Interpreting SysMLv2 into the Category of Relations for Architecture, Analysis and Design

OMG Q4, December 2024

Mathsig Presentation to SE DSIG

C.E. Dickerson, S. Ji
Loughborough University

M.K. Wilkinson
BAE Systems

K. Oppl
Xephor Solutions

Mathsig: OMG Mathematical Formalism Domain Special Interest Group (DSIG)

mathsig/2024-12-01

Background

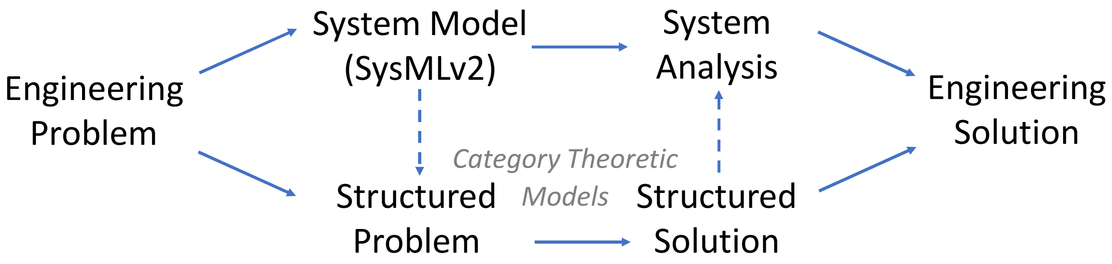
- Ongoing research with industry partners within and outside OMG
- Mathsig Presentations
  - September 2024: AI PTF, C4 DTF, Mathsig
    A Brief Introduction to Category Theory for Systems and Software Engineers [1]
  - June 2024: Mathsig
    ROSETTA Implementation in SysMLv2 Revisited [2]
  - March 2023: SE DSIG
    Implementation of ROSETTA in SysMLv2 and Underlying Maths Formalisms [3]
- This builds on and extends UPR 1.0 Constraint Driven Design [4]

Investigating practical methods for industrial applications supported by commercially available tools and founded in mathematics.

Context: Using SysMLv2 to Solve Engineering Problems

Category Theoretic Models Directly Reflect Solutions

Transformation of SysMLv2 models into mathematical abstractions



Purpose of presentation: inform nonspecialists in the SE DSIG how category theory might be used to enhance SysMLv2 tool expressiveness.

Topics of the Presentation

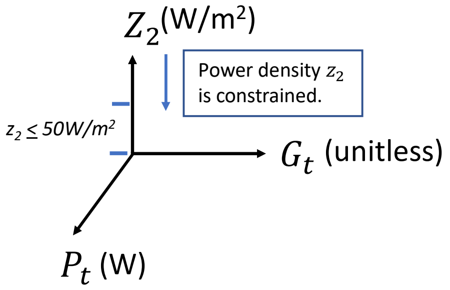
- ***Structuring Problems at a Higher Level of Abstraction***
- Definition, Usage, and Analysis in SysMLv2
- Category Theory as a Language for Architecture, Analysis and Design
- Conclusions: Complementing SysMLv2 with Category Theory

Category theory can be leveraged for structured modelling & analysis.

Reference Problem: Safety Constrained Radar System Design

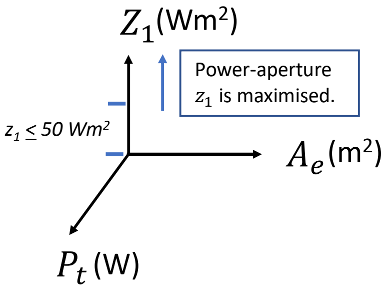
A safety requirement on power density (z_2) constrains the power-aperture product (z_1).

System design analysis is driven by the competing objectives z_1 and z_2.



Safety Design Space: $X_2 = (P_t, G_t)$

At the safety perimeter of d meters, power density z_2 must be $\leq 50W/m^2$.



Performance Design Space $X_1 = (P_t, A_e)$

Radar detection performance is improved by increasing power-aperture product.

Using the Category of Relations (*Rel*) for System Modelling

Correspondence with object orientation

Category theory reveals how different types of structures are related to one another [5, §1.3].

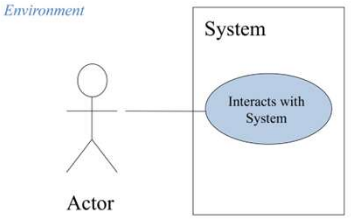
Key ideas:

- Models are central to systems and software
- Mathematical models are realised in mathematical
  - relational structures to include
  - graphical structures (e.g., in object orientation)
- Predicates can express *relationships between objects*.

This use case diagram states, 'Actor interacts with System.'

In category theory, a *relationship* between two objects is represented by an *arrow* called a *morphism*.

A category is concerned with its morphisms; *not details of its objects*.



