S*Metamodel Mapping

for

MagicDraw/Cameo Systems Modeler

Version 19

Version 1.11.4
2/13/2023

Systematica®

Do more with less

By: S* Patterns Community

Revision History

Date	Version	Description	Author
November 01. 2013	1.0	Initial Content	William Schindel, ICTT
January 19, 2015	1.1	Updated detail mapping	Jason Sherey, ICTT
December 9, 2016	1.2	Add S* Icon Table	Stephen Lewis, ICTT
January 23, 2017	1.3	Table and Scope Edits	Stephen Lewis, ICTT
January 27, 2017	1.4	Table Implementation Statuses	Stephen Lewis, ICTT
March 6-8, 2017	1.5	Updates to Implementation Statuses, SysML Elements Mapping	Stephen Lewis, ICTT
March 9, 2017	1.6	Profile Installation Section	Stephen Lewis, ICTT
June 9, 2017	1.7	Implementation Status Updates	Stephen Lewis, ICTT
June 13, 2017	1.8	Implementation Status Column Edits	Stephen Lewis, ICTT
April 26, 2019	1.9.1	Updated formats and mappings	Jason Sherey, ICTT
November 22, 2019	1.9.1a	Create Commons License nomenclature	Jason Sherey, ICTT
June 24, 2021	1.10.1	Updated mappings	Stephen Lewis, ICTT
January 12, 2022	1.11.1	Changed “Physical System” language to “Design Component”, Changed Architectural Relationship and Architectural Relationship Role Mappings, Added Interface Element Relationship to S* Metaclass Mapping Details, Added Relates AR, Relates FI, Relates IO, Relates Sys to S* Metarelationship Mapping Details. Edited Mapping Strategy section.	Stephen Lewis, ICTT
October 12, 2022	1.11.2	Updated Reference Details	Stephen Lewis, ICTT
February 10-13, 2023	1.11.3-4	Risk Analysis Mappings	Stephen Lewis, ICTT

Corporate Officer
ICTT System Sciences
378 South Airport Street
Terre Haute, IN 47803
812-232-2208

Table of Contents

1 Document Overview

1.1 Document Scope

This document describes the mapping between OMG SysML® and the S*Metamodel of the Systematica® methodology and its models and patterns. This document is further configured for Dassault Systemes MagicDraw/Cameo Systems Modeler Version 19. This mapping is implemented in version 01232023 of the Systematica Profile in MagicDraw/Cameo Systems Modeler.

1.2 Document Organization

This document is organized in the following fashion:

1. **Document Overview:** Reviews the scope, organization and references for this document.
2. **Framework Overview:** Reviews basic OMG SysML schema structures and concepts.
3. **Mapping Overview:** Provides a high-level overview of how the schema configuration uses the structures and concepts described in Section 2 and their mapping to S*Metaclasses and S*Metarelationships.
4. **Detailed Mapping:** Lists each S*Metaclass and S*Relationship and describes how each is mapped into MagicDraw.

1.3 Applicable Documents

The following are referenced or otherwise applicable to this document:

1. Friedenthal, S., et al.. *A Practical Guide to SysML*. Boston: Morgan Kaufmann OMG Press, 2012.
2. ICTT System Sciences. “Systematica Metamodel, Version 8.0.33.” March 2022.
3. ICTT System Sciences. “S*Metamodel to SysML Map.” *Visualizing and Understanding Systems Processes and Systems Pathologies, Volume I: Modeler’s Guide, V1.6.2*. INCOSE System Sciences Working Group, SP Modeling Team, December, 2013.
4. Object Modeling Group (OMG). “What Is SysML.” *OMG SysML*, 2018, omgsysml.org/what-is-sysml.htm. Accessed 18 Oct. 2018.

2 OMG SysML Overview

“The OMG Systems Modeling Language (OMG SysML™) is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems” (OMG). Being a graphical modeling language, SysML uses elements on a variety of diagrams to model a system’s requirements and designs.

