataExchange [*]	A list of System Data Exchanges that exchange System Data Element
implements: InformationElement [0..1]	Information Element that is implemented by System Data Element
performanceParameterSet: PerformanceParameterSet [*]	Performance Parameter Set specified for System Data Element
measurements: PerformanceMeasurementSet [*]	A list of Performance Measured Sets with measured Performance Parameter Set values

Profile diagram

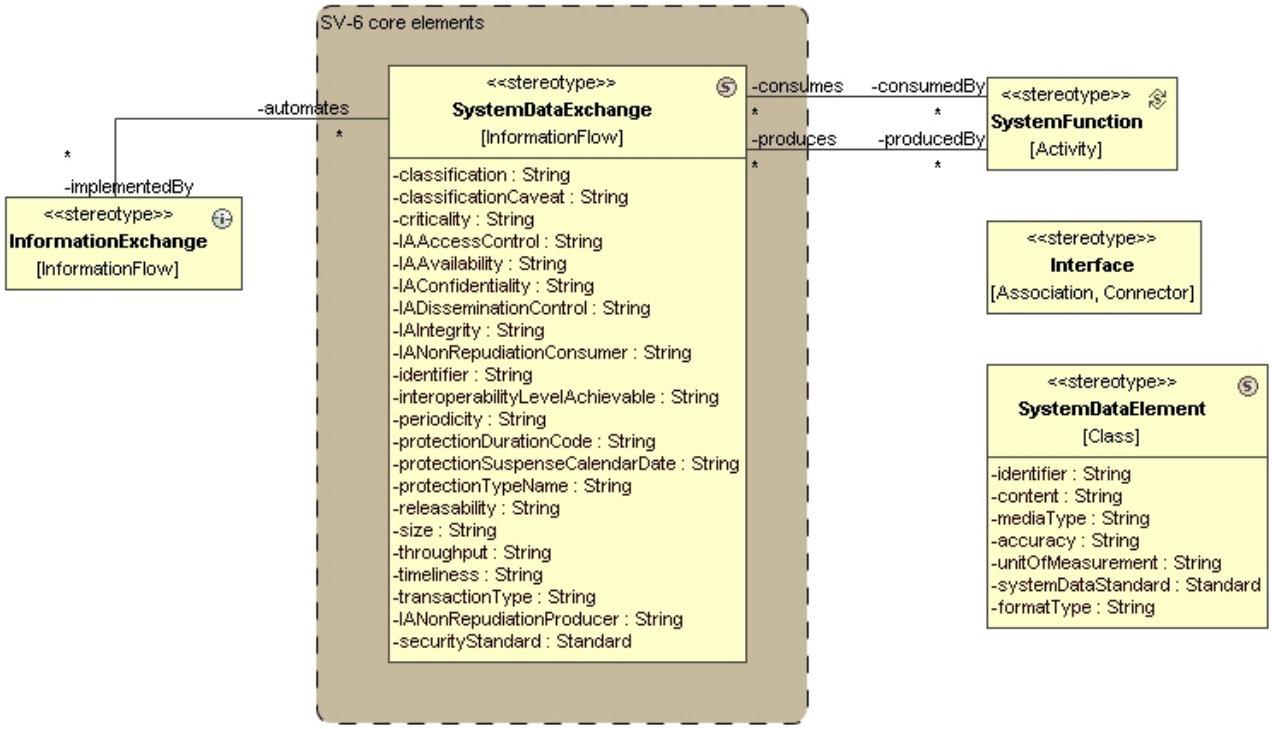

Figure 14 -- SV-6 profile diagram

Creating the Product

OV-2 and SV-1 has to be created upfront in order to generate the SV-6.

If OV-2 and SV-1 are created according to MagicDraw's suggested way – Information Exchanges, System Data Exchanges and System Data elements are fully described, there is nothing more to model in order to get the SV-6. Go to MagicDraw report engine and run the generation of the SV-6.

Creating the product

SV-1 has to be created upfront in order to generate the SV-6.

If SV-2 is created according to MagicDraw's suggested way – Interfaces, System Data Exchanges and System Data Elements are fully described, there is nothing more to model in order to get the SV-6. SV-6 just gathers information about System Data Exchanges, System Data Elements and attributes of System Data Exchange and generates the document. Report may be generated for whole model or for selected scope.

Report template

In System Data Exchange Matrix report information about System Data Exchange attributes are grouped according to System Data Exchange attributes types:

- *System Data Exchange Associations* section associates System Data Exchange with producing and consuming Systems and System Functions.
- *System Data Element Description* section provides information about specific attributes of System Data Element: name, identifier, content, format type, media type, accuracy, unit of measurement, and data standard.
- *Nature of Transaction* section includes the list of System Data Exchange attributes that specifies nature of transaction: transaction type, interoperability level achieved, and criticality.
- *Performance Attributes* section defines System Data Exchange performance attributes: periodicity, timeliness, throughput, and size.
- *Information Assurance* section lists System Data Exchange information assurance attributes: access control, availability, confidentiality, dissemination control, integrity, non-repudiation producer and non-repudiation consumer.
- *Security* section presents System Data Exchange security attributes: protection (type name, duration, date), classification, classification caveat, releasibility, and security standard.

In each section items are sorted according to System Data Exchange identifier.

User can customize report by selecting System Data Exchange attributes sections that should be included to the report. Inside each section the list of System Data Exchange attributes can also be customized.

Relationship to other products

Information Exchanges from **OV-2** are used in System Data Exchanges (property *automates*)

Interfaces from **SV-1** are used in SV-6 (property *interface*).

System Data Elements from **SV-11** are used in SV-6 (property *DataElement*)

System Data Exchanges are consumed or produced by System Function (properties *consumedBy* and *producedBy*)

System Data Exchange may use a Standard from TV (property *securityStandard*)

Sample

2. System Data Exchange Associations

Interface Name	System Data Exchange Name and Identifier	Sending System Name	Sending System function Name	Receiving System Name	Receiving System function Name
NA	name flow for BDA Reports (Imagery/Text) identifier NA	DJFLCC **participant** NA		JFC **participant** NA	
NA	name flow for BDA Reports (Imagery/Text) identifier NA	JFACC **participant** NA		WOC **participant** NA	
NA	name flow for BDA Reports (Imagery/Text) identifier	JFC **participant** NA		JFACC **participant** NA	

Figure 15 -- SV-6 sample

SV-7. Systems Performance Parameters Matrix

Description

"The Systems Performance Parameters Matrix product specifies the quantitative characteristics of systems and system hardware/software items, their interfaces (system data carried by the interface as well as communications link details that implement the interface), and their functions. It specifies the current performance parameters of each system, interface, or system function, and the expected or required performance parameters at specified times in the future.

In more detail, SV-7 builds on SV-1, SV-2, SV-4, and SV-6 by specifying performance parameters for systems and system hardware/software items and their interfaces (defined in SV-1), communications details (defined in SV-2), their functions (defined in SV-4), and their system data exchanges (defined in SV-6)." [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

SV-7 in a **table** that is implemented as special GUI in MagicDraw. User will be provided a special table-like GUI for filling out the SV-7 data. As in MagicDraw everything is based on the underlying model, creating SV-7 will also create the UML model.