(a) Predicate (verb-noun phrase) in Use Case diagram

$$A \xrightarrow{P} S$$

(b) Predicate as a morphism in the Category of Relations (*Rel*)

Rel : the category of relations. The objects of *Rel* are sets. Its morphisms are binary relations.

Using the Category of Relations (*Rel*)to Express Domain Knowledge

Example of knowledge representation using (relational) morphisms

Structures in category theory can be used to represent knowledge (e.g., ontologies, models).

Key ideas:

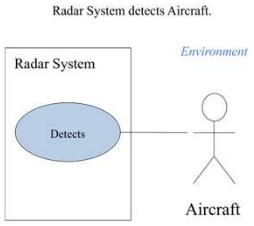
Relations can be used to express properties of objects.
Expression of properties as morphisms in the category *Rel,*

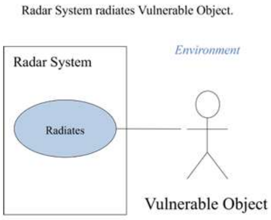
$F: X_1 \rightarrow Z_1$ (e.g., performance: power-aperture product)

$H: X_2 \rightarrow Z_2$ (e.g., safety: power density)

These two *morphisms are abstract structural elements* that reveal how the radar transmitter and antenna relate to system design.

Radar system domain knowledge can be expressed using the morphisms to define system properties.





**In [6], Architecture is investigated as a mathematical class and properties of structures.*

Analysis Case 1: Radar System Performance

Radar power-aperture product (Wm²) drives detection

F: $X_1 \rightarrow Z_1$; $R(z_1) \subseteq Z_1$ (performance requirement: maximise detection)

X_1 is a *design space*[1] with two attributes power (W), aperture (m^2)

Z_1 is an *objective space*[1]*:* attribute power-aperture product (Wm^2)

Design problem: maximise z_1 = power-aperture product (Wm^2)

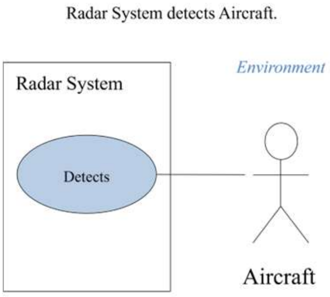
OMG UPR 1.0 Constraint Driven Design[1]

Requirement (as a constraint) →

Knowledge Map

$$\begin{tikzcd}
R \arrow[d, dashed] \arrow[r, phantom, "\subseteq"] & Z_1 \\
V \arrow[r, phantom, "\subseteq"] & X_1 \arrow[u, "F"']
\end{tikzcd}$$

Solution Set →



The association line in the use case can be expressed as a binary relation F between the objective and design spaces.

[1] *For further details refer to [4]* https://www.omg.org/spec/UPR/1.0/About-UPR/

Analysis Case 1: Radar System Performance

Radar power-aperture product (Wm^2) system analysis

$F: X_1 \rightarrow Z_1$; $R(z_1) \subseteq Z_1$ (performance requirement: maximise detection)

X_1 is a *design space* with two attributes power (W), aperture (m^2)

Z_1 is an *objective space:* attribute power-aperture product (Wm^2)

Design problem: maximise z_1 = power-aperture product (Wm^2)

At this stage of analysis, the requirement set R is unconstrained.

OMG UPR 1.0 Constraint Driven Design

$$\begin{tikzcd} R \arrow[d, dashed, "F^{-1}"'] \arrow[r, phantom, "\subseteq"] & Z_1 \\ V \arrow[r, phantom, "\subseteq"] & X_1 \arrow[u, "F"'] \end{tikzcd}$$

Requirements Transformation

Knowledge Map

Solution Set

Mathematical Analysis[1]

If $R \cap F(X_1) \neq \phi$, then $F: X_1 \rightarrow Z_1$ implies an inverse relation,

$$F^{-1}: R \rightarrow V$$

The inverse relation F^{-1} is not necessarily defined on the whole of R. It is only defined on $R \cap F(X_1)$.

Therefore, the structured solution set is,

$$V = F^{-1}(R \cap F(X_1))$$

---

[1] *For further details refer to [7]* Architecture, Analysis, and Design of Systems Using Extensions of Category Theory

Analysis Case 2: Radar System Safety

Radar power density (W/m^2) drives the safety constraint

$H: X_2 \rightarrow Z_2$; $S(z_2) \subseteq Z_2$ (safety requirement: constrain radiation)

X_2 is a *design space*[1] with two attributes power (W), gain (unitless)

Z_2 is an *objective space*[1]*:* attribute power density at range *d* (W/m^2)

Design problem: constrain z_2 = power density (W/m^2) at range *d*.