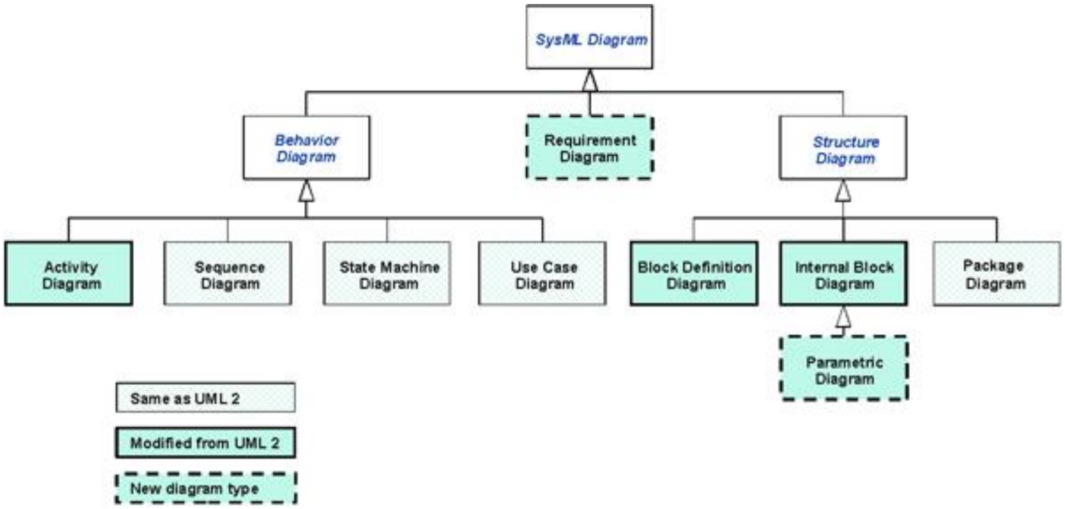


Figure 1: SysML Diagram Types (www.omgsysml.org)

“The «block» is the basic unit of structure in SysML and can be used to represent hardware, software, facilities, personnel, or any other system element. The system structure is represented by block definition diagrams and internal block diagrams. A block definition diagram describes the system hierarchy and system/component classifications. The internal block diagram describes the internal structure of a system in terms of its parts, ports, and connectors. The package diagram is used to organize the model” (OMG).

“The behavior diagrams include the use case diagram, activity diagram, sequence diagram, and state machine diagram. A use-case diagram provides a high-level description of functionality that is achieved through interaction among systems or system parts. The activity diagram represents the flow of data and control between activities. A sequence diagram represents the interaction between collaborating parts of a system. The state machine diagram describes the state transitions and actions that a system or its parts perform in response to events” (OMG).

“SysML includes a graphical construct to represent text based requirements and relate them to other model elements. The requirements diagram captures requirements hierarchies and requirements derivation, and the satisfy and verify relationships allow a modeler to relate a requirement to a model element that satisfies or verifies the requirements. The requirement diagram provides a bridge between the typical requirements management tools and the system models” (OMG).

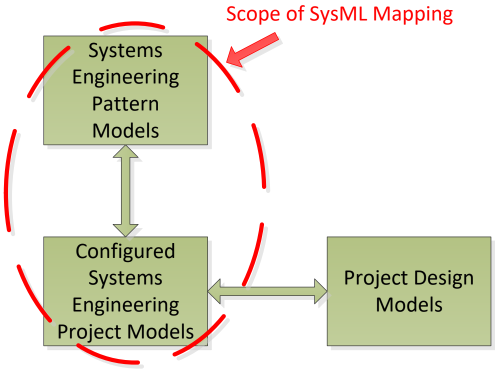
“The parametric diagram represents constraints on system property values such as performance, reliability, and mass properties, and serves as a means to integrate the specification and design models with engineering analysis models” (OMG).

“SysML also includes an allocation relationship to represent various types of allocation, including allocation of functions to components, logical to physical components, and software to hardware” (OMG).

3 Schema Configuration Overview

The mapping described in this document covers only systems engineering pattern and configured model spaces. Mappings for detail design models and any relationships between them and the configured systems engineering models are not contained in this document.

Figure 2: Mapping Model Scope


Scope of SysML Mapping
Systems Engineering Pattern Models
Configured Systems Engineering Project Models
Project Design Models

3.1 *Mapping Summaries*

The mapping for using OMG SysML with Systematica can be summarized by the following mapping between the Summary S*Metamodel Diagram and the SysML Mapping Overview Diagram.

