



Making Model-Based System Design Aware of Design Requirements: Challenges in Using and Extending SysML

Mara Nikolaidou, Christos Kotronis
System Engineering Group

System Engineering Group

- Dimosthenis Anagnostopoulos, Professor
- Cleopatra Bardaki, Assistant Professor
- Christos Michalakelis, Associate Professor
- *Mara Nikolaidou*, Professor
- Vassilis Dalakas, Research Assistant
- Anargyros Tsadimas, Research Assistant
- George-Dimitrios Kapos, Post Doc
- *Christos Kotronis*, *Post Doc*
- Giannis Routis, PHD student
- Dimitrios Gkoulis, PHD student

Outline

- Background on MBSD and SysML
- Proposed approach on MBSD
- Athens Metro case study and demo
- Remote Patient Monitoring case study and demo
- Discussion

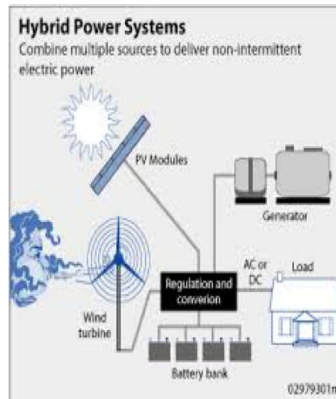


Background

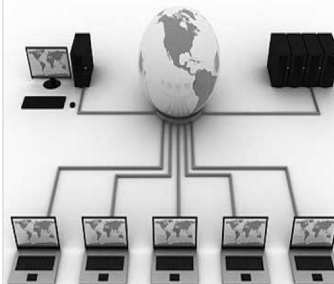
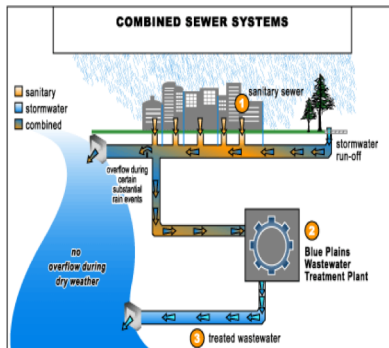
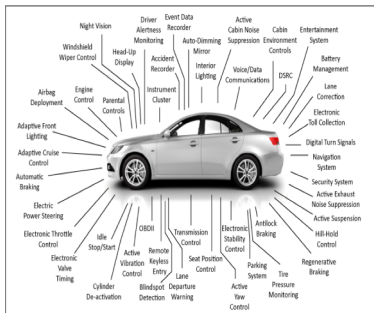
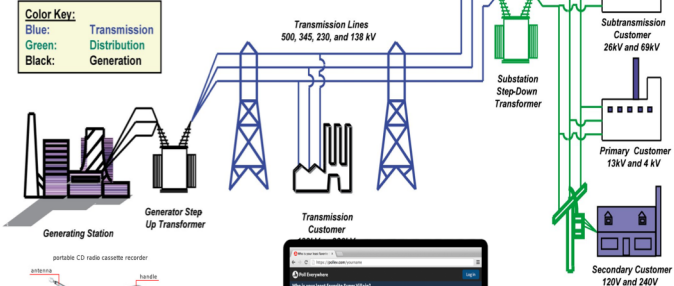
The concept of SoS

Systems of Systems

Hierarchies of independent and interactive sub-systems



Basic Structure of the Electric System



SoS Design is complicated

- Different stakeholders have a different concern to explore



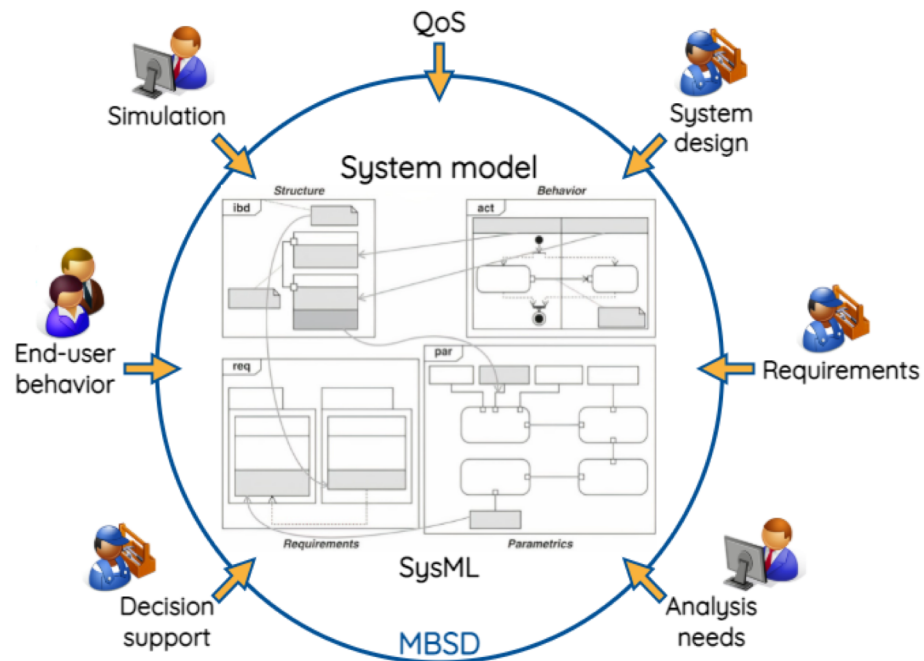
SoS Design is complicated

- Different stakeholders have a different concern to explore
- Different tools and methodologies should be integrated
- The System Engineer gets a headache
- Becomes too complex to be effectively performed



Model-Based Systems Design (MBSD)

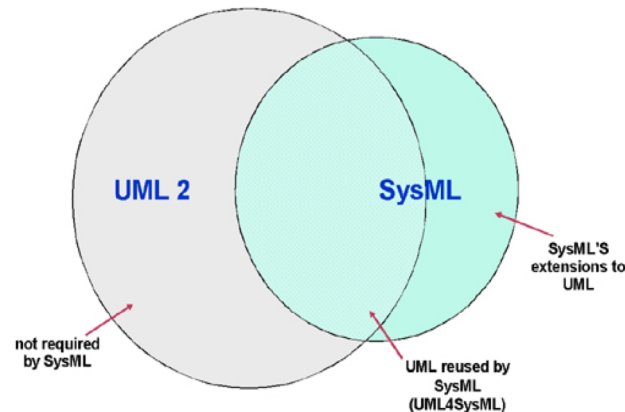
- A single model throughout the whole process
- Diverse pieces of information (requirements and specifications) are combined as distinct views
- Seamless transition to the next phase
- Lowers cost of changes to preceding activities
- A comprehensive perspective of the system is developed and is available at any time



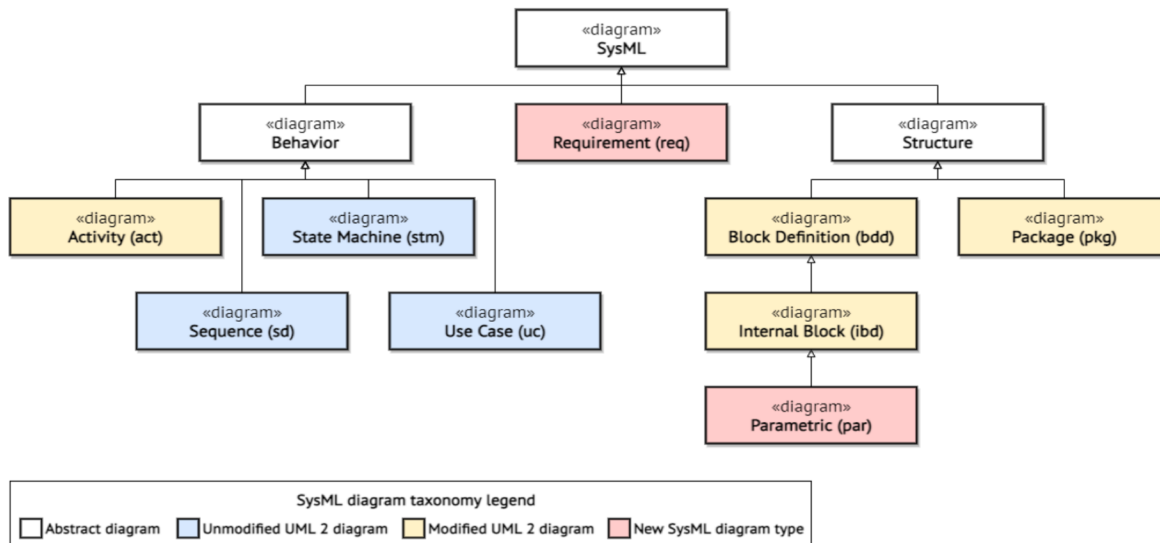
Systems Modeling Language (SysML)

- A OMG Standard established in 2010
- Current version 1.7 (under publication)
- Defined as a UML extension (a profile)

to become for system engineering as popular as UML for software engineering



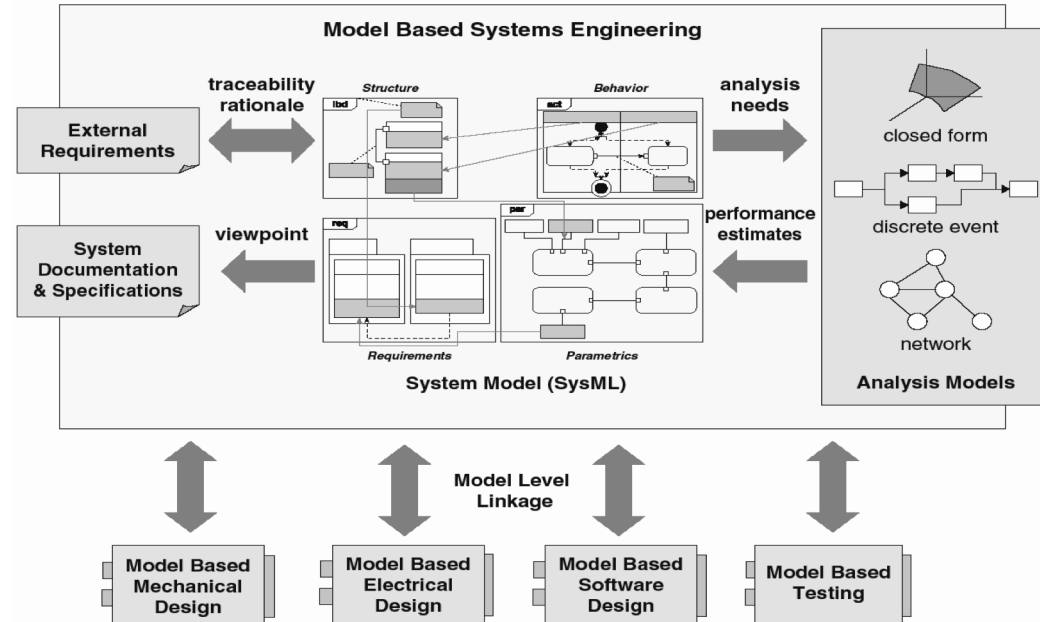
Systems Modeling Language (SysML)



- Facilitates MBSE of SoSs
 - Information regarding the systems is defined in a standardized fashion
- SysML is extendable
 - Profiles is the mean to extend/restrict SysML functionality
 - Domain-specific languages (DSLs) may be defined to target specific domains needs

How to use SysML for MBSD?

- Provide an integrated MBSD framework¹
 - for system design
 - supporting design activities
 - exploring system behavior
- Advocate integration of SysML system models with external models
 - Not all engineering activities can be performed with SysML (e.g. analysis)



¹“System model as a framework for analysis and traceability”, presented in the work of Friendenthal et al.
S. Friedenthal, A. Moore, and R. Steiner, A practical guide to SysML: the systems modeling language. Morgan Kaufmann, 2014

How to use SysML for MBSD?