OMG UPR 1.0 Constraint Driven Design[1]

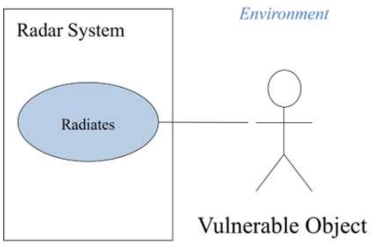
$$\begin{tikzcd} S \arrow[d, dashed] \arrow[r, phantom, "\subseteq"] & Z_2 \\ W \arrow[r, phantom, "\subseteq"] & X_2 \arrow[u, "H"'] \end{tikzcd}$$

Requirement (as a constraint)

Knowledge Map

Solution Set

Radar System radiates Vulnerable Object.



The association line in the use case can be expressed as a binary relation H between the objective and design spaces.

---

[1] *For further details refer to [4]* https://www.omg.org/spec/UPR/1.0/About-UPR/

Analysis Case 2: Radar System Safety

Radar power density (W/m^2) safety analysis

$H: X_2 \rightarrow Z_2$; $S(z_2) \subseteq Z_2$ (safety requirement: constrain radiation)

X_2 is a *design space* with two attributes power (W), gain (unitless)

Z_2 is an *objective space:* attribute power density at range *d* (W/m^2)

Design problem: constrain z_2 = power density (W/m^2) at range *d*.

At this stage of analysis, the requirement set S is constrained at the range d by $z_2 \leq 50\ \mathrm{W/m^2}$.

OMG UPR 1.0 Constraint Driven Design

Requirements Transformation

Knowledge Map

$$\begin{tikzcd} S \arrow[r, phantom, "\subseteq"] \arrow[d, dashed, "H^{-1}"'] & Z_2 \\ W \arrow[r, phantom, "\subseteq"] & X_2 \arrow[u, "H"'] \end{tikzcd}$$

Solution Set →

Mathematical Analysis[1]

If $S \cap H(X_2) \neq \phi$, then $H: X_2 \rightarrow Z_2$ implies an inverse relation,

$$W^{-1}: S \rightarrow W$$

The inverse relation H^{-1} is not necessarily defined on the whole of S. It is only defined on $S \cap H(X_2)$.

Therefore, the structured solution set is,

$$W = H^{-1}(S \cap H(X_2))$$

[1] *For further details refer to [7]* Architecture, Analysis, and Design of Systems Using Extensions of Category Theory

---

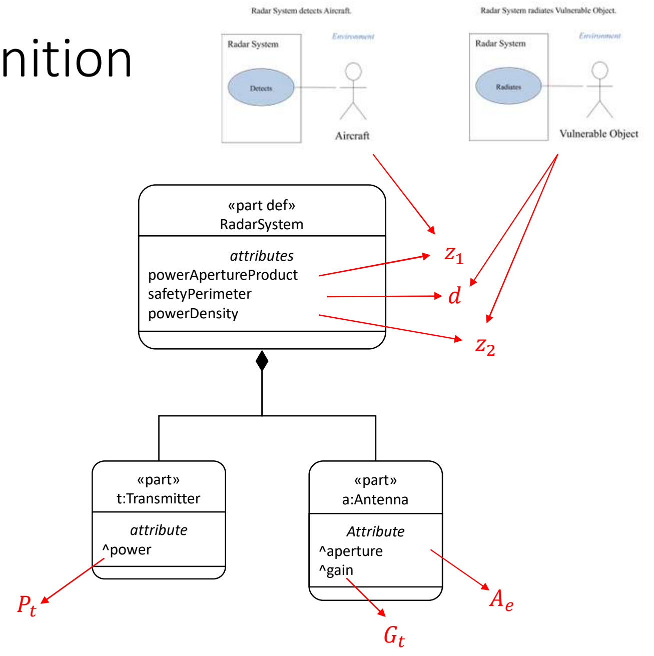
Topics of the Presentation

- Structuring Problems at a Higher Level of Abstraction
- ***Definition, Usage, and Analysis in SysMLv2***
- Category Theory as a Language for Architecture, Analysis and Design
- Conclusions: Complementing SysMLv2 with Category Theory

This section proposes preliminary concepts for a future SysMLv2.1.

Symbolic Variable Definition

- *Requirement:* A user shall be able to define a **symbolic variable** within a system element that is characterised by such a variable.
- For example, a transmitter component of a Radar system is characterised by the power of the transmitter. This characteristic could be modelled by a variable, named P_t.
- This could be supported by SysMLv2 Part Definition, Part Usage, Attribute Definition and Attribute Usage, but needs be more constrained.
- Requires analysis tool to read the attributes as **symbolic variables**



Radar System detects Aircraft.

Radar System radiates Vulnerable Object.