Figure 3: Systematica Summary Meta-Model Diagram

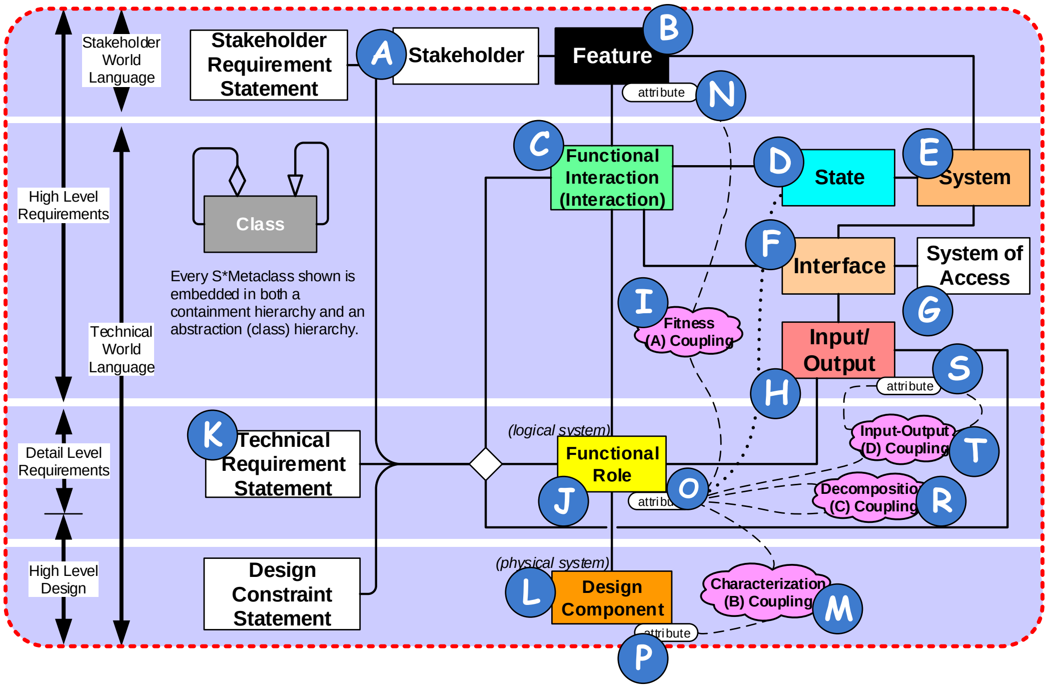
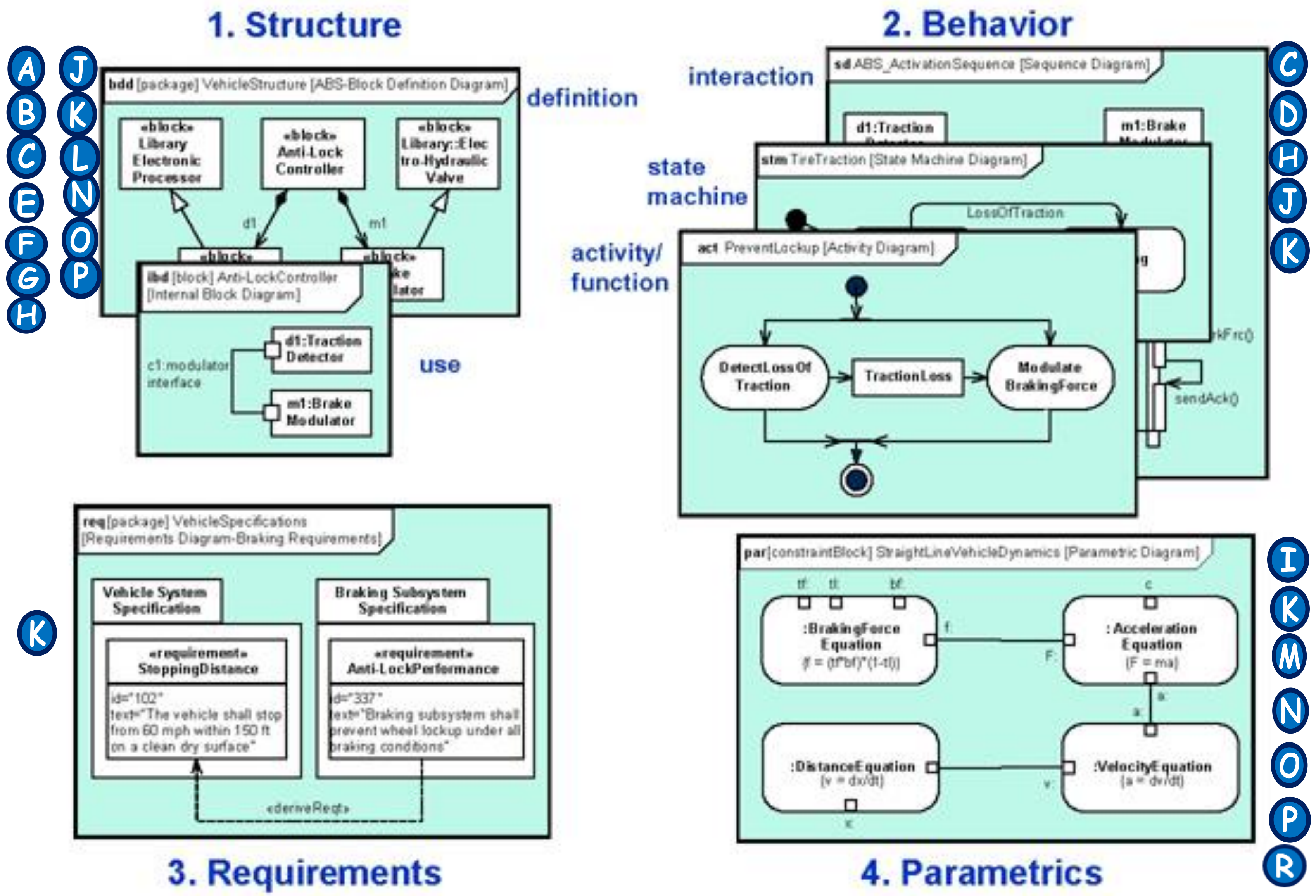