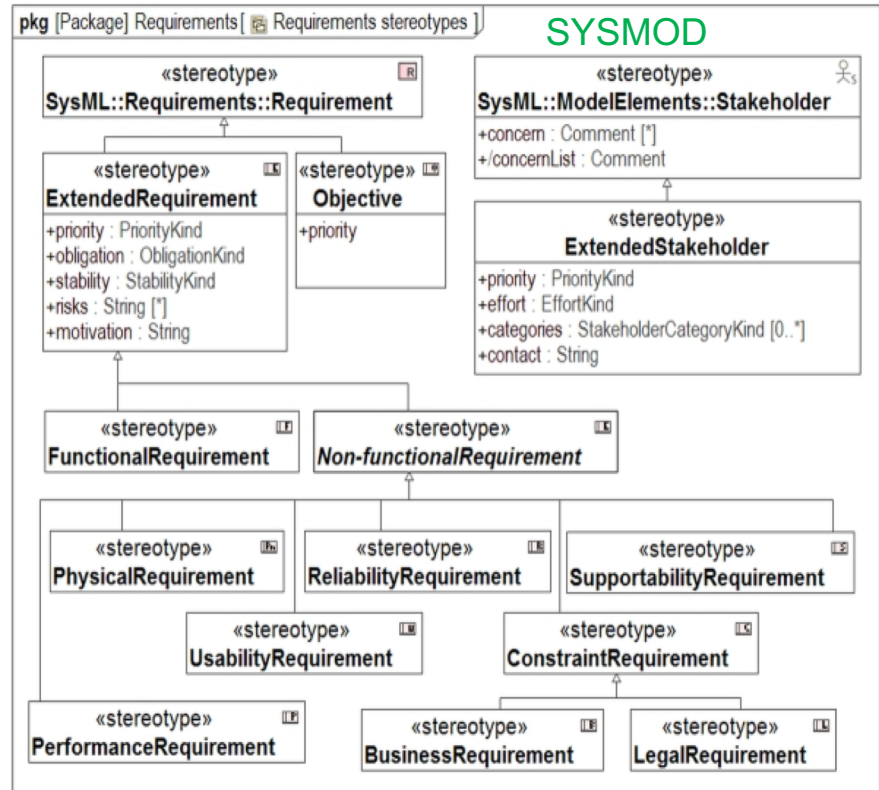
- Frameworks / approaches are needed to specialize the support of specific stakeholders or system categories¹
 - Systems Modeling Toolbox (SYSMOD)²
 - Unified Architecture Framework (UAF)³
 - Lots of profiles are developed, while there is a need to integrate them

	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Interaction Scenarios Is	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Metadata Md	Metadata Taxonomy Md-Tx	Architecture Viewpoints ^a Md-Sr	Metadata Connectivity Md-Cn	Metadata Processes ^a Md-Pr	-	-	Conceptual Data Model, Logical Data Model,	Environment Pm-En	Metadata Constraints ^a Md-Ct		Metadata Traceability Md-Tr
Strategic St	Strategic Taxonomy St-Tx	Strategic Structure St-Sr	Strategic Connectivity St-Cn	-	Strategic States St-St	-			Strategic Constraints St-Ct	Strategic Deployment, St-Rm Strategic Phasing St-Rm	Strategic Traceability St-Tr
Operational Op	Operational Taxonomy Op-Tx	Operational Structure Op-Sr	Operational Connectivity Op-Cn	Operational Processes Op-Pr	Operational States Op-St	Operational Interaction Scenarios Op-Is			Operational Constraints Op-Ct	-	-
Services Sv	Service Taxonomy Sv-Tx	Service Structure Sv-Sr	Service Connectivity Sv-Cn	Service Processes Sv-Pr	Service States Sv-St	Service Interaction Scenarios Sv-Is			Service Constraints Sv-Ct	Service Roadmap Sv-Rm	Service Traceability Sv-Tr
Personnel Pr	Personnel Taxonomy Pr-Tx	Personnel Structure Pr-Sr	Personnel Connectivity Pr-Cn	Personnel Processes Pr-Pr	Personnel States Pr-St	Personnel Interaction Scenarios Pr-Is	Physical schema, real world results	Measurements Pm-Me	Competence, Drivers, Performance Pr-Ct	Personnel Availability, Personnel Evolution, Personnel Forecast Pr-Rm	Personnel Traceability Pr-Tr
Resources Rs	Resource Taxonomy Rs-Tx	Resource Structure Rs-Sr	Resource Connectivity Rs-Cn	Resource Processes Rs-Pr	Resource States Rs-St	Resource Interaction Scenarios Rs-Is			Resource Constraints Rs-Ct	Resource evolution, Resource forecast Rs-Rm	Resource Traceability Rs-Tr
Security Sc	Security Taxonomy Sc-Tx	Security Structure Sc-Sr	Security Connectivity Sc-Cn	Security Processes Sc-Pr	-	-			Security Constraints Sc-Ct	-	-
Projects PJ	Project Taxonomy PJ-Tx	Project Structure PJ-Sr	Project Connectivity PJ-Cn	Project Activity PJ-Pr	-	-			-	Project Roadmap PJ-Rm	Project Traceability PJ-Tr
Standards Sd	Standard Taxonomy Sd-Tx	Standards Structure Sd-Sr	-	-	-	-	-	Standards Roadmap Sd-Rm	Standards Traceability Sd-Tr		
Actuals Resources Ar		Actual Resources Structure, Ar-Sr	Actual Resources Connectivity, Ar-Cn	Simulation ^b					Parametric Execution/ Evaluation ^b	-	-
Dictionary * Dc											
Summary & Overview SmOv											
Requirements Rq											

1. Wolny, S. et al. (2020). Thirteen years of SysML: a systematic mapping study. SSM, 19(1), 111-16
2. Weillkiens, T. (2016). SYSMOD-The systems modeling toolbox-pragmatic MBSE with SysML. Lulu. com.
3. Hause, M. et al. (2017). Technology update on the unified architecture framework (UAF). INSIGHT, 20(2), 71-78.

How to use SysML for MBSD?

- Frameworks / approaches are needed to specialize the support of specific stakeholders or system categories¹
 - Systems Modeling Toolbox (SYSMOD)²
 - Unified Architecture Framework (UAF)³
 - Lots of profiles are developed, while there is a need to integrate them (some become part of the standard)



1. Wolny, S. et al. (2020). Thirteen years of SysML: a systematic mapping study. *SSM*, 19(1), 111-169.

2. Weilkens, T. (2016). SYSMOD-The systems modeling toolbox-pragmatic MBSE with SysML. Lulu. com.

3. Hause, M. et al. (2017). Technology update on the unified architecture framework (UAF). *INSIGHT*, 20(2), 71-78.

Obstacles in SysML adoption

- Engineers are skeptical towards SysML
- Main reasons *
 - Lack of KPIs to promote its application.
 - It is still too expensive and unmanageable.
 - Lack of training
 - There is still no easy-to-follow, user-oriented methodology to allow its use.
 - Domain-specific profiles are not widely available
 - **Requirement management and Virtual Verification/Validation are characterized as critical issues to be managed.**

According to a study performed (35 companies: aerospace, vehicle, plant) by INCOSE German chapter

* System Engineering in Industrial Practice, Sven-Olaf Schulze et.al

https://www.hni.uni-paderborn.de/fileadmin/Fachgruppen/Seniorprofessur_Gausemeier/systemsengineerings/20150706_SE-Studie_enGB_Einzelseiten.pdf

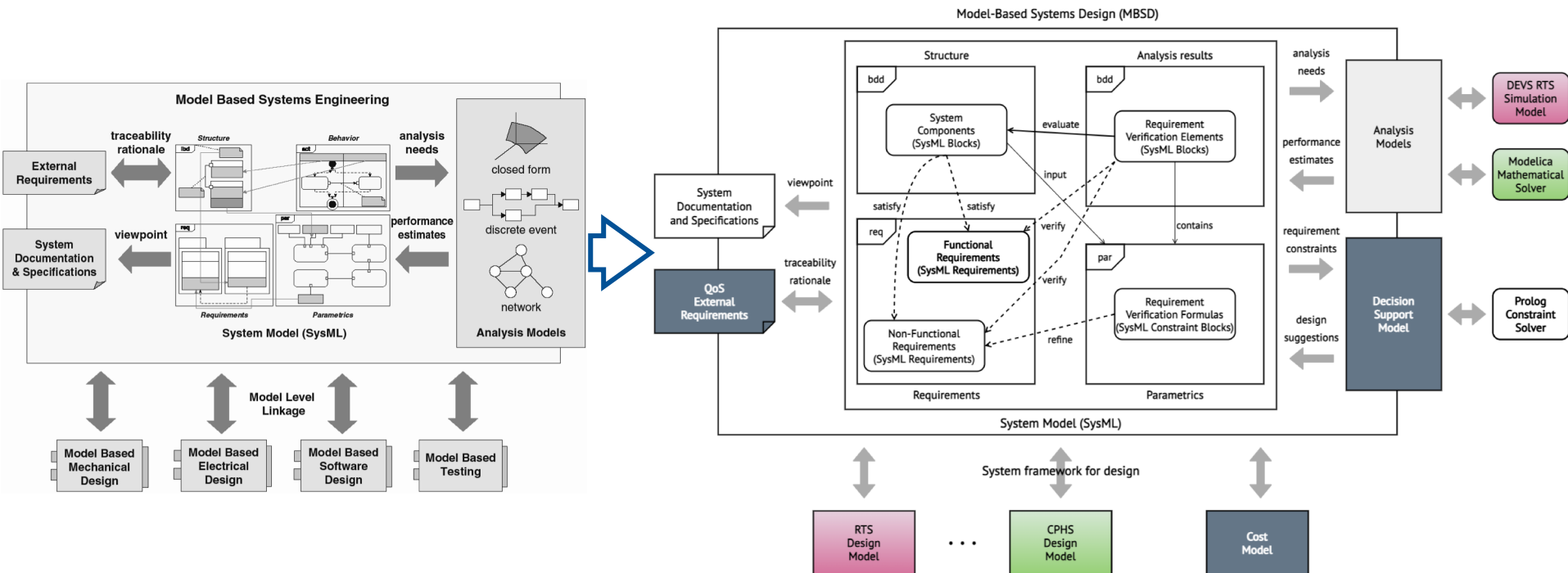
What can we do to smooth those obstacles?

- Hide complexity to make the whole process more manageable
 - Provide a single point of communication
 - Hide interaction with external models/tools and integrate results into the SysML model
- Promote automation
 - to help the designer in tedious (e.g. repeatable computations) or not that tedious (e.g. decision making) tasks
- Focus on requirement analysis and verification in system design
 - Non-functional (performance, quality, cost) are nasty ones
 - Quantitative requirement verification requires extensive computations and external analysis tools (for example simulation)
- Provide re-usable profiles for specific domains or tasks
 - They usually work well with a specific modeling tool, the one they are built for
 - Provide instructions on how to apply them



Managing non-functional requirements in MBSD using SysML

Extend MBSD SysML framework by Friedenthal et al.



⇒ Extend framework for QoS-aware MBSD to determine if system design can serve QoS.