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Performance Parameter Set	<<PerformanceParameterSet>> [Class]	
Performance Parameter Type	<<PerformanceParameterType>> [Property]	
Performance Measurement Set	<<PerformanceMeasurementSet>> [InstanceSpecification]	
Performance Measurement	<<PerformanceMeasurement>> [Slot]	
Time Period	See the Time definitions	

PerformanceParameterSet

Description

The set of technical performance characteristics. Performance Parameter Set can be specified for System, Hardware or Software Item, System Function, Communications Link, Communications Systems, Communications Network, LAN, WAN, MAN, System Data Element and System Data Exchange.

Performance Parameter Set has Performance Parameter Types as child elements.

Extends UML Class.

Attributes

measuredSystems: Standards/ PerformanceSubject [*]	Performance subjects for which Performance Parameter Set is specified. Performance Subjects are System, Hardware or Software Item, System Function, Communications Link, Communications Systems, Communications Network, LAN, WAN, MAN, System Data Element and System Data Exchange.

PerformanceParameterType

Description

Thi is the technical performance characteristic measured. Grouped by Performance Parameter Set.

Extends UML Property.

Attributes

unitOfMeasure: String	Unit of performance parameter measurement
thresholdValue: String	Value of Performance Parameter Type that is acceptable at the indicated time
objectiveValue: String	Desired operational goal, beyond which any gain in utility does not warrant additional expenditure

PerformanceMeasurementSet

Description

This is an instance of Performance Parameter Set that contains actual measured values of Performance Parameter Set. Measured values are taken for specified Time Period. Performance Measurement elements are child elements of Performance Measurement Set. Holds references to the measured subject and time at which the measurement was performed.

Extends UML InstanceSpecification.

Attributes

timePeriod: TimePeriod [1]	Time Period when the measurements were/will be performed
measuredSystem: Standards/Performance Subject [1]	System view element (System, Hardware or Software Item, System Function, Communications System, Communications Link, Communications Network, LAN, WAN, MAN, System Data Element, System Data Exchange) for which performance parameters values are measured

PerformanceMeasurement

Description

Actual measured value of Performance Parameter Type.

Extends UML Slot.

Profile Diagram

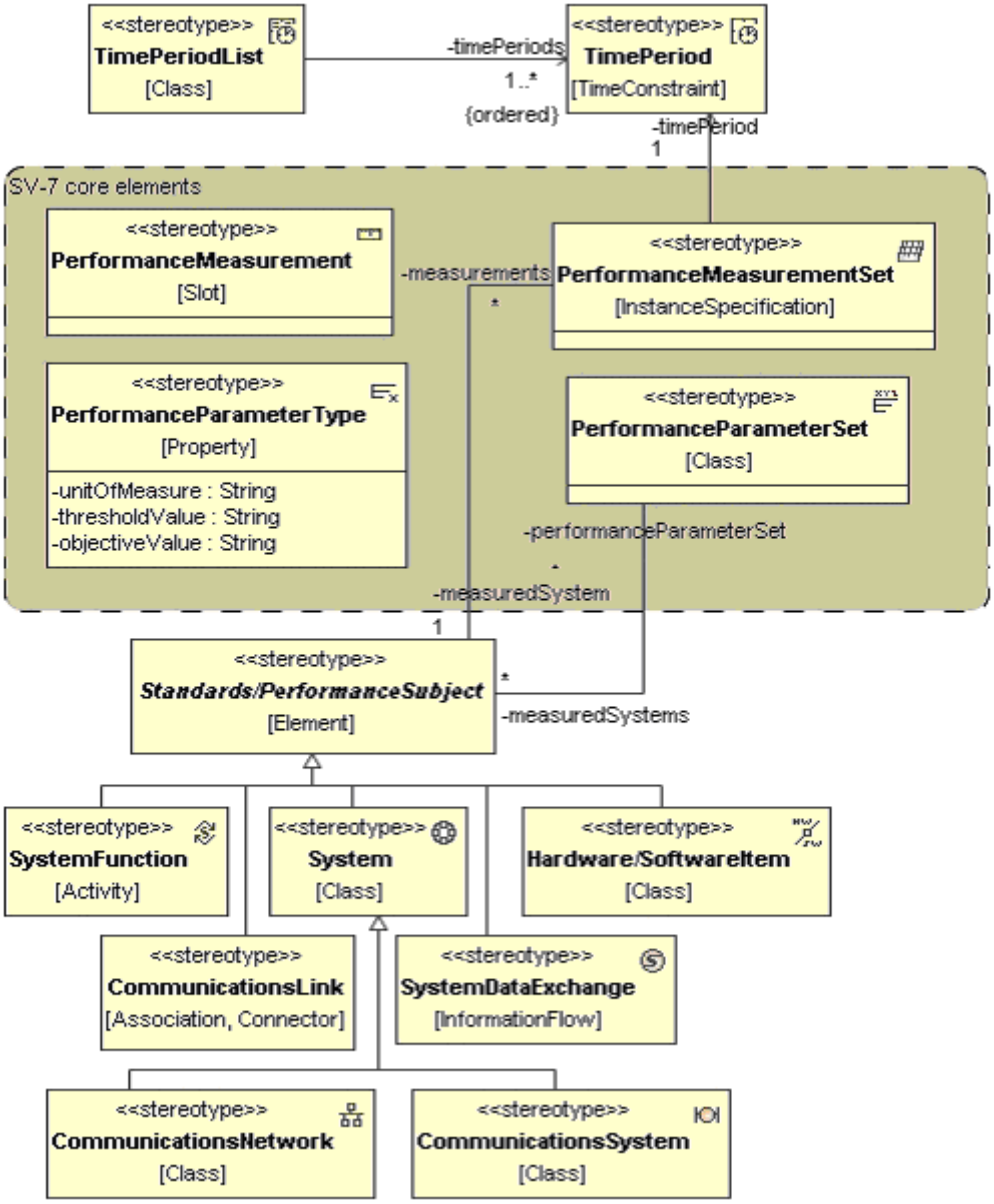

Figure 16 -- SV-7 profile diagram

Creating the Performance Parameter Sets

Performance Parameter Sets can be accessed from DoDAF main menu. Straight from Performance Parameter Sets dialog user can create Performance Parameter Sets and assign them for Performance Subjects. Made changes will be automatically applied for Performance Subjects specifications.

Performance Parameters Types can be defined on Performance Parameter Set specification. It can be invoked straight from Performance Parameter Sets dialog.

For each Performance Parameter Type objective value, threshold value and unit of measure can be specified.

Creating the product

When user will create SV-7 in the row scope he will have to select Standards/Performance Subjects (System, Hardware or Software Item, System Function, Communications System, Communications Link, Communications Network, LAN, WAN, MAN, System Data Element, System Data Exchange) for which

Performance Parameter Types values will be measured. In the column scope he will have to select Time Periods when the measurements will be performed. After that, user will be provided with a special GUI-based table, where rows will be Standards/Performance Subjects (Systems, System Functions, etc) and Performance Parameter Types. Columns will be collected from the selected Time Periods. User will fill in the cells, filling the measurement result for a Performance Measurement Type at a specific Time Period.