Stakeholder World Language
High Level Requirements
Technical World Language
Detail Level Requirements
High Level Design
Stakeholder Requirement Statement
Stakeholder
Feature
attribute
Class
Every S*Metaclass shown is embedded in both a containment hierarchy and an abstraction (class) hierarchy.
Functional Interaction (Interaction)
State
System
Interface
System of Access
Fitness (A) Coupling
Input/ Output
attribute
Input-Output (D) Coupling
Decomposition (C) Coupling
Technical Requirement Statement
(logical system)
Functional Role
attribute
(physical system)
Design Component
attribute
Characterization (B) Coupling
Design Constraint Statement
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
R
S
T

Figure 4: Systematica OMG SysML Mapping Overview Diagram


1. Structure
2. Behavior
3. Requirements
4. Parametrics
A
B
C
E
F
G
H
J
K
L
N
O
P
definition
use
interaction
state machine
activity/ function
C
D
H
J
K
I
K
M
N
O
P
R
bdd [package] VehicleStructure [ABS-Block Definition Diagram]
«block» Library Electronic Processor
«block» Anti-Lock Controller
«block» Library::Electro-Hydraulic Valve
d1
m1
ibd [block] Anti-LockController [Internal Block Diagram]
d1:Traction Detector
m1:Brake Modulator
c1:modulator interface
sd ABS_ActivationSequence [Sequence Diagram]
d1:Traction Detector
m1:Brake Modulator
stm TireTraction [State Machine Diagram]
LossOfTraction
act PreventLockup [Activity Diagram]
DetectLossOf Traction
TractionLoss
Modulate BrakingForce
sendAck()
req [package] VehicleSpecifications [Requirements Diagram-Braking Requirements]
Vehicle System Specification
Braking Subsystem Specification
«requirement» StoppingDistance
id="102"
text="The vehicle shall stop from 60 mph within 150 ft on a clean dry surface"
«requirement» Anti-LockPerformance
id="337"
text="Braking subsystem shall prevent wheel lockup under all braking conditions"
«deriveReqt»
par [constraintBlock] StraightLineVehicleDynamics [Parametric Diagram]
:BrakingForce Equation
(f = (tf*bf)*(1-tl))
: Acceleration Equation
(F = ma)
:DistanceEquation
(v = dx/dt)
:VelocityEquation
(a = dv/dt)
tf:
tl:
bf:
f:
F:
c
a
v:
x:

Note that the Package and Use Case diagrams are not shown in this example, but are respectively part of the structure and behavior pillars

The Systematica Meta-Model mapping to OMG SysML can be summarized by the following table:

Mapping Letter	Systematica Meta-Model Element	SysML Element Type
A	Stakeholder	Block with «Stakeholder» stereotype
B	Feature	Block with «Feature» stereotype
C	Functional Interaction (Interaction)	Activity with «Functional Interaction» stereotype
D	State	State with «State» stereotype
E	System	Block with «Logical System» stereotype
F	Interface	Proxy Port with «ifc port» stereotype typed by an Interface Block with an «Interface Definition» stereotype
G	System of Access	Block with «System of Access» stereotype
H	Input/Output	itemFlow that conveys a Classifier with an «Input_Output Definition» stereotype
I	Fitness (A) Couplings	Constraint Block with «Fitness Coupling» stereotype
J	Functional Role	Block with «Logical System» stereotype being referenced by an Interaction via a shared aggregation relationship with «Has Role» stereotype.
K	Requirement Statement	Requirement with «Requirement Statement» stereotype
L	Design Component	Block with «Design Component» stereotype
M	Characterization (B) Couplings	Constraint Block with «Characterization Coupling» stereotype
N	Feature Attribute	Attribute with «Feature Primary Key» or «Feature Attribute» stereotypes
O	Role Attribute	Attribute with «Logical System Attribute» stereotype
P	Design Component Attribute	Attribute with «Design Component Attribute» stereotype
R	Decomposition (C) Matrix Couplings	Constraint Block with «Decomposition Coupling» stereotype
S	IO Attribute	Attribute with «IO Attribute» stereotype
T	Input-Output (D) Coupling	Constraint Block with «Input Output Coupling» stereotype