Objective of this effort

⇒ Two_fold

- Hide complexity from the system designer by promoting automation and external tool integration
- Support non-functional, quantitative requirement exploration in MBSD
- Provision of reusable domain-specific profiles
 - For a specific modeling environment (MagicDraw Product Family)
- Compliance with existing frameworks
 - SYSMOD and UAF



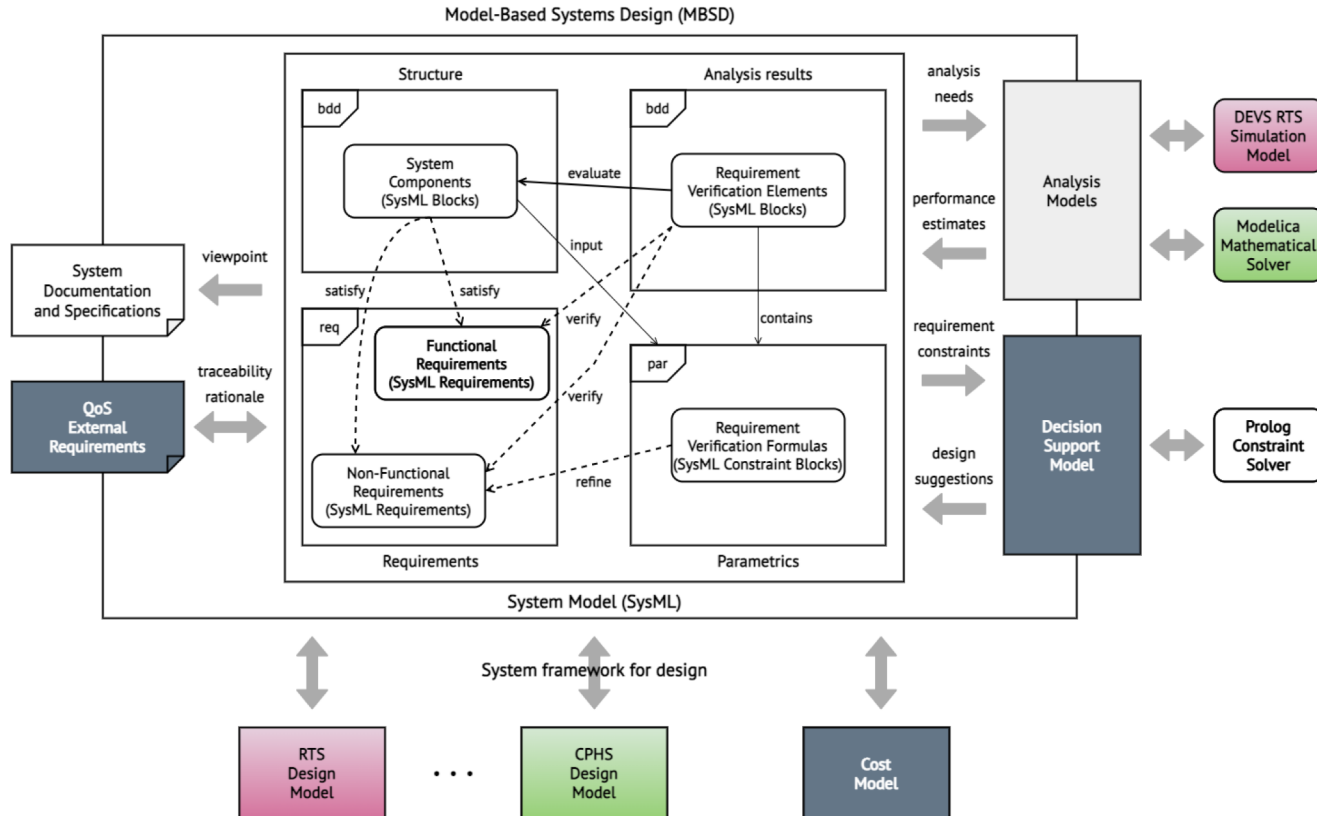
Design & development QoS-aware MBSD SysML framework

Provided Functionality

- Description and management of QoS
 - provide SysML requirement entity extensions (SysML profiles)
- Integration of QoS analysis models into system model
 - compute provided QoS based on results obtained via simulations or mathematical solvers (external tool integration- plugins)
- Verification QoS requirements graphically
 - check whether they can be met by the system (plugins)
- Proposal of system design solutions
 - taking into account the requirements (external tool integration- plugins)

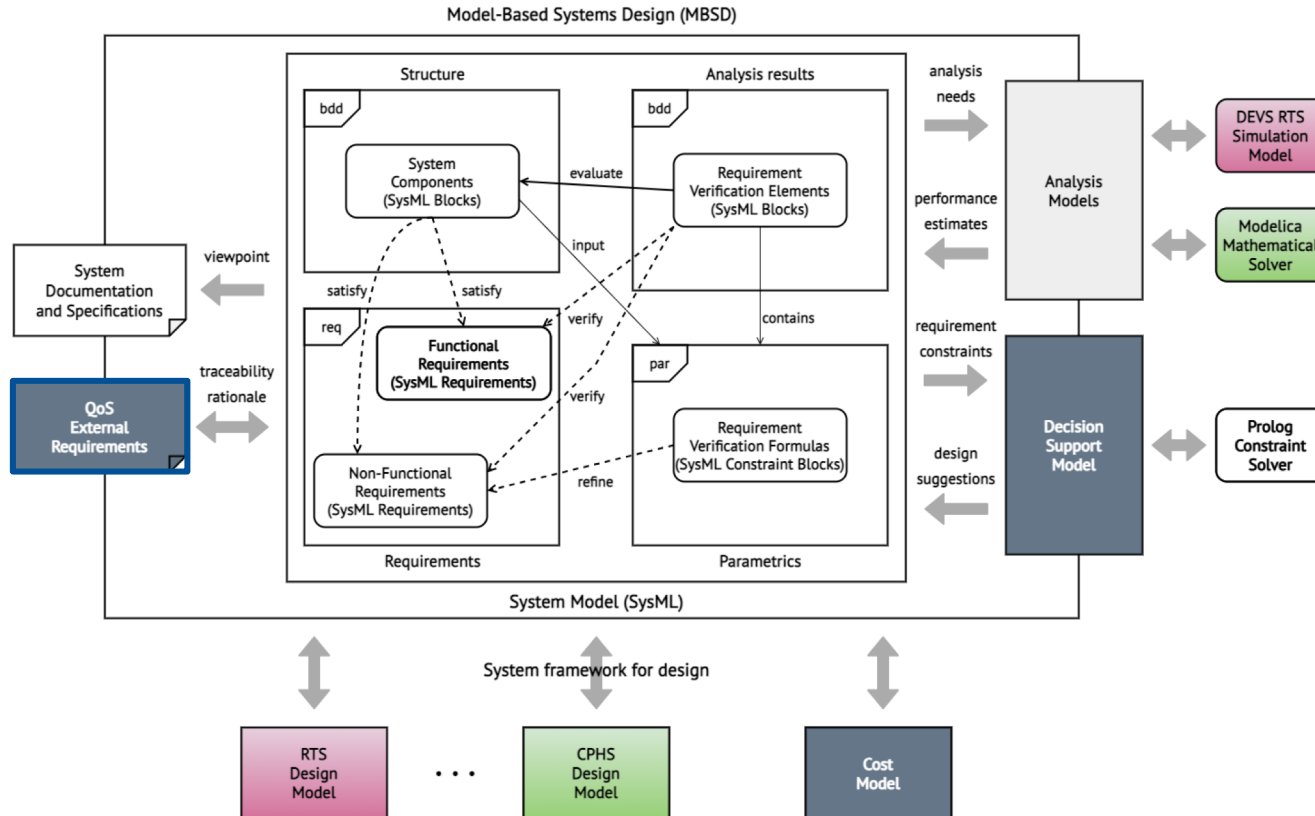
The framework was applied into two different domains, Railway Transportation Systems (RTS) and Cyber-Physical Human System (CPHS) (through specific SysML profiles).

Integrated QoS-aware MBSD SysML framework

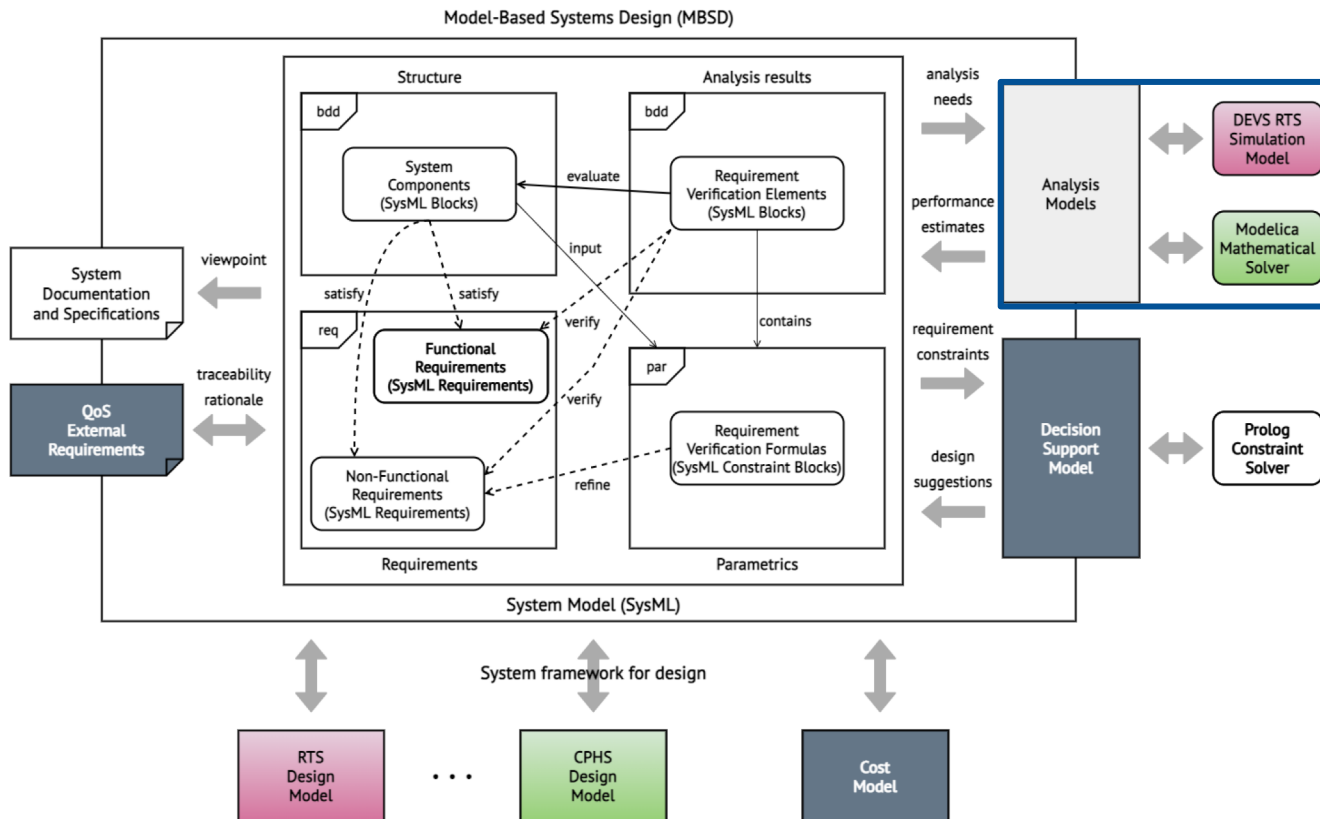


Integrated QoS-aware MBSD SysML framework

QoS requirements need to be specified, expressed, and integrated into system model.