Despite this is a product that does not map to any diagram; DoDAF model will be also created in order to keep integrity with the rest of the model.

Relationships to other products

Systems from **SV-1/SV-2** will be used in SV-7

Hardware or Software Items from **SV-1/SV-2** will be used in SV-7

System Functions from **SV-4** will be used in SV-7

Time Periods from **Time Definitions** will be used in SV-7

Sample

	Performance Range		
	1 month	2 months	3 months
AFATDS [Systems View::SV-1]			
-Response time [Systems V...	2	2	2
-Operations per second [S...	200	200	200
APS [Systems View::SV-1]			
-Response time [Systems V...	2	3	2
-Operations per second [S...	342	350	360
ASAS [Systems View::SV-1]			
-Response time [Systems V...	5	6	7
-Operations per second [S...	222	234	345
CIGSS [Systems View::SV-1]			
-Response time [Systems V...	1	1	1
-Operations per second [S...	170	180	190
ADSI [Systems View::SV-1]			
-Response time [Systems V...	11	12	14
-Operations per second [S...	145	150	160

Figure 17 -- SV-7 sample

Sample model

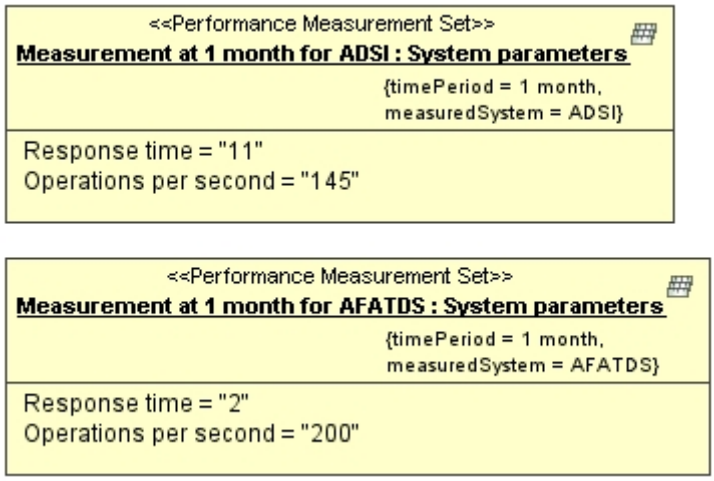

Figure 18 -- SV-7 sample model

SV-8. Systems Evolution Description

Description

"The Systems Evolution Description captures evolution plans that describe how the system, or the architecture in which the system is embedded, will evolve over a lengthy period of time. Generally, the timeline milestones are critical for a successful understanding of the evolution timeline.

SV-8, when linked together with other evolution products such as SV-9 and TV-2, provides a clear definition of how the architecture and its systems are expected to evolve over time. In this manner, the product can be used as an architecture evolution project plan or transition plan." [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

SV-8 is a **diagram** that is based on a **UML State Machine Diagram** in MagicDraw. There is no standard way to do what, but we find semantic similarities between milestone and state – you can call a milestone a certain state of the subject system.

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Milestone	<<Milestone>> [State]	◇
System	See SV-1 data element definition	
Time Period	See the Time definitions	

Milestone

Description

A scheduling event that signifies the completion of a major deliverable or a set of related deliverables.

Extends UML State.

Attributes

version: String	Milestone version
timePeriod: TimePeriod [1]	Time period for milestone
systemGroup: System [*]	Systems required to complete a milestone

Profile diagram

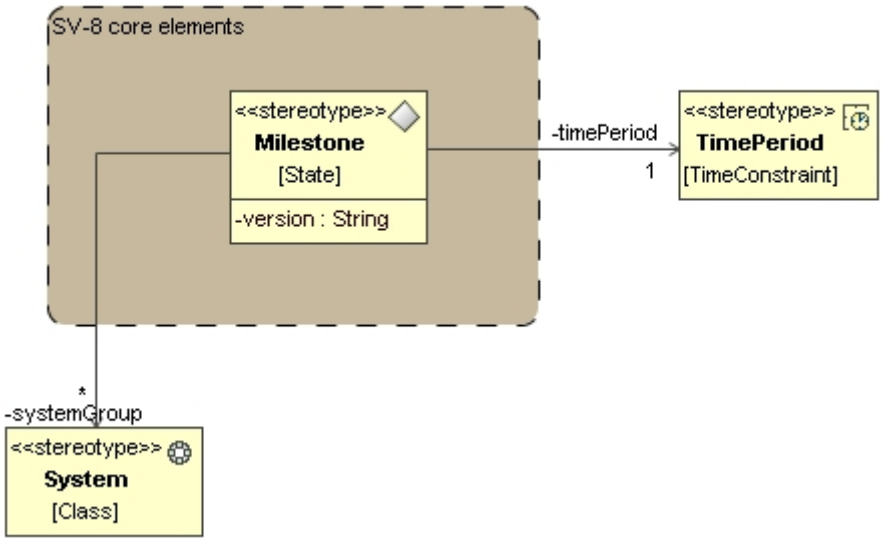

Figure 19 -- SV-8 profile diagram

Creating the product

Creating is basically equal to creation of a regular UML statemachine diagram. Create a set of connected milestones. Then assign Time Periods and Systems required for each Milestone.

Relationship to other products

Systems from **SV-1/SV-2** are used in Milestones.

Time Periods are used in Milestones.

Sample

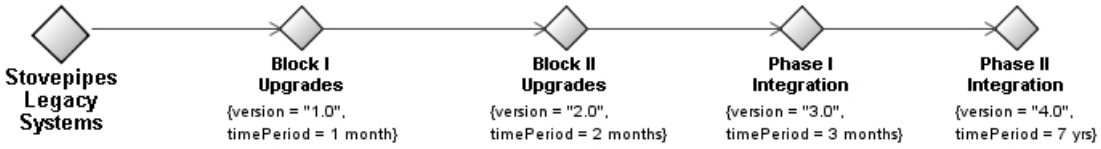

Figure 20 -- SV-8 sample

SV-9. Systems Technology Forecast

Description

“The Systems Technology Forecast defines the underlying current and expected supporting technologies. It is not expected to include predictions of technologies as with a crystal ball. Expected supporting technologies are those that can be reasonably forecast given the current state of technology and expected improvements. New technologies should be tied to specific time periods, which can correlate against the time periods used in SV-8 milestones.” [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

SV-9 in a **table** that is implemented as special GUI in MagicDraw. User will be provided a special table-like GUI for filling out the SV-9 data. As in MagicDraw everything is based on the underlying model, creating SV-9 will also create the UML model.

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Technology	<<Technology>> [Class]	
Technology Forecast Profile	<<TechnologyForecastProfile>> [Package]	
Timed Technology Forecast	<<TimedTechnologyForecast>> [Usage]	N/A
Time Period	See the Time definitions	
Standard	See TV-1 data element definition	
Standards Profile	See TV-1 data element definition	
Reference Model	See TV-1 data element definition	
Timed Standards Forecast	See TV-2 data element definition	

Technology

Description

Technology.