3.2 High Level Mapping Strategy

The mapping was constrained to only map the classes and relationships of the S*Metamodel. The Systematica Process Views and Transactions are not mapped in this document.

The main mapping choices consisted of:

- Mapping most S* classes and relationships to standard SysML elements staying as close to the intended use of each SysML element as possible,
- Systematica Configured Model classes are mapped to SysML classes whose generalizations are the Systematica Pattern classes,
- Where no standard SysML element matches a Systematica class, a SysML “block” is used with a specialized stereotype,

- Where no standard SysML relationship matches a Systematica relationship, a SysML "dependency" is used with a specialized stereotype,
- Required, Capability, Baseline, Best In Class, or other value types of each Modeled Attribute are mapped to tags of those Modeled Attributes,
- To allow for possible use of SysML Activity Diagrams, SysML Activities are used to model parts of S* Interaction-Role-Requirement relationships and classes even if no diagram is actually drawn,
- S* Architectural Relationships have been bifurcated into simple and reified cases for mapping. In the simple case, it is mapped to the Association relationship between systems and given an Architectural Relationship stereotype. This is a natural fit for the simple case because SysML associations are restricted to only 2 roles. In the reified case, which involves more than 2 roles, it is mapped to a block with Architectural Relationship Definition. That block is then able to participate in dependency relationships with the N number of Architectural Relationship Roles.
- Using SysML classes instead of parts or properties to model components and referenced classes. The components are modeled under the container's namespace and also participate in a composition with the container. This results in minimal use of the parts that occur from the association because Systematica emphasizes that a class is defined by its relationships across all of its uses as opposed to a class being defined as a library entity with independent contextual references.

4 Detail Mapping

4.1 S*Metaclass Mapping Details

The Systematica Class mappings to SysML can be summarized by the following table:

Modeled Class Name	Core Properties	SysML Element
Allocation Decision	Author, Change Date, Change Description, Class Level, Definition, ID, Version, Name, Organization Owner, Status	«allocate» dependency
Alternative	Author, Change Date, Change Description, Class Level, Definition, ID, Version, Name, Organization Owner, Status, Rank, Score	«allocate» dependency
Architectural Relationship	Name, Definition, Domain Diagram, Logical Architecture Diagram, Physical Architecture Diagram, Status, Version, Date Changed, Changed by, Change Description, Issue/Discussion Tree	Simple: Name of the Association relationship between systems with «architectural_relationship» stereotype; Reified: Block with «Architectural Relationship Definition» stereotype
Architectural Relationship Role	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Simple: Association Target and Source Roles; Reified: Dependency with «Has AR Role» stereotype
Attribute Coupling	Name, Definition, Owner, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Constraint Block with «Attribute Coupling» stereotype

Modeled Class Name	Core Properties	SysML Element
Attribute Coupling Map	Name, Definition, Reference Owner, Reference, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Constraint of a Constraint Block, Artifact referenced or linked to by Constraint
Attribute Role	Name, Definition, Owner, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Parameter of Constraint Blocks
Class	Name, Definition, Contact Information, Location, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	This is a superclass of other Metaclasses and is not mapped.
Counter Requirement	Name, Definition, Contact Information, Location, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Requirement with «Counter Requirement» stereotype
Design Component Attribute Role	Name, Definition, Owner, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Parameter of Constraint Blocks
Design Constraint	Name, Definition, Status, Version, Date Changed, Changed by, Change Description, Issue/Discussion Tree	Requirement with «Design Constraint» stereotype
Design Constraint Statement	Name, Definition, Status, Version, Date Changed, Changed by, Change Description, Issue/Discussion Tree, Owner, Reference	Requirement with «Design Constraint» stereotype