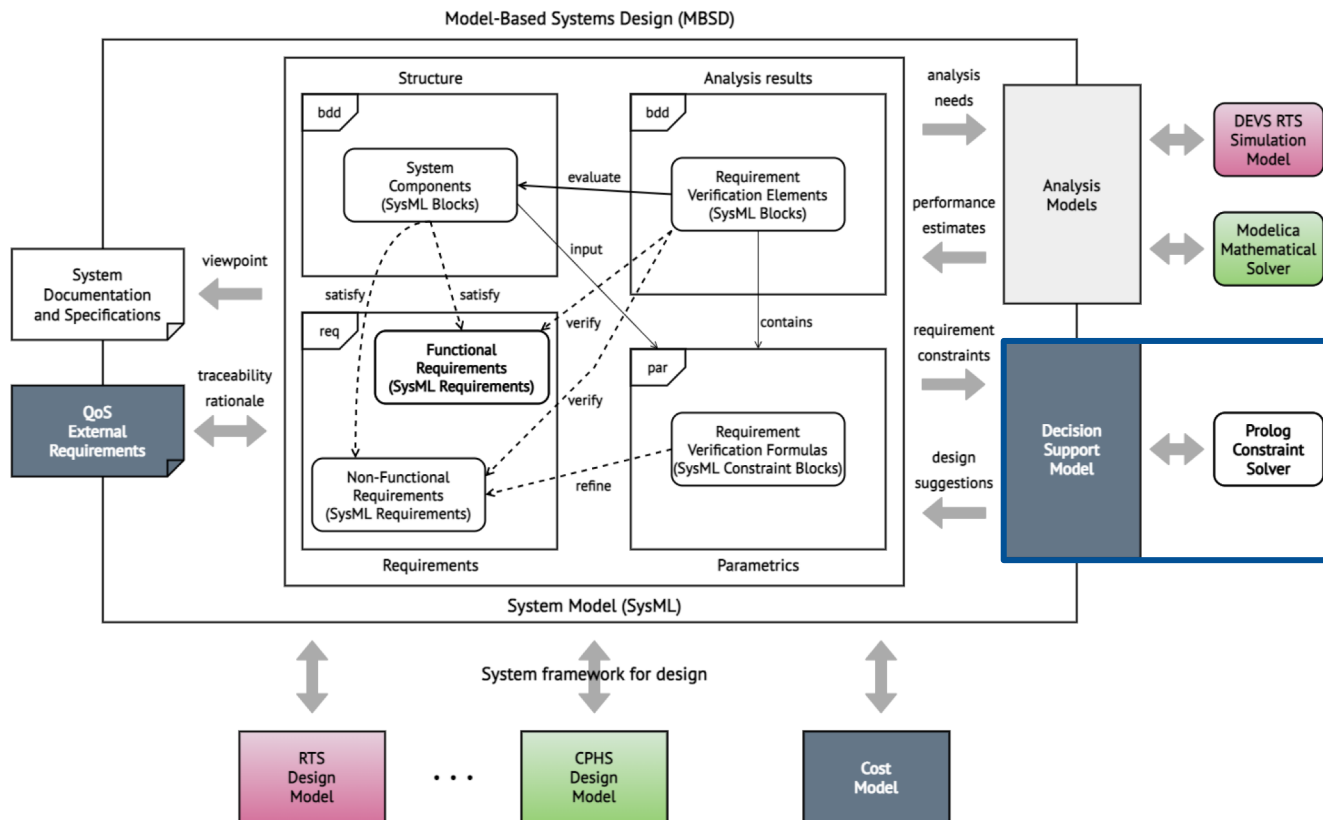


Integrated QoS-aware MBSD SysML framework



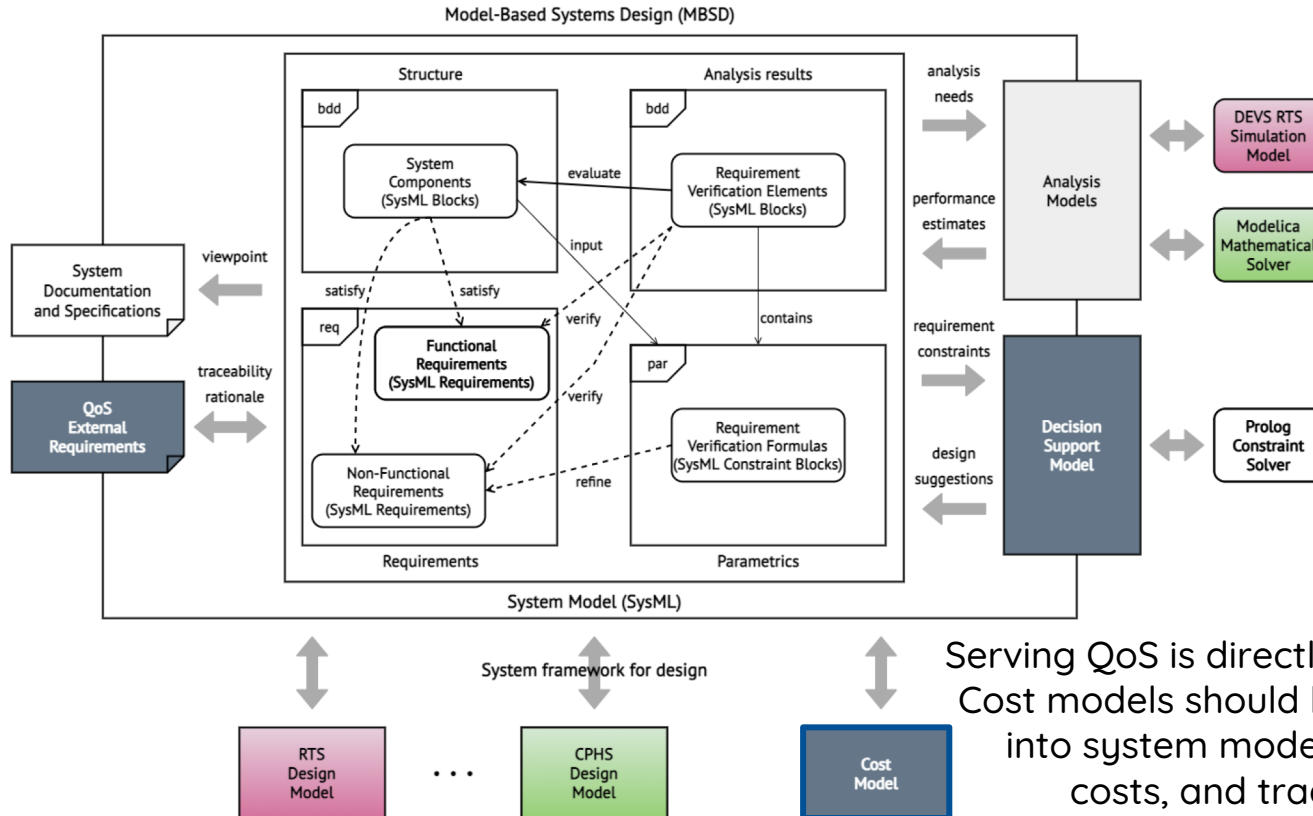
Analysis models are required to assess whether associated requirements are satisfied (or not).

Integrated QoS-aware MBSD SysML framework

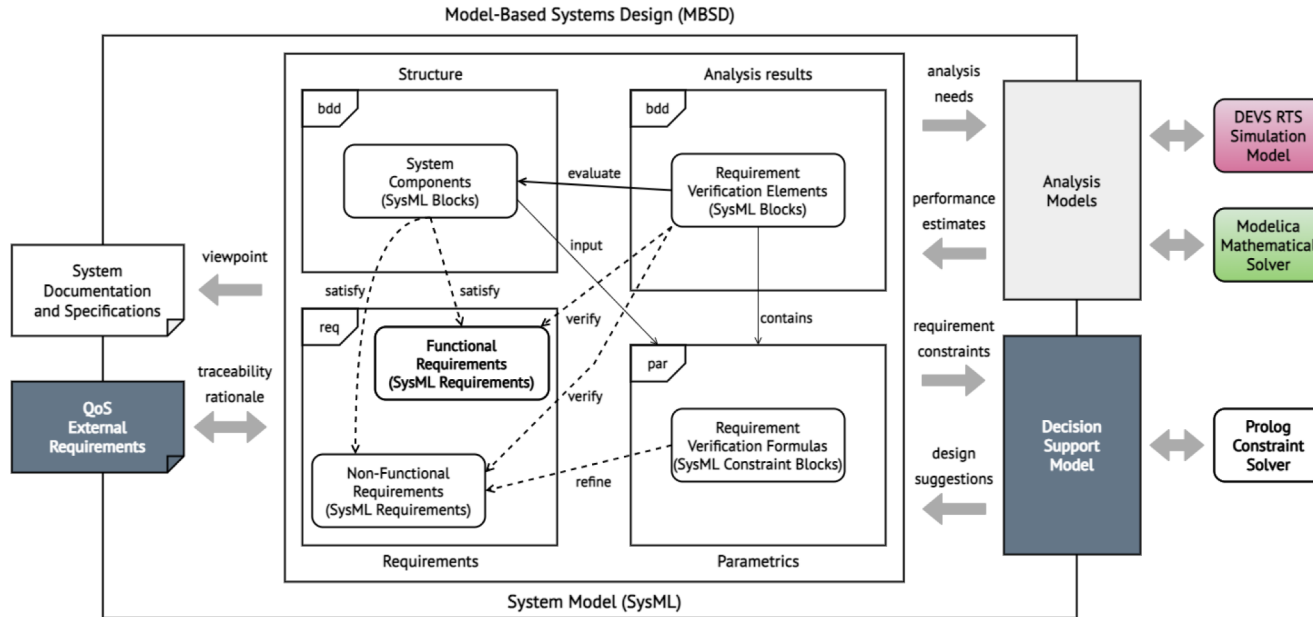


Satisfying QoS requirements requires the designer to decide upon different key aspects like system configuration.

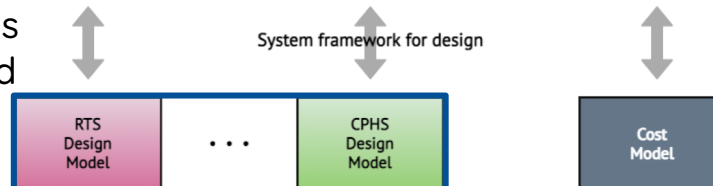
Integrated QoS-aware MBSD SysML framework



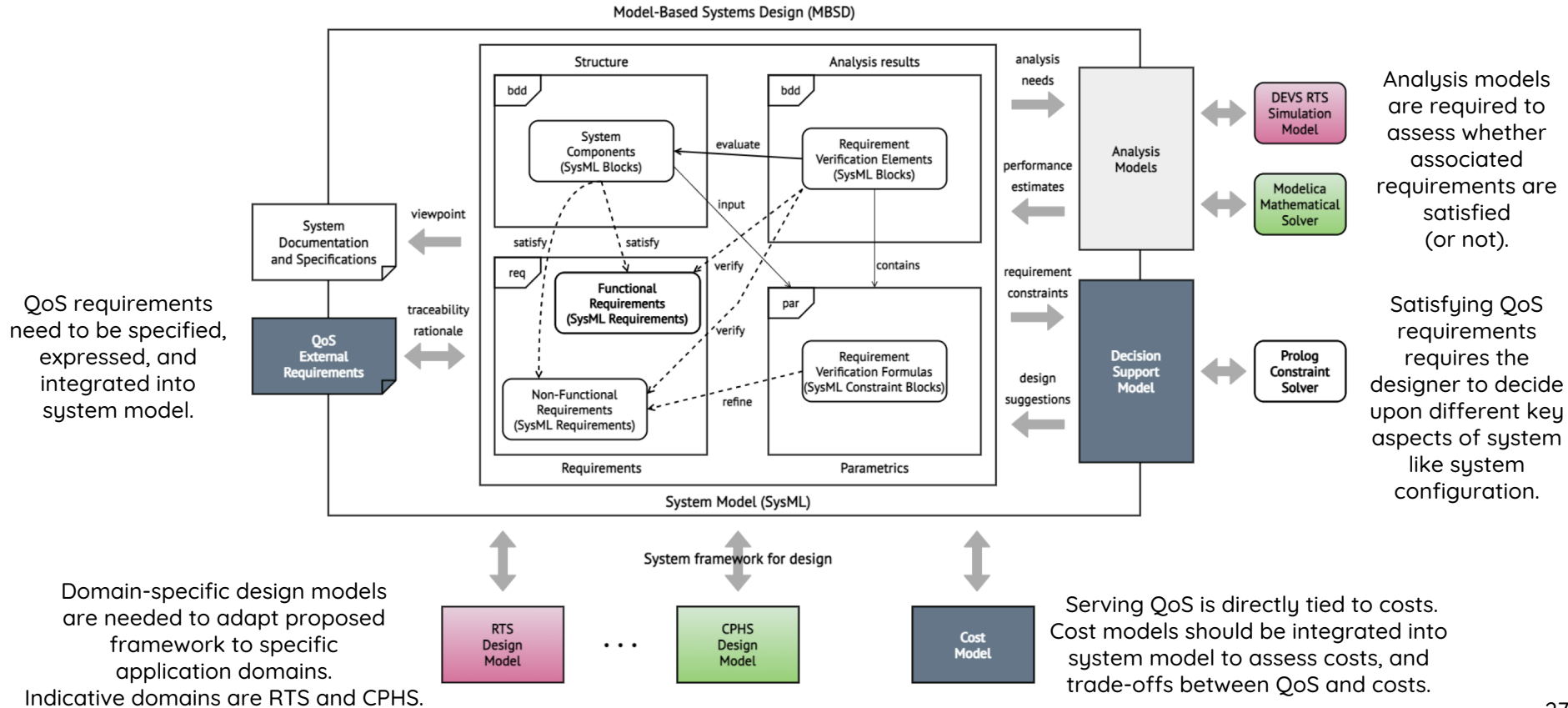
Integrated QoS-aware MBSD SysML framework



Domain-specific design models are needed to adapt proposed framework to application domains. Indicative domains are RTS and CPHS.