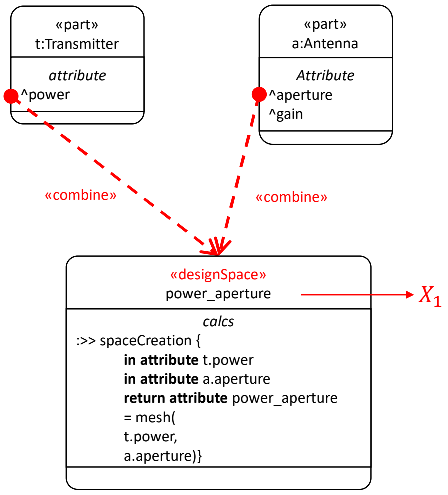
«part def»
RadarSystem
attributes
powerApertureProduct
safetyPerimeter
powerDensity

«part»
t:Transmitter
attribute
^power

«part»
a:Antenna
Attribute
^aperture
^gain

Space* Definition

- *Requirement*:
  - A user shall be able to specify a **Design Space** by selecting and combing a set of specified variables, represented by a symbolic variable. E.g., Design Space $X_1 = (P_t, A_e)$
  - A user shall be able to specify an **Objective Space** by selecting and combining a set of specified variables, represented by a symbolic variable. E.g., Object Space Z_1
  - The specification of the spaces shall optionally allow the user to specify the boundary for each of the variables applicable to the space definition so that the space is constrained
- *Proposal*:
  - «designSpace def» & «objectiveSpace def» extends Definition with a calcs compartment that utilises «cal def»;
  - «combine» that extends Dependency for visualisable traceability.
- «combine» can be realised by *Cartesian Product.* Ideally, this should be a machine-readable operation which will combine one-dimensional **variables** into a multi-dimensional **space**. *function* mesh() is a type of Cartesian Product.



«part»
t:Transmitter
attribute
^power

«part»
a:Antenna
Attribute
^aperture
^gain

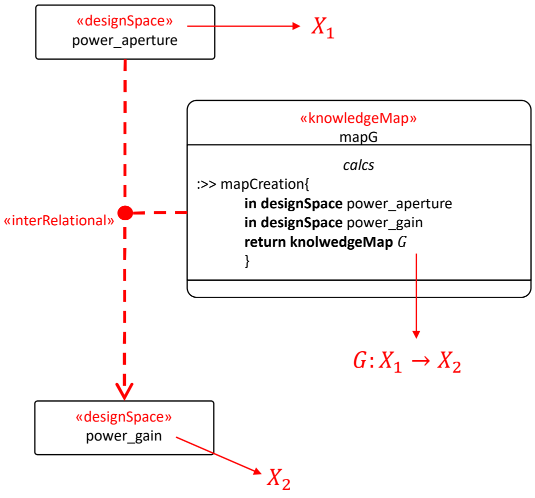
«designSpace»
power_aperture
calcs

```
:>> spaceCreation {
        in attribute t.power
        in attribute a.aperture
        return attribute power_aperture
        = mesh(
        t.power,
        a.aperture)}
```

* *For further details refer to [4],* https://www.omg.org/spec/UPR/1.0/About-UPR/

Relations Definition

- *Requirement*:
  - A user shall be able to specify **Binary Relations** between **symbolic variables**. E.g., a mathematical function, $f(P_t, A_e) = P_t A_e$, in which the inverse may **NOT** be a function.
  - A user shall be able to specify a **Relation** (of relations) that connects two specified **Spaces**. These could be a relation between two Design Spaces, a relation between two Objective Spaces, or a relation between a Design Space and an Objective Space. E.g., $F: X_1 \rightarrow Z_1$
- The first requirement is already addressed by SysMLv2
- Whilst there is a –nary Dependency graphical/textual notation, but it is rather abstract for analysis.
- *Proposal*:
  - «knowledgeMap def» extends Definition with a calcs compartment that utilises «cal def»;
  - «interRelational» that extends –nary Dependency for visualisable inter-relational structure and traceability.



Analysis Cases Issues

«analysis»
radarPerformance

subject
radarAlternatives [*] :>Radar

objective
doc maximize the detection range with varying power and aperture

calcs
:>> evaluationFunction {
in attribute radar.t.power
in attribute radar.a.aperture
return attribute radarPerformance = f(
radar.t.power,
radar.a.aperture)}

«analysis»
radarSafety

subject
radarAlternatives [*] :>Radar

Objective
doc minimise power density with varying power and aperture

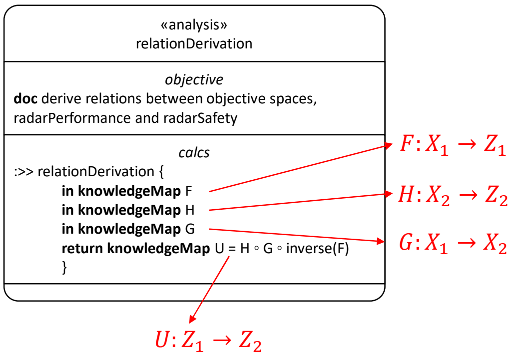
calcs
:>> evaluationFunction {
in attribute radar.t.power
in attribute radar.a.aperture
In attribute radar.safetyPerimter
return attribute radarSafety = h(
radar.t.power,
radar.a.gain, radar.safetyPerimeter)}

- Complex multi-objectives design involves **multiple** trade-off analysis cases
- What is the relation between the cases, when they have **shared variables**?
- Need a way to model and analyse the coupling.
- Machine-readable modelling of shared variables in a physical process or in software is a challenge.