Extends UML Class

TechnologyForecastProfile

Description

Describes a set of Technologies that can emerge in specific Time Period. The list of Technologies is coordinated with architecture transition plans. It will be a root package for adding the SV-9 data. As an example a system might be written initially in C++, then Java, and then an emerging language that is predicted to meet the needs of the architecture. The forecast used to show both the trusted and available technology for an architecture, while showing technologies in the future that can be applied for various reasons.

Extends UML Package.

Attributes

timePeriod: TimePeriod [1]	Time Period for which Technology Forecast Profile is made
basedOn: ReferenceModel [*]	Reference Models on which Technology Forecast Profile is based
basedOn: StandardsProfile [*]	Standards Profiles on which Technology Forecast Profile is based

TimedTechnologyForecast

Description

Relationship used to relate prediction about availability of emerging technology to System View element (System, Hardware or Software Item, System Function, Communications System, Communications Link, Communications Network, LAN, WAN, MAN, System Data Element, System Data Exchange) and Time Period for which forecast is made.

Basically it will be invisible to the user, as user will work with special UI, and the Timed Technology Forecast will be created in the underlying model.

Extends UML Usage.

Attributes

discussion: String	Textual notes regarding Technology status, likely commercial market acceptance, and risk assessment of adopting the technology forecast
timePeriod: TimePeriod [1]	Time Period for which technology forecast is made
requiredBy: TimedStandardsForecast [*]	A list of Timed Standards Forecasts that require Timed Technology Forecast
retiredStandard: Standard [*]	A list of current Standards that may be retired of phased out

Profile Diagram

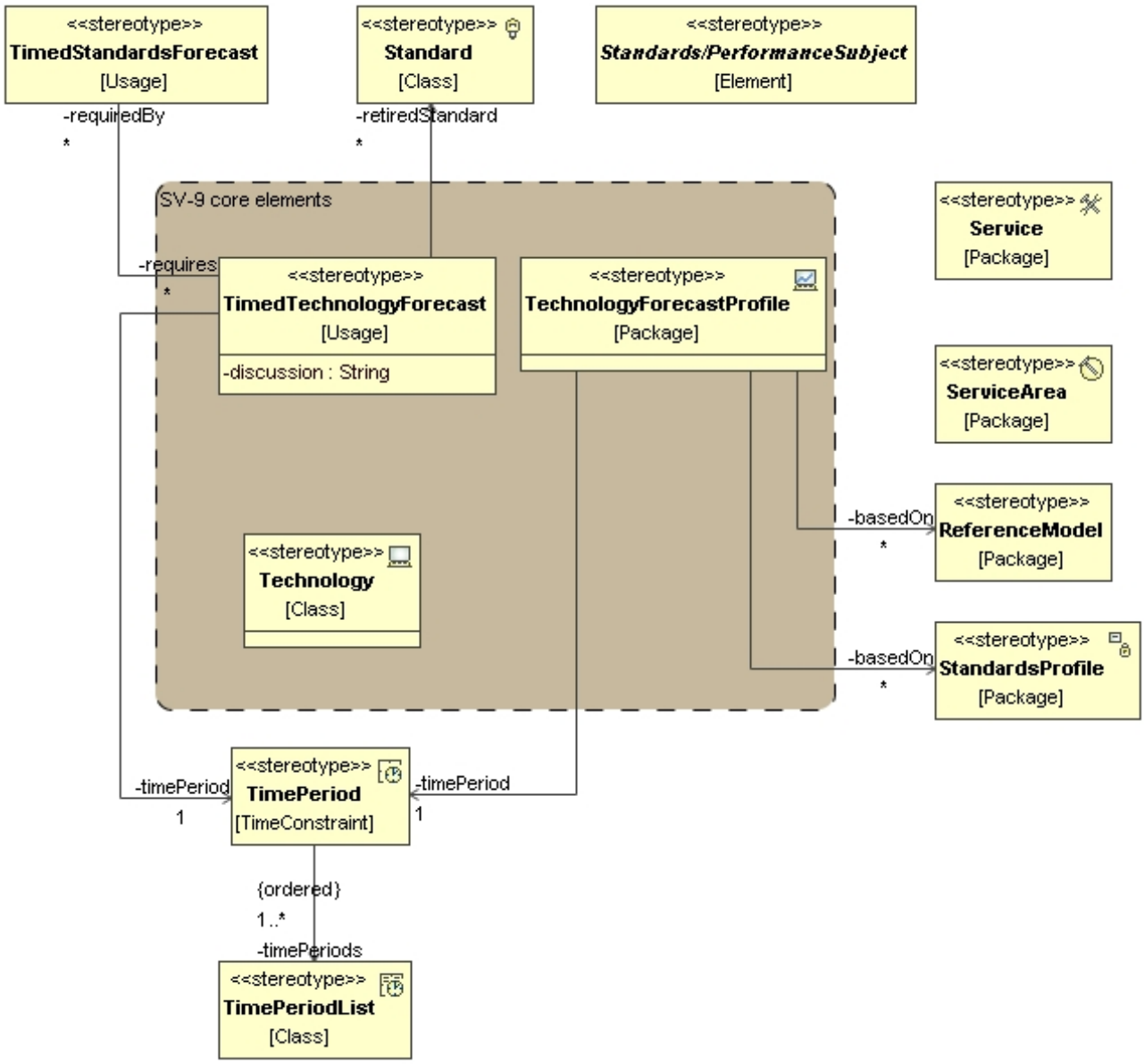

Figure 21 -- SV-9 profile diagram

Creating the product

When user will create SV-9 in the row scope he will have to select Services (or Systems) potentially impacted by technology forecast. In the column scope he will have to select Time Periods for which forecast will be made. After that, user will be provided with a special GUI-based table, where rows will be Services (or Systems). Columns will be collected from the selected Time Periods. User will fill in the cells, creating the Technology elements that are used at a specific Time Period.

Despite this is a product that does not map to any diagram; DoDAF model will be also created in order to keep integrity with the rest of the model.

Relationship to other products

Systems from **SV-1**/**SV-2** may be used in SV-9 in order to show the applied technologies.

Sample

	Technology Forecasts		
	1 month	2 months	3 months
Application Software [Systems...			
Support Application [Syste...		Java [Systems View	
Application Platform [Systems ...			
Operating System [System...		Windows [Systems \	Linux [Systems View
Data Management [System...	Oracle 9i [Systems '		

Figure 22 -- SV-9 sample

Sample model

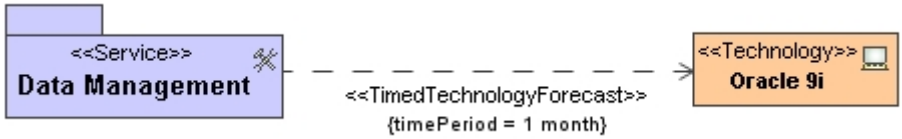

Figure 23 -- SV-9 sample model

SV-10a. Systems Rules Model

Description

"Systems rules are constraints on an architecture, on a system(s), or system hardware/software item(s), and/or on a system function(s). While other SV products (e.g., SV-1, SV-2, SV-4, SV-11) describe the static structure of the Systems View (i.e., what the systems can do), they do not describe, for the most part, what the systems *must* do, or what it *cannot* do." [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

SV-10a is similar to OV-6a.

System Rule is implemented as **UML Constraint** in MagicDraw, so every DoDAF element in MagicDraw project can have an Operational Rule assigned.