Integrated QoS-aware MBSD SysML framework

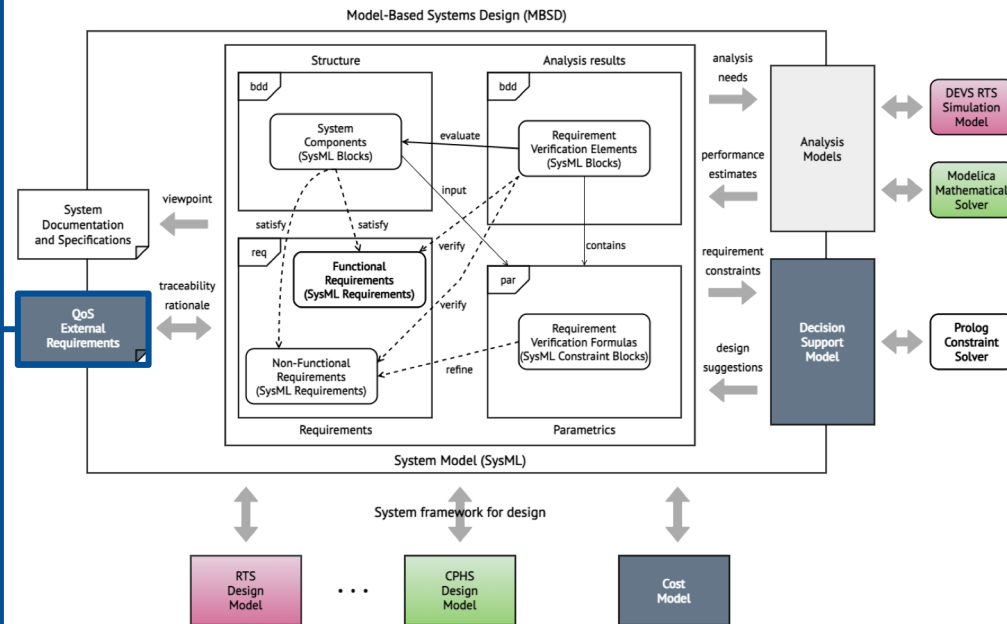
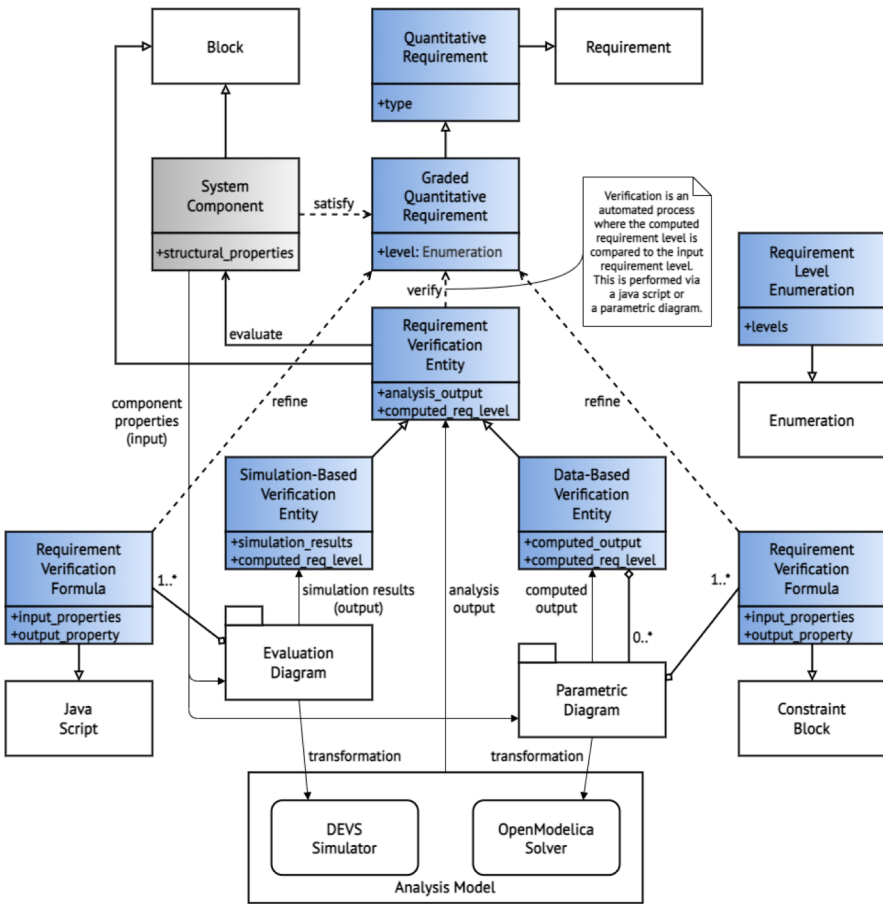




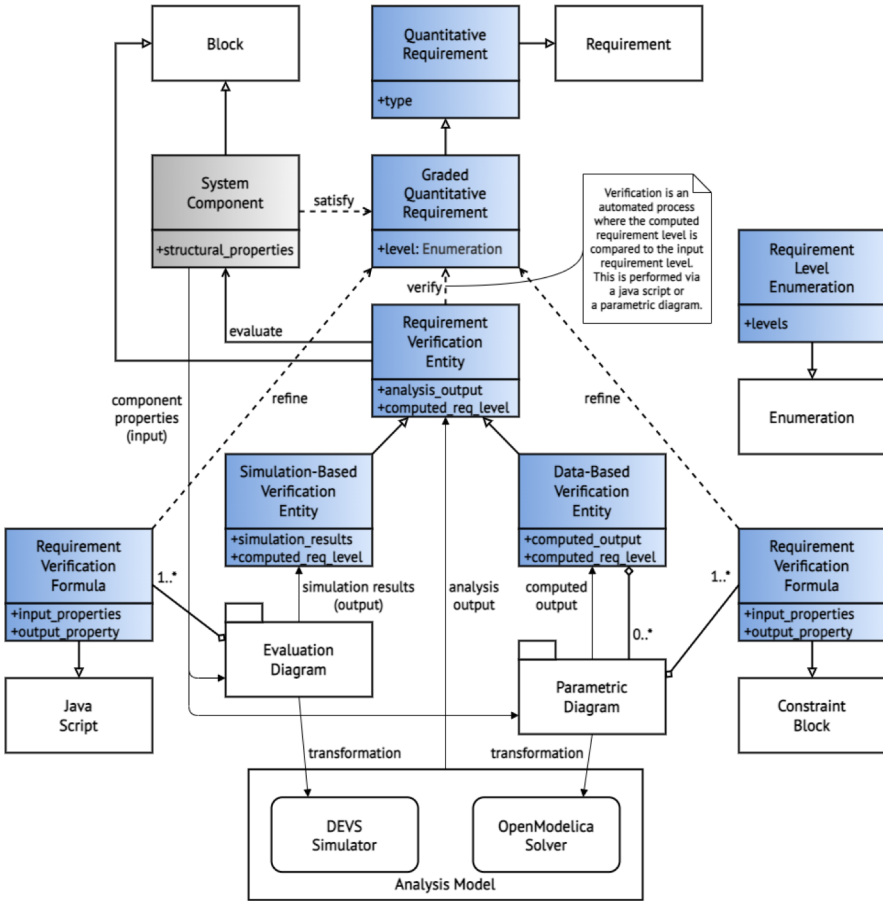


Managing Non-functional Requirements

Extending SysML to integrate QoS requirements



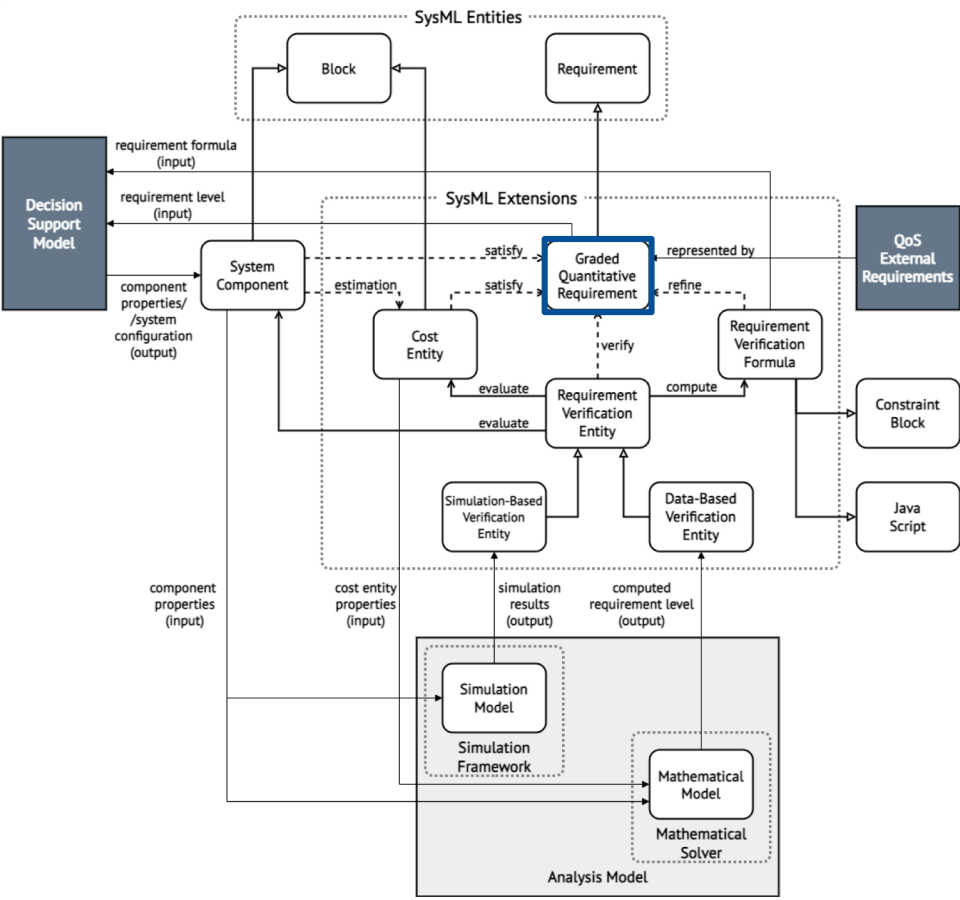
Extending SysML to integrate QoS requirements



- QoS SysML profile
- Collection of extensions to serve QoS¹
 - define quantitative requirements
 - grade them with levels
 - define requirement verification entities
 - incorporate analysis results from diagrams
 - verify requirements against analysis results
 - levels should be achieved to satisfy requirements

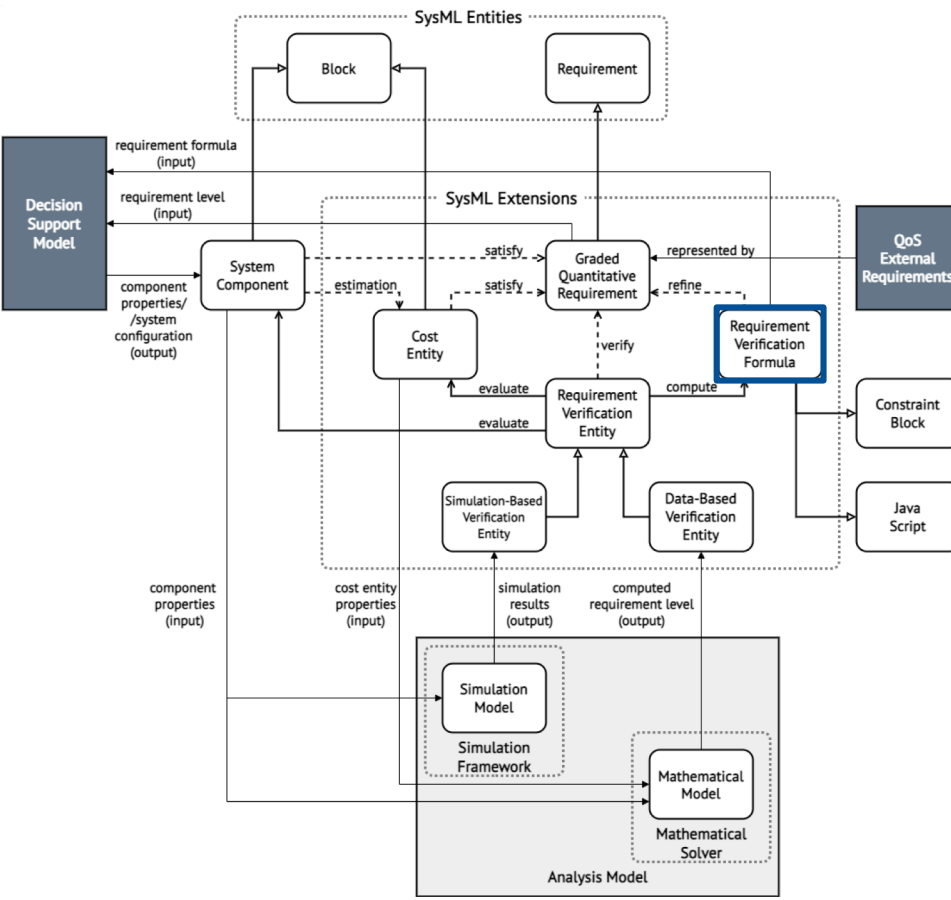
1. Kotronis, C., Routis, I., Tsadimas, A., Kapos, G. D., Dalakas, V., Nikolaidou, M., & Anagnostopoulos, D. (2021). Engaging human participants in CPHS design: A model-based approach.