Relations Usage

It is useful to integrate the multiple model elements into a single matrix representation (such as ROSETTA)

- With various relations defined, one could start to use these relations a specific analysis case to derive further relations for understanding the complexity



«analysis»
relationDerivation

objective
doc derive relations between objective spaces, radarPerformance and radarSafety

calcs

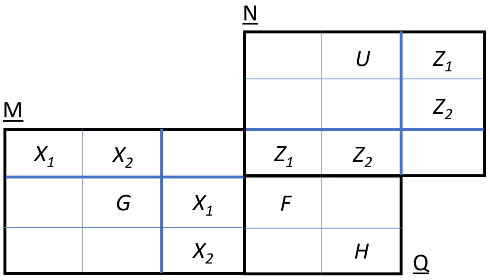
```
:>> relationDerivation {
        in knowledgeMap F
        in knowledgeMap H
        in knowledgeMap G
        return knowledgeMap U = H ◦ G ◦ inverse(F)
}
```

$F: X_1 \rightarrow Z_1$

$H: X_2 \rightarrow Z_2$

$G: X_1 \rightarrow X_2$

$U: Z_1 \rightarrow Z_2$



Relational transformation (of the relation G):

(X_1, X_2) with (X_1, Z_1) and (X_2, Z_2) → $\exists U$, $(Z_1, Z_2) = U$

The new relation *identified* is denoted as *U*, which is *defined* by the chain of relations (Z_1, X_1), (X_1, X_2), and (X_2, Z_2). Thus, *U* is defined by the composition,

$$U = H \circ G \circ F^{-1}.$$

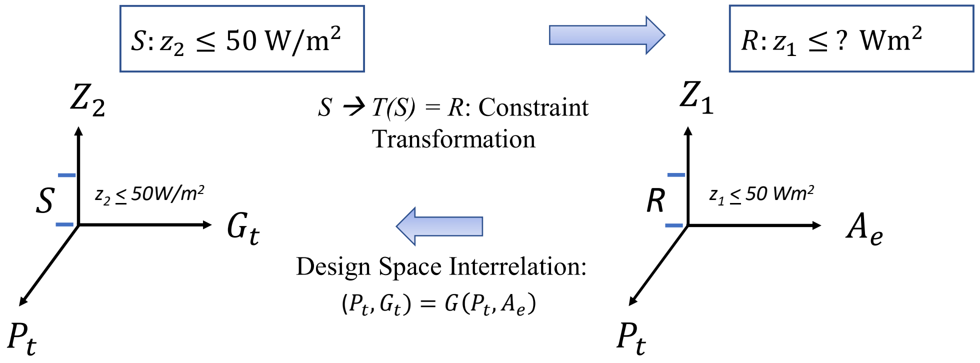
Topics of the Presentation

- Structuring Problems at a Higher Level of Abstraction
- Definition, Usage, and Analysis in SysMLv2
- ***Category Theory as a Language for Architecture, Analysis and Design***
- Conclusions: Complementing SysMLv2 with Category Theory

Models can be structured using the language of category theory.

Engineering Representation of a Constraint Transformation T

***One 'value'** of the transformation of the safety requirement S constrains power-aperture product R to:* $z_1 \leq 50 Wm^2$.



$S: z_2 \leq 50\ \mathrm{W/m^2}$

$R: z_1 \leq ?\ \mathrm{Wm^2}$

$S \rightarrow T(S) = R$: Constraint Transformation

Design Space Interrelation:
$(P_t, G_t) = G(P_t, A_e)$

Safety Design Space: $X_2 = (P_t, G_t)$

Performance Design Space $X_1 = (P_t, A_e)$

At the safety perimeter of d meters,
power density z_2 *must be* $\leq 50W/m^2$.

*What is **the general form** of the*
constraint transformation T: $Z_2 \rightarrow Z_1$ *?*

Matrix View of a Relational Structure for Safety Analysis:
ROSETTA Framework when the objectives are *dependent* variables

There are three *defined relations*,

$F: X_1 \rightarrow Z_1$
$H: X_2 \rightarrow Z_2$
$G: X_1 \rightarrow X_2$

There is one ***implied relation***,

$U(G): Z_1 \rightarrow Z_2$,

which is identified using *relational transformation*.