SV-10a is a **document**, which is implemented as MagicDraw **report**, using model information mainly from OV products.

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
System Rule	<<SystemRule>> [Constraint]	{s}

System Rule

Description

Statement that defines or constrains some aspect of the mission, or the architecture. In contrast to Operational Rule, System Rule focuses on constraints imposed by some aspect of operational performance requirements that translate into system performance requirements. The rule is described in low detail and is primarily a human readable rule. The rule focuses on aspects of systems design and implementation.

Extends UML Constraint.

Attributes

type: RuleType [1] = Structural Assertion	System Rule type. Possible values are defined in the RuleType enumeration: Structural Assertion Action Assertion Derivation

Profile Diagram

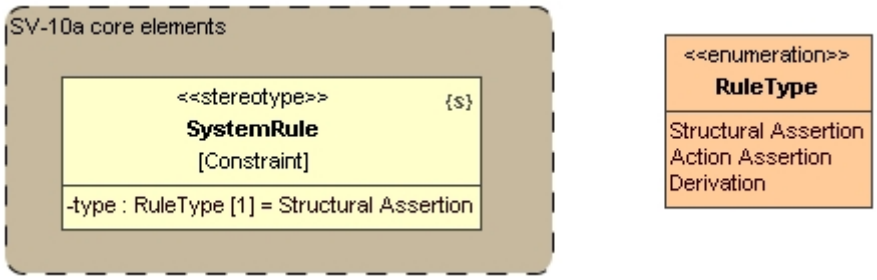

Figure 24 -- SV-10a profile diagram

Creating the product

In general **SV-10a** is just a collection of **Operational Rules** (**UML constraints**) from the model compiled in a document. User may generate report for whole project or for selected scope.

A System Rule can be applied to **any** element in the DoDAF model.

Report Template

In Systems Rules Model report System Rules are grouped according to type:

- *Structural assertions* are added to “Structural Assertions” section of the report.
- *Action assertions* are added to “Action Assertions” section of the report.

- *Derivations* are added to “Derivations” section of the report.

Inside each group System Rules are listed in numbered manner and sorted according to System Rule name. Below each System Rule name constraint specification and information about constrained elements is provided.

User can customize report by selecting the type of System Rules, for which report should be generated. The list of constrained elements may also be included optionally.

Relationship to other products

The SV-10a is a collection of System Rules from the rest of the model.

Sample

1.Structural Assertions

1. Using Link-33

All systems using the Link-33 communications terminals that receive Message A4, Request for Active Missile Tracks, must respond within 1second with a Message A6, Active Missile Tracks Update

Figure 25 -- SV-10a sample

SV-10b. Systems State Transition Description

Description

“The Systems State Transition Description is a graphical method of describing a system (or system function) response to various events by changing its state. The diagram basically represents the sets of events to which the systems in the architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action. SV-10b can be used to describe the detailed sequencing of system functions described in SV-4.” [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

SV-10b is a diagram that is based on **UML State Machine diagram**.

UML Profile for DoDAF

No additional mapping to UML is needed for producing SV-10b.

Creating the product

This is implemented as a UML State Machine diagram.

Relationship to other products

Systems, Systems Nodes from **SV-1** may have state machines as UML property O*wned Behavior*.

System Functions from **SV-4** may have state machines as UML property O*wned Behavior*.

Sample

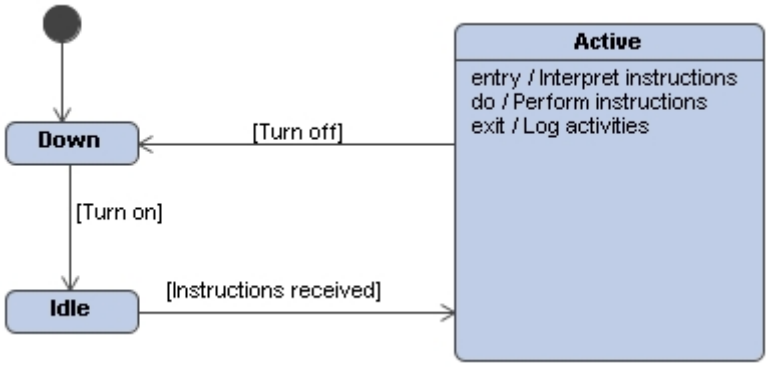

Figure 26 -- SV-10b sample

SV-10c. Systems Event-Trace Description

Description

"The Systems Event-Trace Description provides a time-ordered examination of the system data elements exchanged between participating systems (external and internal), system functions, or human roles as a result of a particular scenario. Each event-trace diagram should have an accompanying description that defines the particular scenario or situation. SV-10c in the Systems View may reflect system-specific aspects or refinements of critical sequences of events described in the Operational View." [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

SV-10c is a **diagram** that is based on **UML Sequence Diagram**.

UML Profile for DoDAF

No additional mapping to UML is needed for producing SV-10c.

Creating the product

This is a regular UML sequence diagram. You may drag in the objects from other DoDAF product from the data browser to this diagram in order to create lifelines.

Relationship with other products

SV elements may be used as types for the interaction attributes that will be set as *represents* property for the lifelines in SV-10c.

Sample

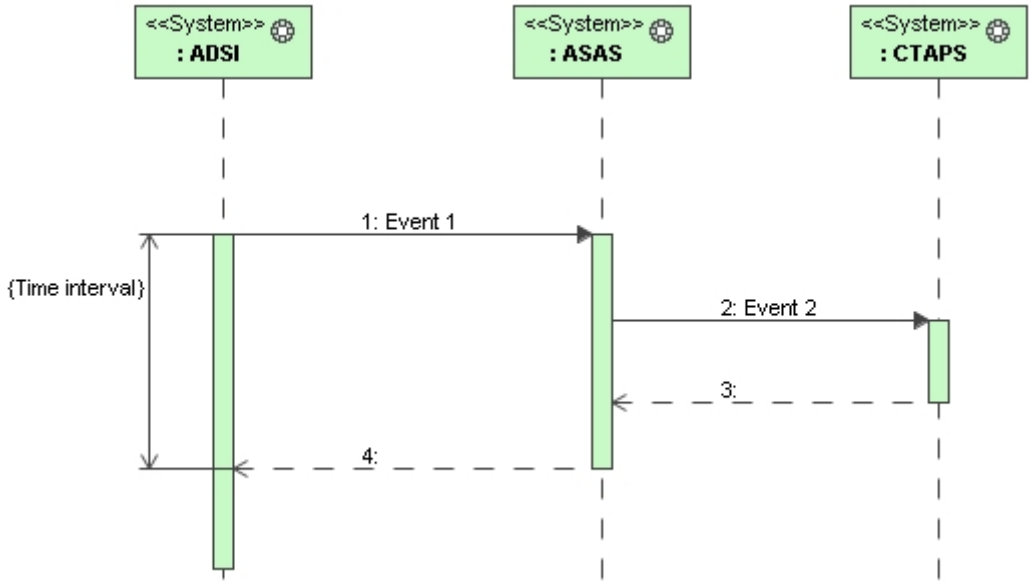

Figure 27 -- SV-10c sample

SV-11. Physical Schema

Description

"The Physical Schema product is one of the architecture products closest to actual system design in the Framework. The product defines the structure of the various kinds of system data that are utilized by the systems in the architecture.