Basic SysML extensions: Graded requirement



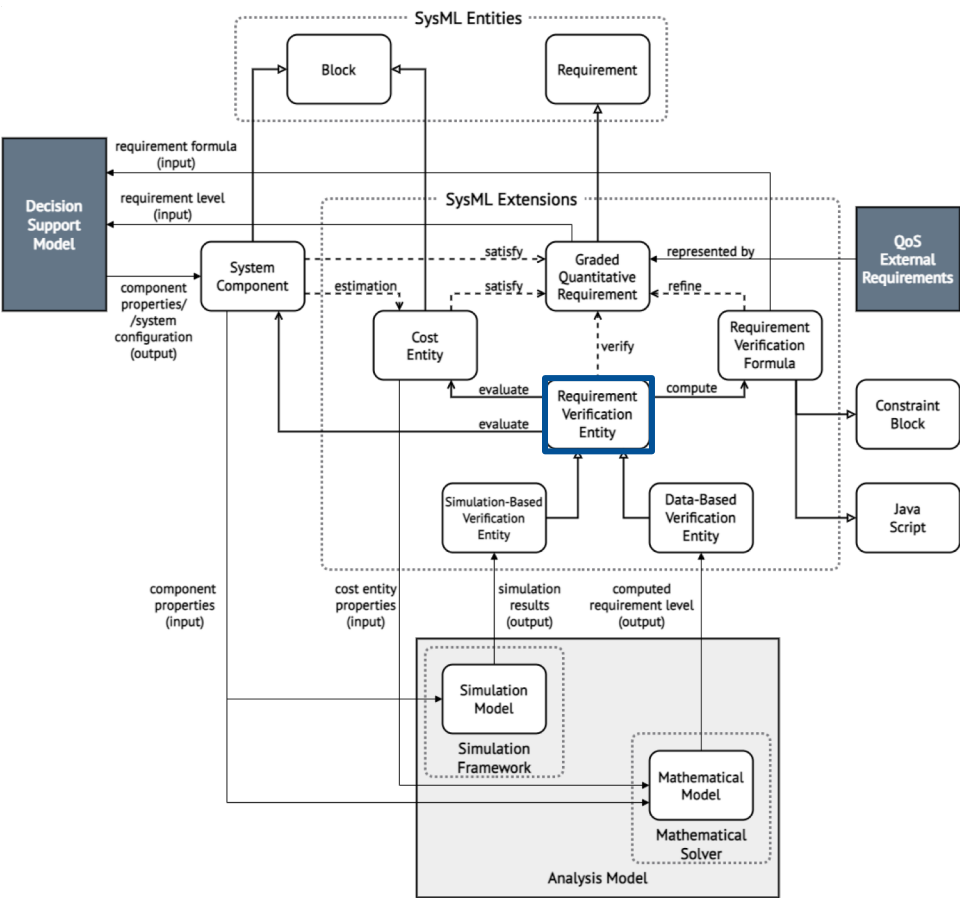
- Extends SysML Requirement
- Serves QoS
- Relates text-based QoS external requirements to system design

Basic SysML extensions: Requirement verification



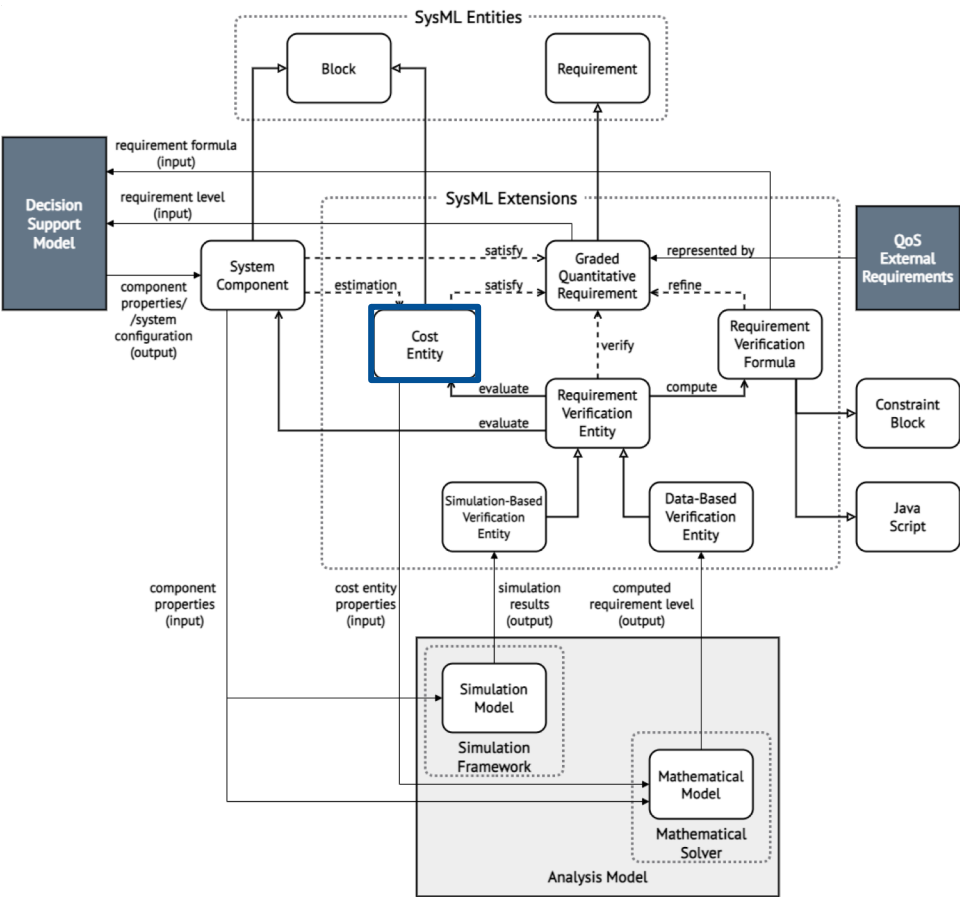
- Defined as
 - SysML Constraint Block extension
 - or JS code
- Described as
 - mathematical equation of quantitative indicators
 - or logical expression of quality indicators (“if-then-else” formula)
- Depicts conditions
 - must be satisfied by properties of model entities
 - for associated requirement(s) to be verified

Basic SysML extensions: Requirement verification



- Defined as SysML Block
- Acts as test/verification case
 - support quantitative requirements verification
- Primary goal:
 - store analysis results, i.e. simulation or mathematically-computed values
 - check values against requirements

Basic SysML extensions: Cost entity

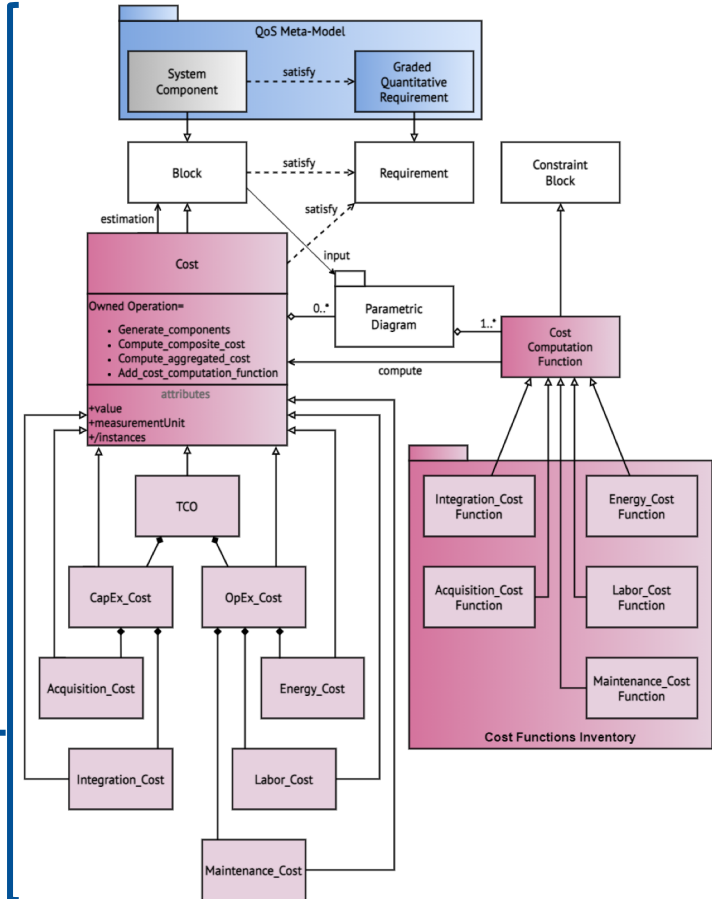
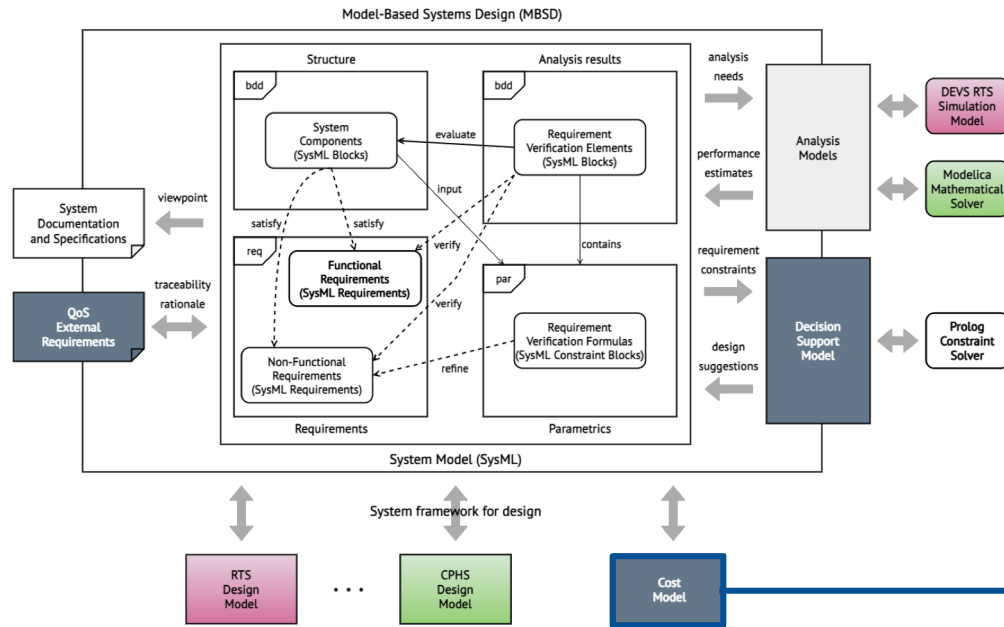