The matrices of a ROSETTA framework use standard row-column notation, e.g., (Z_1, X_1) is the relation F, but they have a different algebra than ordinary matrices.

ROSETTA: Relational Oriented Systems Engineering Technology Trade-off and Analysis

N (Target)

M (Source)

				$U(G)$	Z_1
					Z_2
X_1	X_2		Z_1	Z_2	
	G	X_1	F		
		X_2		H	

Q (Transformation)

Algebra of the relational transformation (of the relation G):

(X_1, X_2) with (X_1, Z_1) and $(X_2, Z_2) \rightarrow \exists U$, $(Z_1, Z_2) = U(G)$

The new relation *identified* is denoted as U and is *defined* by the chain of relations (Z_1, X_1), (X_1, X_2), and (X_2, Z_2). Thus, the relation $(Z_1, Z_2) = U(G)$ is defined by the composition,

$$U = H \circ G \circ F^{-1}.$$

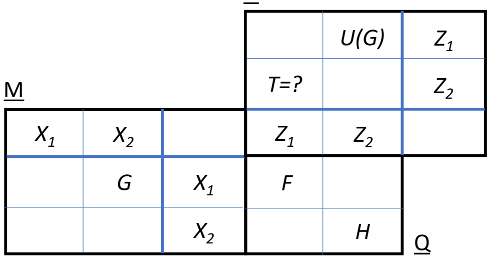
Comparison of Relational Transformation of *G* using ROSETTA with Relational Composition in *Rel*

There are four *defined or implied* relations,

$F: X_1 \rightarrow Z_1$	(X_1, Z_1)
$H: X_2 \rightarrow Z_2$	(X_2, Z_2)
$G: X_1 \rightarrow X_2$	(X_1, X_2)
$U: Z_1 \rightarrow Z_2$	(Z_1, Z_2)

Relational Transformation: Graphical View

$$\begin{tikzcd}
 & Z_1 \arrow[dl, "F^{-1}"'] \arrow[d, dashed, "G \circ F^{-1}"] & {} \arrow[dd, "U"] \\
X_1 \arrow[r, "G"] \arrow[dr, dashed, "H \circ G"'] & X_2 \arrow[d, "H"] & \\
 & Z_2 & {}
\end{tikzcd}$$



			N		
				U(G)	Z_1
M			T=?		Z_2
X_1	X_2		Z_1	Z_2	
	G	X_1	F		
		X_2		H	Q

The relational structure of the framework is complete.

The relational transformation of (X_1, X_2) with (X_1, Z_1) and (X_2, Z_2) implies *a chain of relations* (Z_1, X_1), (X_1, X_2), and (X_2, Z_2). Thus,

$$\boxed{U = U(G) = H \circ G \circ F^{-1}}$$

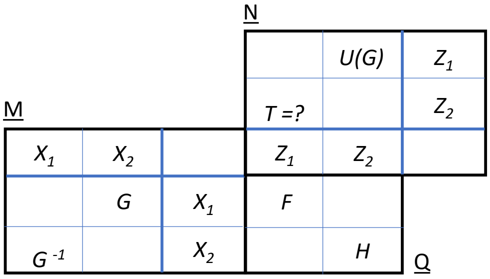
The *problem is structured* by joining up the arrows, but the transformation *T* does not yet belong to the framework.

Matrix View of a Relational Structure for Safety Analysis:

Structured Solution for *T* using the ROSETTA Framework

There are six *defined or implied* relations,

$F: X_1 \rightarrow Z_1$	(X_1, Z_1)
$H: X_2 \rightarrow Z_2$	(X_2, Z_2)
$G: X_1 \rightarrow X_2$	(X_1, X_2)
$G^{-1}: X_1 \rightarrow X_2$	(X_2, X_1)
$U: Z_1 \rightarrow Z_2$	(Z_1, Z_2)
$T: Z_2 \rightarrow Z_1$	(Z_2, Z_1)



			N		
				U(G)	Z_1
M			T =?		Z_2
X_1	X_2		Z_1	Z_2	
	G	X_1	F		
G^{-1}		X_2		H	Q

Formulae can be expressed in SysMLv2.

$$\boxed{\begin{aligned} &F\,(P_t, A_e) = z_1 = P_t\,A_e \\ &H\,(P_t, G_t) = z_2 = P_t\,G_t/4\pi d^2 \\ &G\,(P_t, A_e) = (P_t, 4\pi A_e/\lambda^2) \end{aligned}}$$

Relational transformation (of the relation G^{-1}):

(X_2, X_1) with (X_2, Z_2) and $(X_1, Z_1) \rightarrow \exists T, (Z_2, Z_1) = T(G^{-1})$