SV-11 is an implementation-oriented data model that is used in the Systems View to describe how the information requirements represented in Logical Data Model (OV-7) are actually implemented. Entities represent (a) system data flows in SV-4, (b) system data elements specified in SV-6, (c) triggering events in SV-10b, and/or (d) events in SV-10c." [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

SV-11 is a **diagram** that is based **UML Class Diagram**.

User also may use:

- Regular UML Class Diagram
- SysML Block Definition Diagram
- Any other MagicDraw diagram

UML Profile for DoDAF

Information Element	See OV-3 data element definitions
System Data Element	See SV-6 data element definition

Profile Diagram

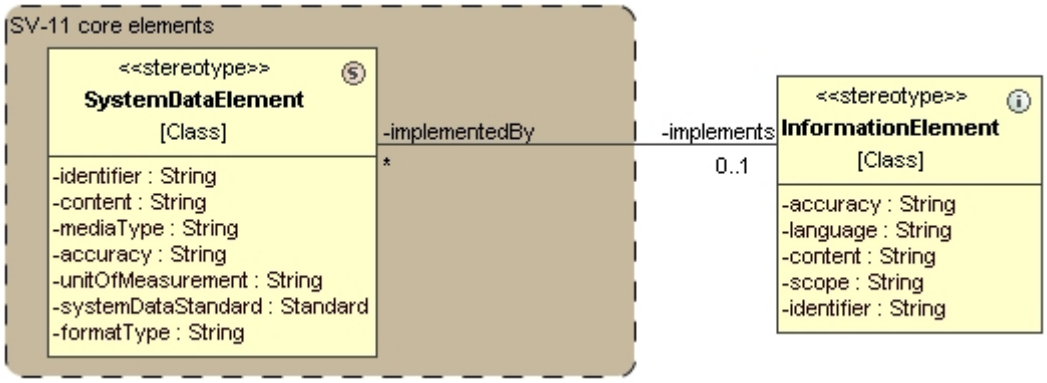

Figure 28 -- SV-11 profile diagram

Creating the product

SV-11 is a regular class diagram with **System Data Elements** used instead of a **UML Class**.

It is likely that a lot of **System Data Elements** will be created to this point, so user can just drag those elements to the diagram from the browser tree.

Relationship with other products

System Data Elements defined in **SV-6** are reused in SV-11.

System Data Element may implement Information Element from **OV-6**.

Sample

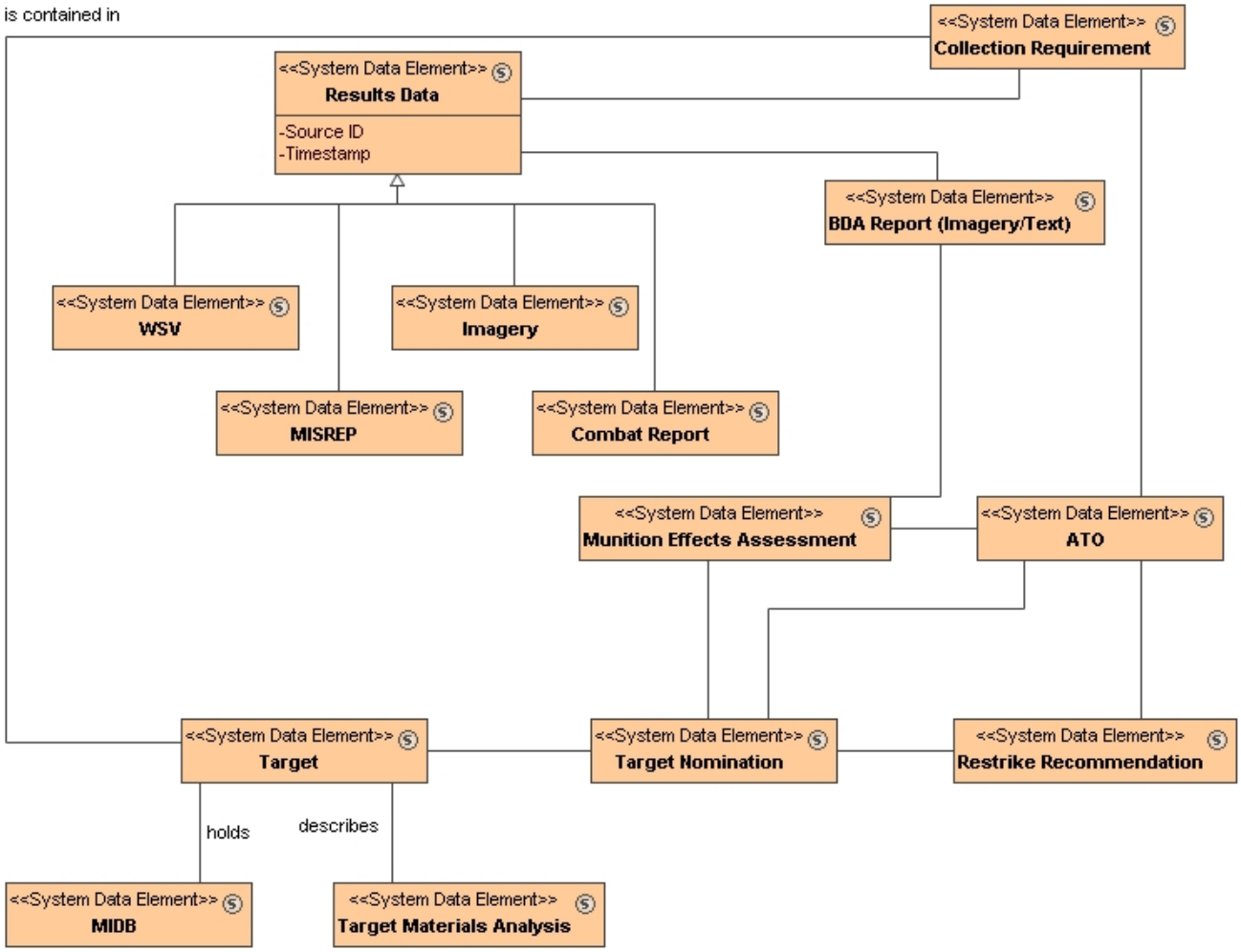

Figure 29 -- SV-11 sample

6 TECHNICAL STANDARDS VIEW PRODUCTS

TV-1. Technical Standards Profile

Description

“The Technical Standards Profile collects the various systems standards rules that implement and sometimes constrain the choices that can be made in the design and implementation of architecture. In most cases, especially in describing architectures with less than a Military Service-wide scope, TV-1 consists of identifying the applicable portions of the JTA and other existing technical standards guidance documents, tailoring those portions, as needed, in accordance with the latitude allowed in these guidance documents, and filling in any gaps.

The standards are referenced as relationships to the systems, system functions, system data, hardware/software items or communication protocols in SV-1, SV-2, SV-4, SV-6, OV-7, and SV-11 products, where applicable.” [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

TV-1 in a **table** that is implemented as special GUI in MagicDraw. User will be provided a special table-like GUI for filling out the TV-1 data. As in MagicDraw everything is based on the underlying model, creating TV-1 will also create the UML model.