- Defined as SysML block
- Used for financial assessment of component/system
- Includes value properties related to costs
- Satisfies cost-related graded quantitative requirements

⇒ We keep separate cost entities

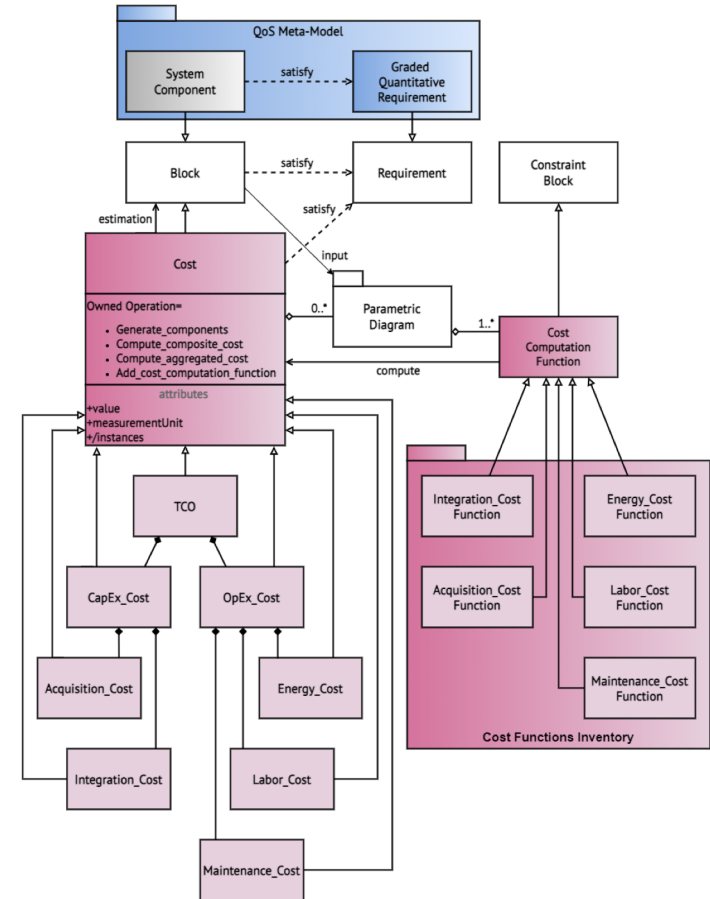
- no component overload
- clearer, accurate, and extensible model
- discrete structural and economic perspectives

Extending SysML to integrate cost concepts



Extending SysML to integrate cost concepts

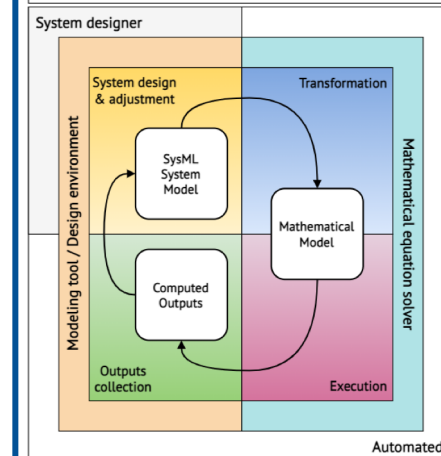
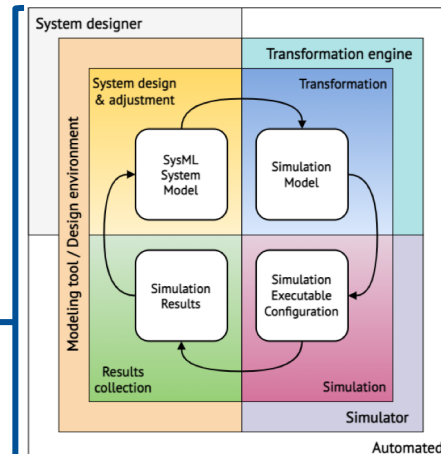
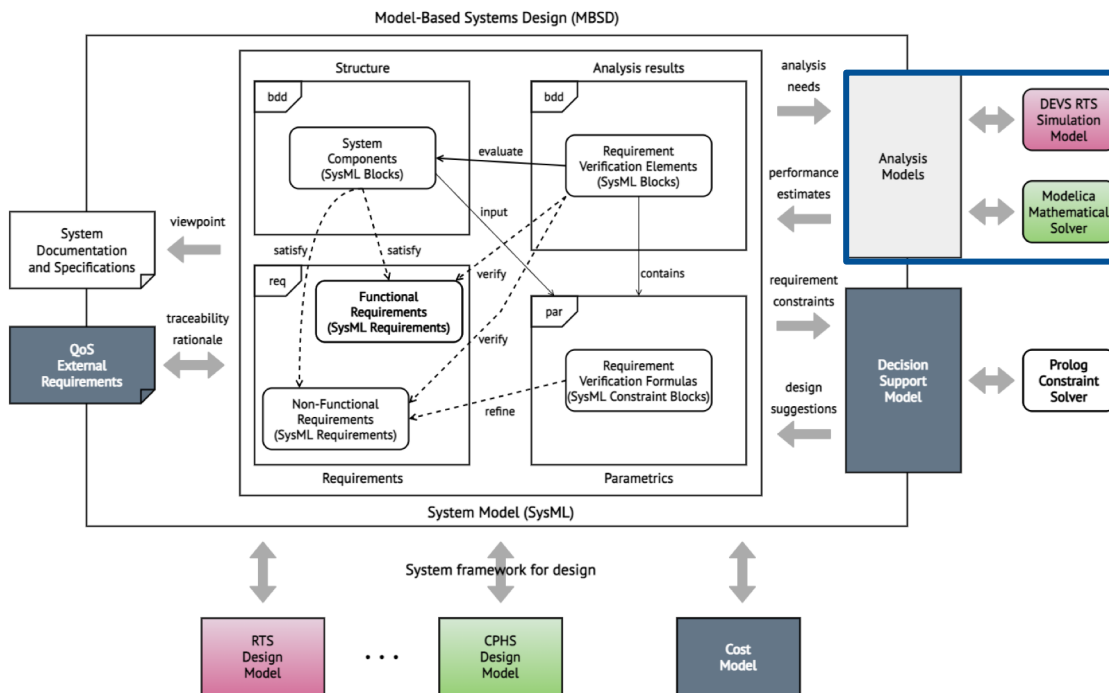
- Cost SysML profile¹
- Generic and extendable
 - encode cost design parameters
 - address different cost categories
 - enable cost analysis
 - capital/operational system expenses
 - individual cost of a single unit
 - traverse system to generate/compute costs
- Cost functions inventory/library
 - enable cost computations
 - readily available to designer





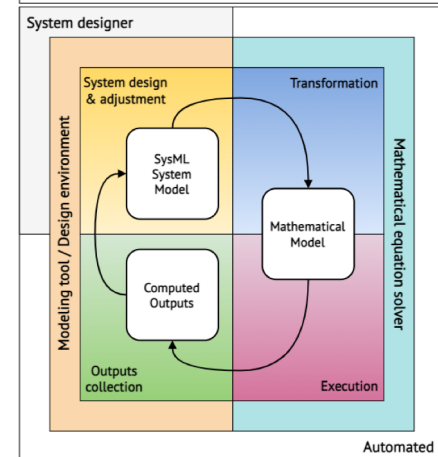
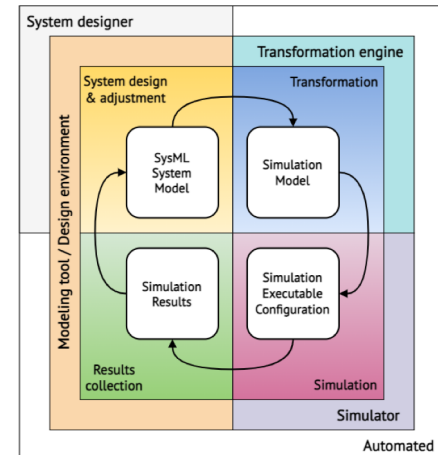
External Tool Integration

Analysis models

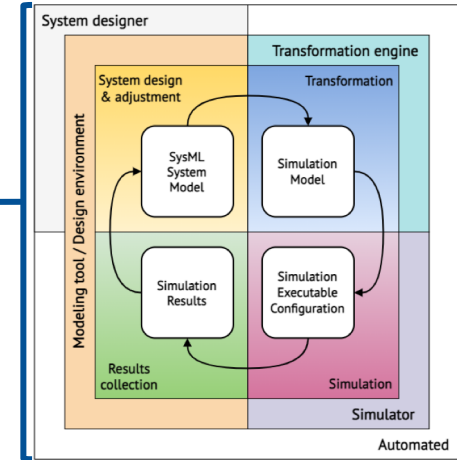
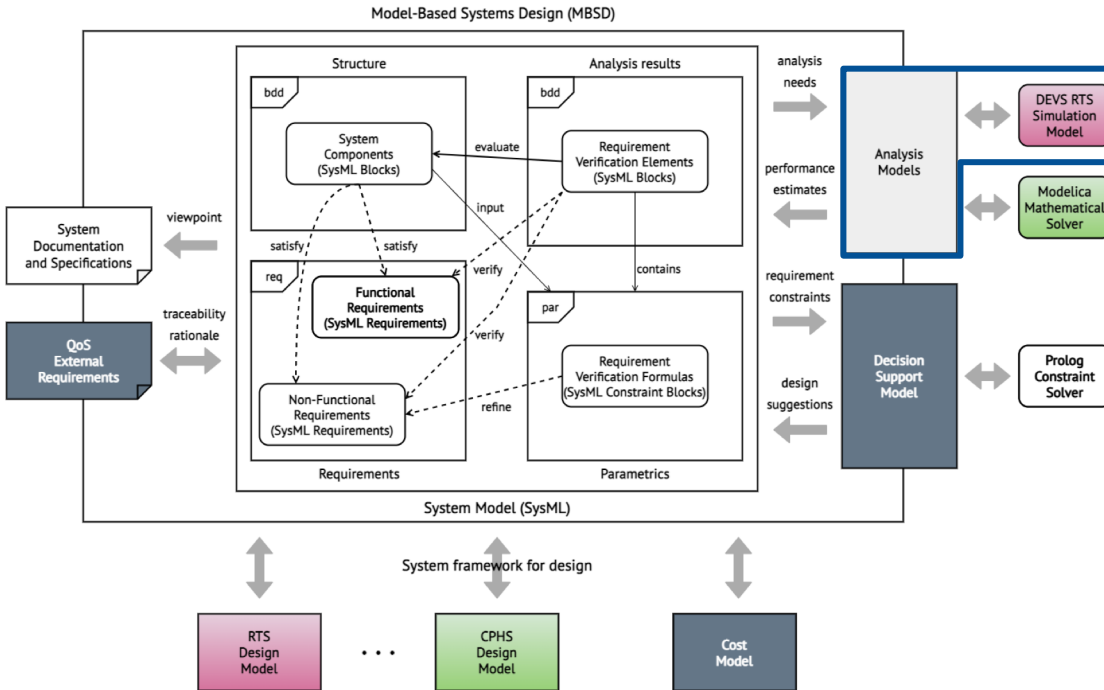


Analysis models

- Produce results
 - to evaluate quantitative requirements
 - are stored in specific profile entities
- 2 kinds of such models
 - Discrete-event simulation
 - Mathematical equations solvers with parametric values and relationships
- Either customized for the specific domain or general