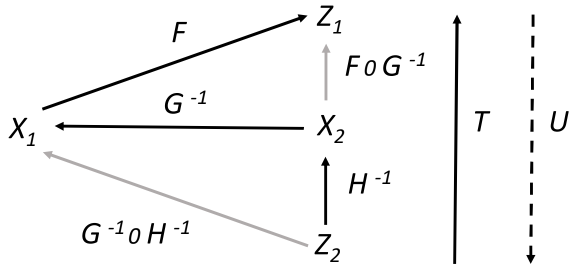
The new relation *identified* is denoted as *T* and is *defined* by the chain of relations (Z_2, X_2), (X_2, X_1), and (X_1, Z_1). Thus, *T* is defined by the composition,

$$\boxed{T = H^{-1} \circ G^{-1} \circ F}$$

Comparison of Relational Transformation *T* of G^{-1} using ROSETTA with Relational Composition in *Rel*

$F\,(P_t\,,A_e) = z_1 = P_t\,A_e$
$H\,(P_t\,,G_t) = z_2 = P_t\,G_t/4\pi d^2$
$G\,(P_t\,,A_e) = (P_t\,,4\pi A_e\,/\lambda^2)$

Relational Transformation: Graphical View



Derivation of T from U(G)

N

M

				U(G)	Z_1
			U^{-1}		Z_2
X_1	X_2		Z_1	Z_2	
	G	X_1	*F*		
		X_2		*H*	

Q

The structured solution for T can also be derived from U(g):

$U = H_0\,G_0\,F^{-1} \rightarrow T = U^{-1} = (H_0\,G_0\,F^{-1})^{-1} = F_0\,G^{-1}{}_0\,H^{-1}.$

Either way, as seen in the algebraic calculations in the annex, when the formulae for *F*, *G*, and *H* are used, a formula for *T* then follows,

$$\boxed{z_1 = T\,(z_2) = d^2\lambda^2 z_2.}$$

For a perimeter with $d\lambda = 1$, $z_2 \leq 50W/m^2 \rightarrow z_1 \leq 50Wm^2$.

---

*Engineering solution and optimisation using constraint driven design**

Requirements Transformation (ISO/IEC/IEEE 15288) using ROSETTA

OMG UPR 1.0

ISO/IEC/IEEE 15288 Requirements Transformation

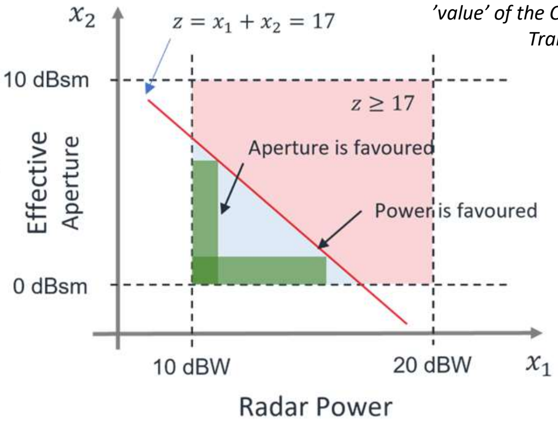
Knowledge Map (Physics)

$$\begin{tikzcd} R \arrow[r, phantom, "\subseteq"] \arrow[d, dashed, "f^{-1}"'] & Z \\ V \arrow[r, phantom, "\subseteq"] & X \arrow[u, "f"'] \end{tikzcd}$$

$V = f^{-1}\,(R \cap f\,(X))$

Representation in the design space X = $(x_1\,, x_2)$ = [10 dBW, 20, dBW] x [0 dBsm, 10 dBsm]
Transformation of requirements (in decibels) defines the solution set V (x_1, x_2), *(blue triangle)*:
Power aperture product = z = f $(x_1, x_2) = x_1 + x_2$
$f^{-1}\,(R \cap f\,(X)) = f^{-1}\,([0, 17] \cap [10, 30])$
$= f^{-1}\,([10, 17]$
$z \in R = [10, 17] \rightarrow x_1 + x_2$ is a solution
Maximising *z* (dBWsm) *(green rectangles)* is constrained by the *safety* requirement.

Constraint Driven Design for Radar Safety Analysis



$x_1 + x_2 \leq 17$ is just one 'value' of the Constraint Transformation

*Adapted from the Loughborough University WS66 System Design MSc module.

Conclusions (1 of 2): Problem Solving in Category Theory

Modeling, Analysis and Design using Structures

Transform a Use Case predicate into a morphism P in *Rel*.

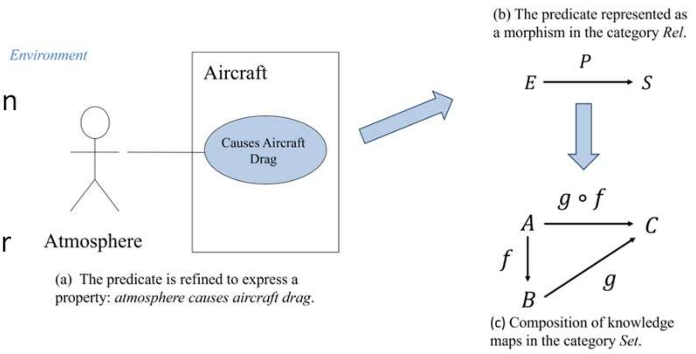
Express domain knowledge models in *Set* to quantify P as a property.