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Standard	<<Standard>> [Class]	
Standards Profile	<<StandardsProfile>> [Package]	
Reference Model	<<ReferenceModel>> [Package]	N/A
Applied Standard	<<AppliedStandard>> [Usage]	N/A
Service Area	<<ServiceArea>> [Package]	
Service	<<Service>> [Package]	
Time Period	See the Time definitions	

Standard

Description

A set of rules set up and established by authority, custom, or general consent as a rule for the measure of quantity, weight, extent, value, or quality as a model or example.

Extends UML Class.

Attributes

type: StandardType	Type of the Standard. Possible values are collected from StandardType enumeration: De Jure De Facto Organizational Project Specific
options: String	Selected Standard options
parameters: String	Selected Standard parameters
startDate: String	Initial date on which Standard is applicable
endDate: String	Date after which Standard is no longer applicable

StandardsProfile

Description

A set of Standards that is made based on Reference Model and adopted for specific domain or platform.

Extends UML Package.

Attributes

applicableDate: String	Start date for use of the Standards Profile
basedOn: ReferenceModel	Reference Model on which Standards Profile is based

ReferenceModel

Description

Model used to select Services and organize Standards.

Extends UML Package.

Attributes

source: String	Reference to the source documentation and organization supporting the Reference Model

AppliedStandard

Description

Relationship that connects Standard with System View element (System, Hardware or Software Item, System Function, Communications System, Communications Link, Communications Network, LAN, WAN, MAN, System Data Exchange, System Data Element) for which Standard should be applied.

Extends UML Usage.

Attributes

appliedAt: TimePeriod [1]	Time Priod at which Standard should be applied

ServiceArea

Description

A group of Services defined for specific purpose.

Extends UML Package.

Attributes

version/date: String	Date or version number for the Service Area forecast (for use in forecast projects)

Service

Description

A service is a distinct part that is provided by a System on one side of an Interface to another System

Extends UML Package.

Attributes

status: String	Applicability of some Standard for this Service

Profile Diagram

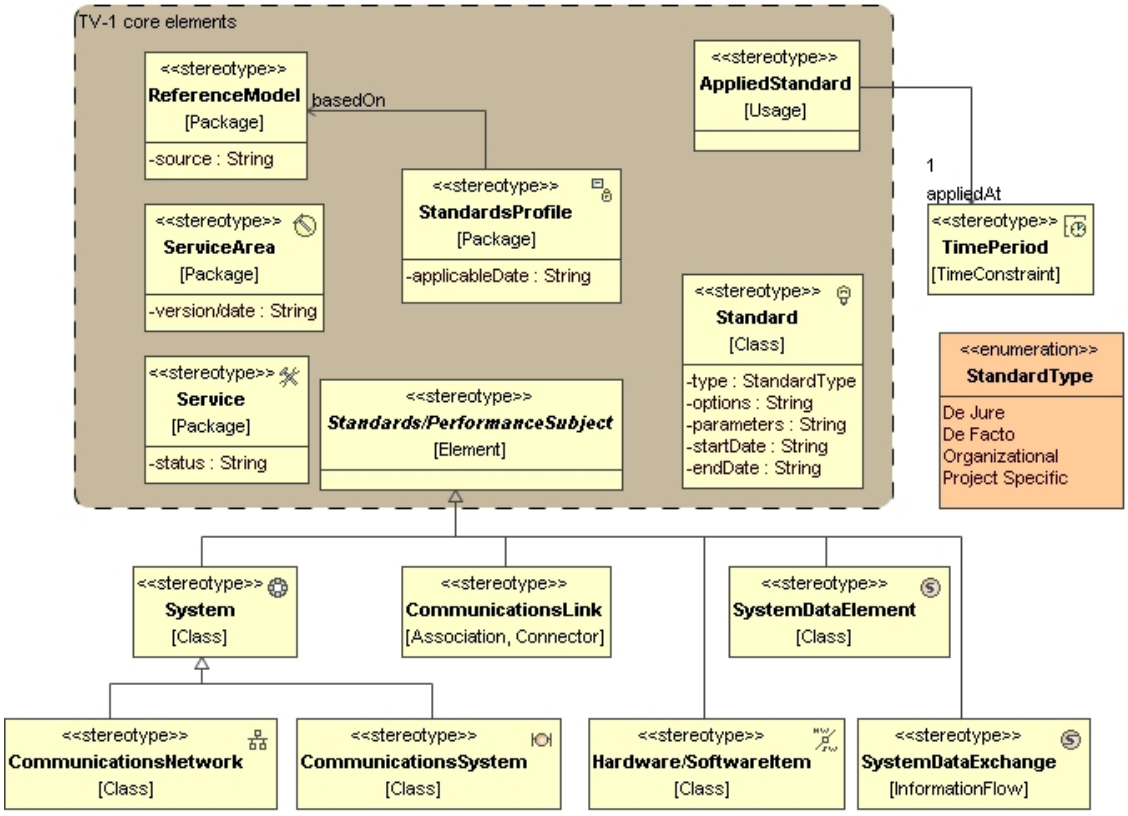

Figure 1 -- TV-1 profile diagram

Creating the product

When user will create TV-1 in the row scope he will have to select Standards that will be associated with the Systems. In the column scope he will have to select Systems that will implement or use Standards. After that, user will be provided with a special GUI-based table, where rows will be selected Standards. Columns will be collected from the selected Systems. User will fill in the cells, creating/selecting the Time Period elements that are will show the time the Standard is applied at the System

Despite this is a product that does not map to any diagram; DoDAF model will be also created in order to keep integrity with the rest of the model.

Relationship to other products

Systems from **SV-1**/**SV-2** will be linked to the Standards.

Sample

	CIGSS	ADSI	IP A/L
Information Technology Stand...			
Operating Environment [Te...			
DII COE [Technical Sta...	1 month= [Utils::...		
Operating System Standar...			
ISO/IEC 9945-1:1996 [...		3 months= [Utils:...	
IEEE 1003.2d:1994 [Te...			

Figure 2 -- TV-1 sample

Sample model

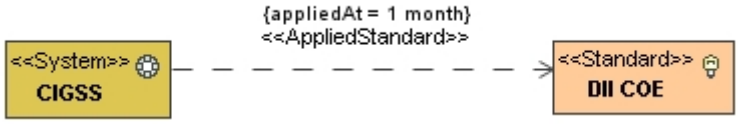

Figure 3 -- TV-1 sample model

TV-2. Technical Standards Forecast

Description

“The Technical Standards Forecast contains expected changes in technology-related standards and conventions, which are documented in the TV-1 product. TV-2 lists emerging or evolving technology standards relevant to the systems covered by the architecture. It contains predictions about the availability of emerging standards, and relates these predictions to the Systems View elements and the time periods that are listed in the SV-8 and SV-9. The same template as in TV-1 may be used to describe TV-2.” [*DoDAF, v1.0, Vol. II*]

Implementation in MagicDraw

TV-2 in a **table** that is implemented as special GUI in MagicDraw. User will be provided a special table-like GUI for filling out the TV-2 data. As in MagicDraw everything is based on the underlying model, creating TV-2 will also create the UML model.