Modeled Class Name	Core Properties	SysML Element
Design Coupling	Name, Definition, Reference Owner, Contact Information, Location, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Constraint Block with «Characterization Coupling» stereotype.
Design Coupling Map	Name, Definition, Reference Owner, Reference, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Constraint within Constraint Block, Artifact referenced or linked to by Constraint
Domain	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Block with «Domain» stereotype
Domain System	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Block with «Logical System» stereotype aggregated within a Domain
Event	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Trigger with «Event» stereotype
Failure Impact	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Block with «Failure Impact» stereotype

Modeled Class Name	Core Properties	SysML Element
Failure Mode	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	State with «Failure Mode» stereotype
Failure Mode Context Element	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Block with «Failure Mode Context Element» stereotype
Feature	Name, Definition, Configuration Rule, PK Name, PK Value, Priority, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Block with «Feature» stereotype
Feature Attribute Role	Name, Definition, Owner, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Parameters of Constraint Blocks
Functional Interaction	Name, Definition, IPK Value, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Activity Block with «Functional Interaction» stereotype.
Functional Role	Name, Definition, RPK Value, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Block with «Logical System» stereotype and in Aggregation relationship with «Has Role» stereotype with Interaction Activity Block
Information Input-Output	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	itemFlow that conveys a Classifier with an «Input_Output Definition» stereotype

Modeled Class Name	Core Properties	SysML Element
Input-Output	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	itemFlow that conveys a Classifier with an «Input_Output Definition» stereotype
Input Role	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Action Pin on an Activity modeling a Requirement Relationship
Interface	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Proxy Port with «ifc port» stereotype typed by an Interface Block with an «Interface Definition» stereotype
Interface Element	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Block with «Interface Element» stereotype.
Issue	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree, Author	Note or tool-specific discussion tree construct
Logical System	Name, Definition, Domain Diagram, Logical Architecture Diagram, Status, Version, Date Changed, Changed by, Change Description, Issue/Discussion Tree	Block with «Logical System» stereotype
Logical System (Advocate)	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Logical System Block with «Advocate» stereotype that only participates in advocate-type relationships

Modeled Class Name	Core Properties	SysML Element
Logical System (Stakeholder)	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Logical System Block with «Stakeholder» stereotype that only participates in stakeholder-type relationships
Modeled Attribute (Feature Attribute)	Name, Definition, Possible Values, Units, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Attribute with «Feature Attribute» or «Feature Primary Key» stereotypes
Modeled Attribute (IO Attribute)	Name, Definition, Possible Values, Units, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Attribute with «IO Attribute» stereotype
Modeled Attribute (Physical System Attribute)	Name, Definition, Possible Values, Units, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Attribute with «Physical System Attribute» stereotype
Modeled Attribute (Role Attribute)	Name, Definition, Possible Values, Units, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Attribute with «Logical System Attribute» stereotype
Modeled Relationship	Name, Definition, Possible Values, Units, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	This is a superclass of other meta-classes and is not mapped.
Modeled Relationship Role	Name, Definition, Possible Values, Units, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	This is a superclass of other meta-classes and is not mapped.

Modeled Class Name	Core Properties	SysML Element
Modeled Statement	Name, Definition, Possible Values, Units, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	This is a superclass of other meta-classes and is not mapped.
Need	ID, Text, Source, Priority, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Requirement with «Need» stereotype
Output Role	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Action Pin on an Activity modeling a Requirement Relationship
Physical Input-Output	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	itemFlow that conveys a Classifier with an «Input_Output Definition» stereotype
Design Component	Name, Definition, Domain Diagram, Physical Architecture Diagram, Configuration Rule, Status, Version, Date Changed, Changed by, Change Description, Issue/Discussion Tree	Block with «Design Component» stereotype
Port	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree, Port PK Rule, Interface PK Rule, IO PK Rule, SOA PK Rule	Flow Property with «Input Output» stereotype.

Modeled Class Name	Core Properties	SysML Element
Rationale	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	«allocate» dependency
Requirements Coupling	Name, Definition, Reference Owner, Contact Information, Location, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Constraint Block with «Fitness Coupling», «Decomposition Coupling», or «Input Output Coupling» stereotype
Requirements Coupling Map	Name, Definition, Reference Owner, Reference, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Constraint within Constraint Block, Artifact referenced or linked to by Constraint
Requirement Relationship	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Activity with «Requirement Transfer Function» stereotype
Requirement Statement	ID, Text, Rationale, RSPK Value, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Requirement «Requirement Statement» stereotype
Role Attribute Role	Name, Definition, Owner, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Parameter of Constraint Blocks
State	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	State with «State» stereotype

Modeled Class Name	Core Properties	SysML Element
System	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Block with «Logical System» stereotype.
System of Access (SOA)	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Logical System block with «System of Access» stereotype.
Transition	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	SysML Transition
Value	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree, Value	Required, Capability, Baseline, Best In Class, or other value types of each Modeled Attribute are mapped to tags of those Modeled Attributes
Interface Element	Name, Definition, Status, Version, Date Changed, Changed By, Change Description, Issue/Discussion Tree	Block with «Interface Element» stereotype.