Mathematical models are needed, e.g. to compose relations to solve for drag, *c*, as a function of altitude, *a*,

$$f: A \rightarrow B$$
$$g: B \rightarrow C$$

$c = g \circ f(a) = (1 - a/10)c_0$

These support *analysis and design*.[1]



(a) The predicate is refined to express a property: *atmosphere causes aircraft drag*.

(b) The predicate represented as a morphism in the category *Rel*.

(c) Composition of knowledge maps in the category *Set*.

Model specification and transformation in Rel.

---

[1] *For further details refer to [1], A Brief Introduction to Category Theory for Systems and Software Engineers.*

Conclusions (2 of 2)

Applications of Category Theory and Why Should OMG Care?

The Category of Relations (Rel)

Offers a language for direct transformation of models into structured solutions.

Can be a *useful language of architecture* for system modeling and analysis.

Categoric theoretic diagrams have promising associations with SysMLv2.

But must be integrated with domain knowledge for *engineering practice*.

Promises intuitive rigorous structured methods for integrating SysMLv2 analysis cases.

Way ahead:

Continue discussions with SE DSIG about future SysML v2.1 opportunities

Investigate further applications to engineering problems and practice and …

Category theoretic foundation of object orientation and software applications

Questions?

Annexes

A-1 Algebraic Solution for the Constrain Transformation

A-2 References

A-1 Algebra of the Constraint Transformation Relational Structure (in dB)

(1) $z_1 = x_1 + x_2$ *Power Aperture Product*

(2) $z_2 = x_1 + x_3 - r$ *Power Density at safety perimeter (d)*

(3) $0 = x_2 - x_3 + k$ (this derives from $x_3 = x_2 + k$)

$r = 4\pi d^2$ $x_1 = P_t$ in dB ($x_1 = x_{11} = x_{21} = P_t$)

$k = 4\pi/\lambda^2$ $x_2 = \mathrm{A_e}$ in dB ($x_2 = x_{12} = \mathrm{A_e}$)

$x_3 = \mathrm{G_t}$ in dB ($x_3 = x_{22} = \mathrm{G_t}$)

$(x_{11}, x_{12}) \in X_1$ and $(x_{21}, x_{22}) \in X_2$ $P_t = x_1 = x_{11} = x_{21}$ is a shared attribute

---

A-1 (continued) Solution for the Transformation by Algebraic Methods

Add equations (2) and (3) to obtain,

(2) + (3): $z_2 = x_1 + x_2 - r + k$

Substitute (1), $x_1 + x_2 = z_1$, into this to obtain,

$z_2 = z_1 - r + k \rightarrow z_1 = z_2 + r - k$ in dB

In SI units we then have,

$z_1 = (z_2)\,(4\pi d^2)/(4\pi/\lambda^2\,) = z_2\, d^2\lambda^2$

This defines the constraint transformation as,

$$\boxed{z_1 = T(z_2) = d^2\lambda^2\, z_2 = 50Wm^2 = 17dBWsm \text{ for } \lambda d = 1.}$$

This agrees with both OMG results [2] and a forthcoming paper.

References

[1] *A Brief Introduction to Category Theory for Systems and Software Engineers, September 2024,* on the Mathsig homepage https://www.omg.org/maths/ , or the AI PTF. A recording of the presentation can be found at, https://www.omg.org/cgi-bin/doc.cgi?ai/2024-09-06

[2] *ROSETTA Implementation in SysMLv2 Revisited, June 2024,* on the Mathsig homepage https://www.omg.org/maths/

[3] *Implementation of ROSETTA in SysMLv2 and Underlying Mathematical Formalisms, March 2023, in the Mathsig archives https://www.omg.org/maths/*

[4] https://www.omg.org/spec/UPR/1.0/About-UPR/

[5] Stanford Encyclopedia of Philosophy, "Category Theory", 29 August 2019. Available online at

https://plato.stanford.edu/entries/category-theory/

[6] C. E. Dickerson, M.K. Wilkinson et al., "Architecture definition in complex system design using model theory," IEEE Syst. J., vol. 15, no. 2, pp. 1847–1860, June 2021.

[7] Dickerson, C.E. and Wilkinson, M.K., "Architecture, Analysis and Design of Systems Using Extensions of Category Theory", *IEEE Open Journal of Systems Engineering*, 23 July 2024. Digital Object Identifier 10.1109/OJSE.2024.3432570. Available open source online at,

Architecture, Analysis, and Design of Systems Using Extensions of Category Theory