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Standards Forecast Profile	<<StandardsForecastProfile>> [Package]	
Timed Standards Forecast	<<TimedStandardsForecast>> [Usage]	N/A
Timed Technology Forecast	See SV-9 data element definition	
Reference Model	See TV-1 data element definition	
Time Period	See the Time definitions	

StandardsForecastProfile

Description

The StandardsForecastProfile is a set of standards that can emerge in specific time period. The specific Time Periods selected and Standards being tracked should be coordinated with architecture transition plans.

Extends UML Package.

Attributes

basedOn: ReferenceModel [*]	Reference Models on which Standards Forecast Profile is based

TimedStandardsForecast

Description

Relationship used to relate prediction about availability of emerging Standard to System View element (System, Hardware or Software Item, System Function, Communications System, Communications Link, Communications Network, LAN, WAN, MAN, System Data Element, System Data Exchange) and Time Period for which forecast is made.

Extends UML Usage.

Attributes

timePeriod: TimePeriod [1]	Time Period for which Timed Standards Forecast is valid
discussion: String	Textual description of purpose of forecast
standardStatus: StandardStatusType	Expected standard status based on forecast. Values are collected from StandardStatusType enumeration. Possible values: Approved Draft Available Updated Replaced Obsolete COTS Implementation Available
requires: TimedTechnologyForecast [*]	A list of Timed Technology Forecasts that require Timed Standard Forecast

Profile diagram

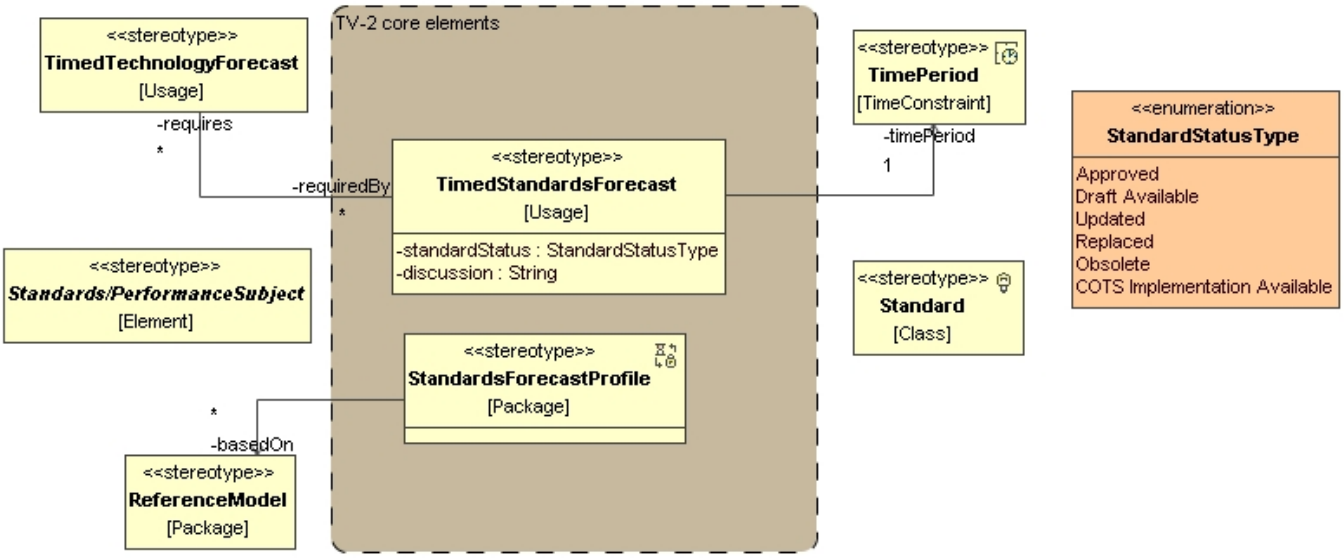

Figure 4 -- TV-2 profile diagram

Creating the product

When user will create TV-2 in the row scope he will have to select Standards that will be associated with the Systems. In the column scope he will have to select Systems that will implement or use Standards. After that, user will be provided with a special GUI-based table, where rows will be selected Standards. Columns will be collected from the selected Systems. User will fill in the cells, creating/selecting the Time Period elements that are will show the time the Standard is applied at the System

Despite this is a product that does not map to any diagrams; DoDAF model will be also created in order to keep integrity with the rest of the model.

Relationship to other products

Systems from **SV-1**/**SV-2** will be linked to the Standards.

Sample

	CIGSS	ADSI	TDL
Information Technology Stand...			
Operating Environment [Te...			
DII COE [Technical Sta...	1 yr= [Utils::Time...		
Operating System Standar...			
ISO/IEC 9945-1:1996 [...		1 yr= [Utils::Time...	
IEEE 1003.2d:1994 [Te...		3 yrs= [Utils::Tim...	

Figure 5 -- TV-2 sample

Sample model

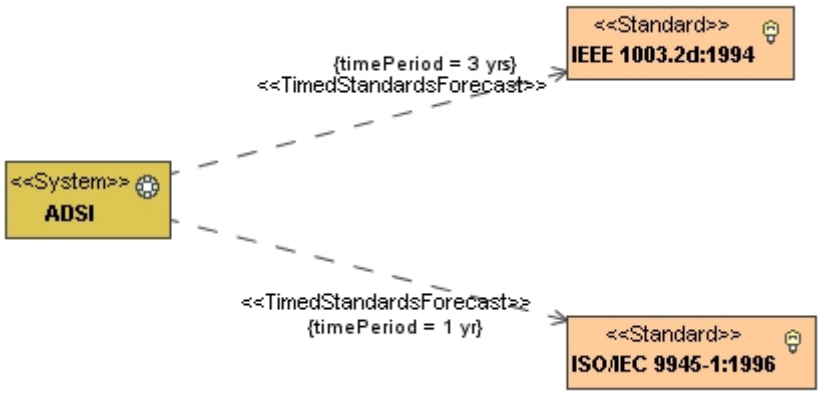

Figure 6 -- TV-2 sample model

7 CROSS-CUTTING ELEMENTS

Time definitions

UML Profile for DoDAF

DoDAF Element Type	UML Stereotype [metaclass]	Notation
Time Period	<<TimePeriod>> [TimeConstraint]	
Time Period List	<<TimePeriodList>> [Class]	

Time Period

Description

Time when the measurements were/will be taken.

Extends UML TimeConstraint.

Time Period List

Description

The set of ordered Time Periods used to define when measurements were or will be taken.

Extends UML Class.

Attributes

timePeriods: TimePeriod [1..*] {ordered}	A ordered list of Time Periods when the measurements were/will be performed

Creating Time definitions

Time Periods can be accessed from DoDAF main menu. For each Time Period name and duration can be specified. User can also change the order of Time Periods.

On SV-7 and SV-9 tables Time Periods are used as column elements. User can choose Time Periods at which measurements and forecasts should be taken. While selecting Time Period scope new Time Periods can be also created.

On SV-8 Time Period is specified for Milestone. User can select Time Period already existing in the model or create the new one.

On TV-1 and TV-2 Time Periods are used in the cells of tables. User is provided with the list of Time Periods. New Time Periods can be also created.