4.2 S*Metarelationship Mapping Details

The S*Metarelationship mappings to SysML can be summarized by the following table:

Modeled Relationship Type	Core Properties	Source Class	Target Class	SysML Relationship
Advocates	(none)	Advocate	Need	Dependency with «Advocated_By (N-ADV)» stereotype

Modeled Relationship Type	Core Properties	Source Class	Target Class	SysML Relationship
Abnormal State Of	DCPK Value Rule, FMPK Value Rule, Probability	Failure Mode	Design Component	Dependency with «Abnormal State Of» stereotype
Allocated To	(none)	Modeled Relationship Role	Class	Association with «Functional Role Allocation» stereotype for LS-DC; Binding Connection for Attribute-Constraint Property or connection between Flow -Action Pin; the trace between a System and the end of an Connector that conveys «Architectural Relationship Definition» items, sometimes indirectly through an Interface port
Appears In	(none)	Class	Domain	Aggregation with between Domain and other classes represented as Blocks.
Benefits	(none)	Stakeholder	Feature	Dependency with «Benefits (FTR-STK)» stereotype
Causes Behavior	(none)	Failure Mode	Counter Requirement Name	Dependency with «Causes Behavior» stereotype
Causes Impact	(none)	Counter Requirement Name	Failure Impact	Dependency with «Causes Impact» stereotype

Modeled Relationship Type	Core Properties	Source Class	Target Class	SysML Relationship
Contains	(none)	Class	Class of Same Type	Composition; Contained class is also included in the namespace of/owned by the containing class. «containment population rule» stereotype indicates that there are whole pk value and part pk value configuration rules for the names of the respective association roles.
Derived From	Sufficiency Status	Class	Class	«trace» dependency
Emerges From	(none)	Architectural Relationship	Input-Output	Dependency with «Resolves (IO-AR)» stereotype
Exemplifies	(none)	Input-Output	Architectural Relationship	Dependency with «Exemplified By (AR-IO)» stereotype
Groups	(none)	Interface	Port	«Input Output» flow property owned by «Interface Definition» interface block
Has Advocate	(none)	Logical System (Stakeholder)	Logical System (Advocate)	Dependency with «Represents (ADV-STK)» stereotype

Modeled Relationship Type	Core Properties	Source Class	Target Class	SysML Relationship
Has Attribute	PK?	Feature	Feature Attribute	Block-Block Attribute SysML relationship
Has Attribute	(none)	Design Component	Design Component Attribute	Block-Block Attribute SysML relationship
Has Attribute	(none)	Functional Role	Role Attribute	Block-Block Attribute SysML relationship
Has Feature	(none)	System	Feature	Dependency with «Offered By (FT-SYS)» stereotype
Has Issue	(none)	Class	Issue	SysML Note link or internal tool relationship for internal tool discussion tree.
Has Previous	(none)	Class	Class	MagicDraw/CSM Configuration Management options
Has Role	IPK Value, RPK Rule	Modeled Relationship	Modeled Relationship Role	Aggregation with «Has Role» stereotype between Interaction and Logical System; Constraint Block-Property composition; Activity-Action Pin composition; Connector-Connector End relationship

Modeled Relationship Type	Core Properties	Source Class	Target Class	SysML Relationship
Has Stakeholder	(none)	Logical System	Logical System (Stakeholder)	Dependency with «Serves (LS-STK)» stereotype
Has State	(none)	Logical System	State	Dependency with «Chronicles (ST-LS)» stereotype
Has Subject	(none)	Domain	Subject	Aggregation with stereotype «Has Subject»
Has Value	(none)	Attribute	Value	value property's-tag value relationship
Has View	(none)	Modeled Relationship	Modeled Statement	SysML «Satisfy» dependency from «Requirement Transfer Function» Activity to «Requirement Statement» Requirement; Dependency with «Is Specified By (RTF-IO CPL)» between Requirement Transfer Function activity block and IO Attribute Coupling constraint block.
Impacts Feature	Feature PK Value Rule, Failure Impact PK Value Rule, Severity	Failure Impact	Feature	Dependency with «Impacts Feature» stereotype
Interacts Through	(none)	System	Port	«System»-«ifc port»-«Input Output» or «System»-«Input Output» containment/nesting

Modeled Relationship Type	Core Properties	Source Class	Target Class	SysML Relationship
Is A Type Of	Consistency Status	Class	Class of same Type	Generalization; Dependency with «Is Configuration Of» stereotype indicates pattern sources of configured model classes
Is Constrained By	(none)	Component	Design Constraint	«specifies» dependency between Design Constraint and other classes
Is Facilitated By Externally	(none)	Port	Logical System:System of Access	«Facilitated By (Port-SOA)» dependency
Is Facilitated By Internally	(none)	Port	Logical System:System of Access	«Facilitated By (Port-SOA)» dependency
Is Linked By Externally	(none)	Port	Architectural Relationship	Dependency with «Resolves (Port-AR)» stereotype
Is Linked By Internally	(none)	Port	Architectural Relationship	Dependency with «Resolves (Port-AR)» stereotype
Is Specified By	RPK Value	Logical System	Requirement Relationship	«allocate» dependency from «Requirement Transfer Function» activity block to Logical System block
Is Triggered By	(none)	Transition	Event	SysML transition-trigger association

Modeled Relationship Type	Core Properties	Source Class	Target Class	SysML Relationship
Is Used During	(none)	Port	Interaction	Abstraction with «Is Used During (IO-FI)» stereotype
Perceives	(none)	Stakeholder	Need	Dependency with «Perceived By (N-STK)» stereotype
Permits Architectural Relationship	(none)	Interface	Architectural Relationship	Ownership between an Interface block with «Interface Definition» stereotype that owns a flow property with «Architectural Relationship» stereotype
Permits Functional Interaction	(none)	Interface	Functional Interaction	Dependency with «Permits FI (IFC-FI)» stereotype
Permits Input-Output	(none)	Interface	Input-Output	Ownership between an Interface block with «Interface Definition» stereotype that owns a flow property with «Input Output» stereotype
Permits SOA	(none)	Interface	Logical System:System of Access	Dependency with «Permits SOA (IFC-SOA»)stereotype
Provides Failure Context	(none)	Failure Mode Context Element	Interaction	Dependency with «Provides Failure Context» stereotype
Provides Failure Mode Context	(none)	Failure Mode Context Element	Failure Mode	Dependency with «Provides Failure Mode Context» stereotype

Modeled Relationship Type	Core Properties	Source Class	Target Class	SysML Relationship
Provides Context	IPK Value	Functional Interaction	Requirement Relationship	Interaction activity owning a Requirement Transfer Function activity block
Provides Interface	(none)	System	Interface	Owns relationship between Block and Proxy Port with «ifc_port» stereotype, then follow Proxy Port type definition to Interface Block with «Interface Definition» stereotype
Receives	(none)	Port	Input-Output	The trace from a Flow Property with «Input Output» stereotype with an in setting to its owning proxy port with «ifc_port» stereotype to an itemFlow whose conveyed classifier is the same type as the Flow Property itself.
Relates AR	(none)	Architectural Relationship	Interface Element	Dependency with «AR-IE» Stereotype
Relates FI	(none)	Functional Interaction	Interface Element	Dependency with «FI-IE» Stereotype
Relates IO	(none)	Input/Output	Interface Element	Dependency with «IO-IE» Stereotype

Modeled Relationship Type	Core Properties	Source Class	Target Class	SysML Relationship
Relates Sys	(none)	Logical System	Interface Element	Dependency with «LS-IE» Stereotype
Replaces	RSPK Value Rule, Counter Requirement PK Value Rule	Counter Requirement	Requirement	Dependency with «Replaces» Stereotype
Requires	(none)	State	Functional Interaction	«Abstraction» dependency from a State to Functional Interaction
Satisfies	Sufficiency Status	Need	Feature	Dependency with «Satisfies (FT-N)» stereotype
Sends	(none)	Port	Input-Output	The trace from a Flow Property with an out setting with «Input Output» stereotype to its owning proxy port with «ifc_port» stereotype to an itemFlow whose conveyed classifier is the same type as the flow property itself.
Transitions From	(none)	Transition	State	SysML state-transition relationship
Transitions To	(none)	Transition	State	SysML state-transition relationship
Uses_Functional_Interaction	FPK Value, IPK Rule	Feature	Functional Interaction	Aggregation with «Uses Functional Interaction» stereotype