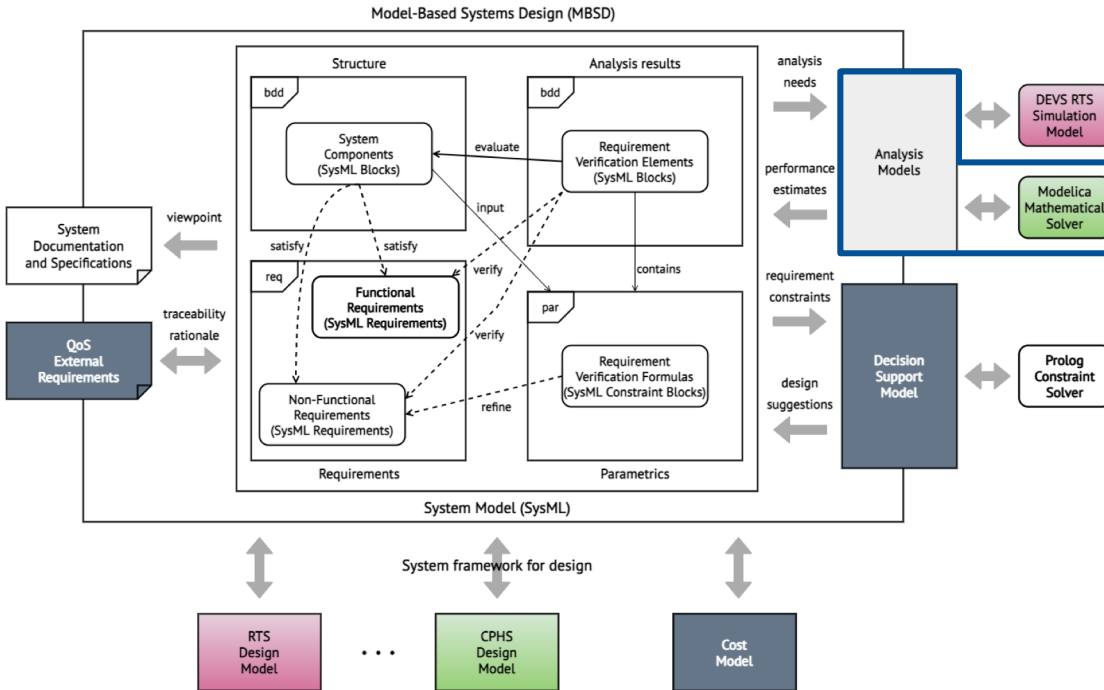


Analysis models

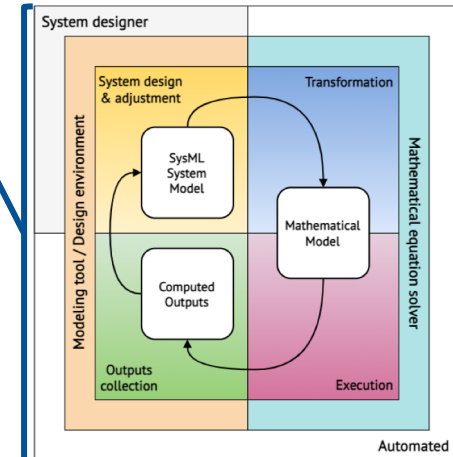


- **RTS → large-scale systems**
 - cannot be modeled by simple functions
 - need repeated simulations
 - extract results for massive commuter populations¹
- **Use of external simulation**
 - DEVS framework

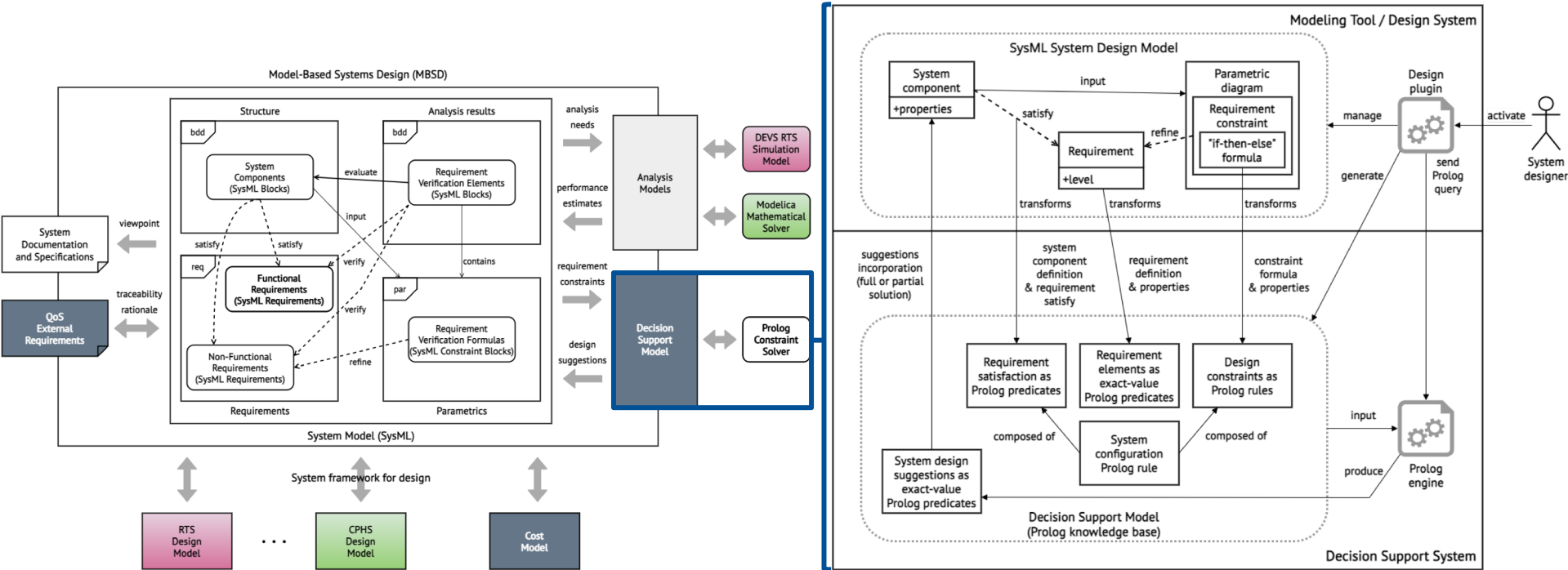
Analysis models



- CPHS → personalized systems¹
 - analyzed with customized functions
 - encode needs of user(s)
- Mathematical solver
 - compute analysis results
 - OpenModelica

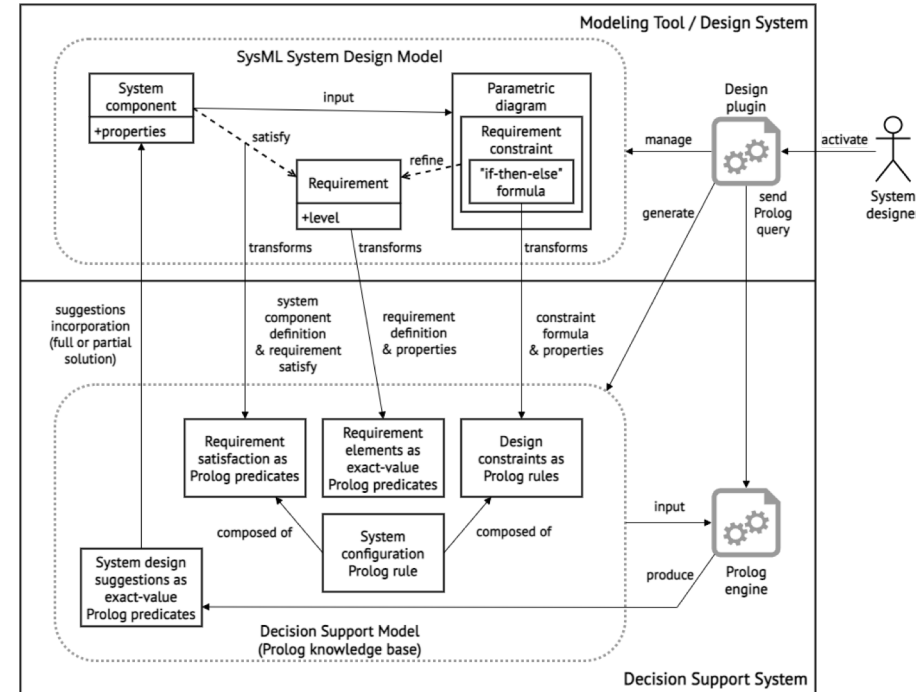


Automated decision-making within SysML models



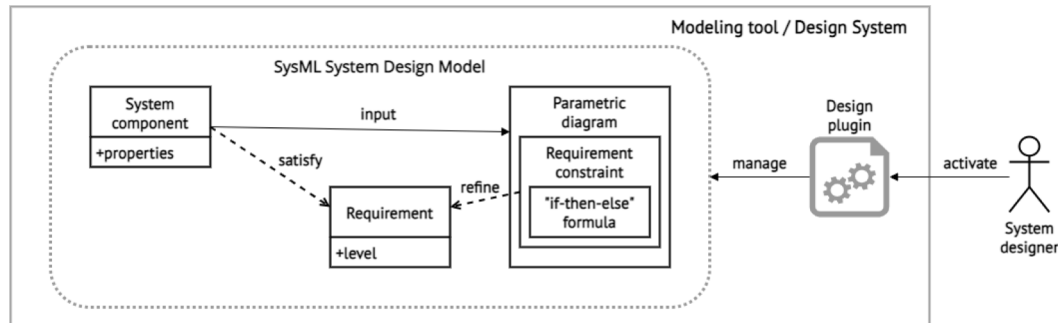
Automated decision-making within SysML models

- Provide system designer with decision support capabilities¹
 - automated suggestions
 - alternative design solutions exploration
 - solutions incorporated back into SysML system model
- Maintain generality
 - without extending SysML
 - applicable to various SoS, described in SysML
 - applied for entire or part of system hierarchy

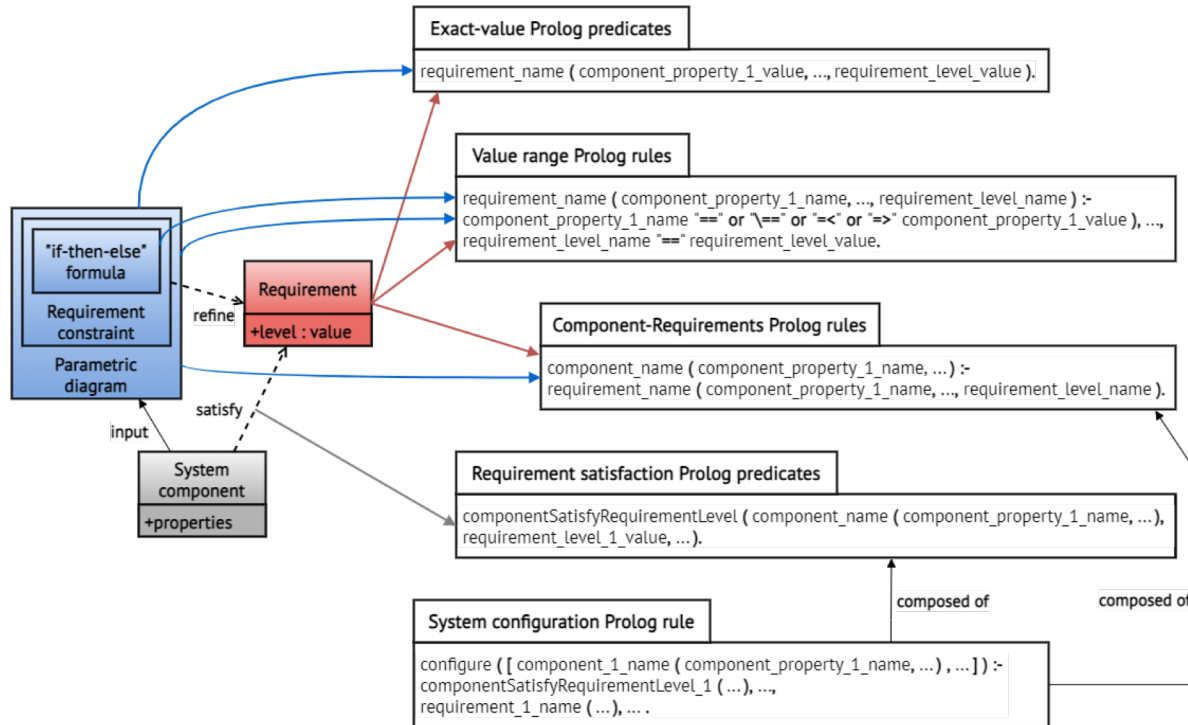


Transforming SysML model to decision model (I)

- Designer activates plugin in design tool
 - Automated SysML-to-decision model transformation process
- Plugin validates SysML model's correctness & completeness
 - Validation rules capture conditions
 - E.g., every constraint must contain a formula and connect with requirement

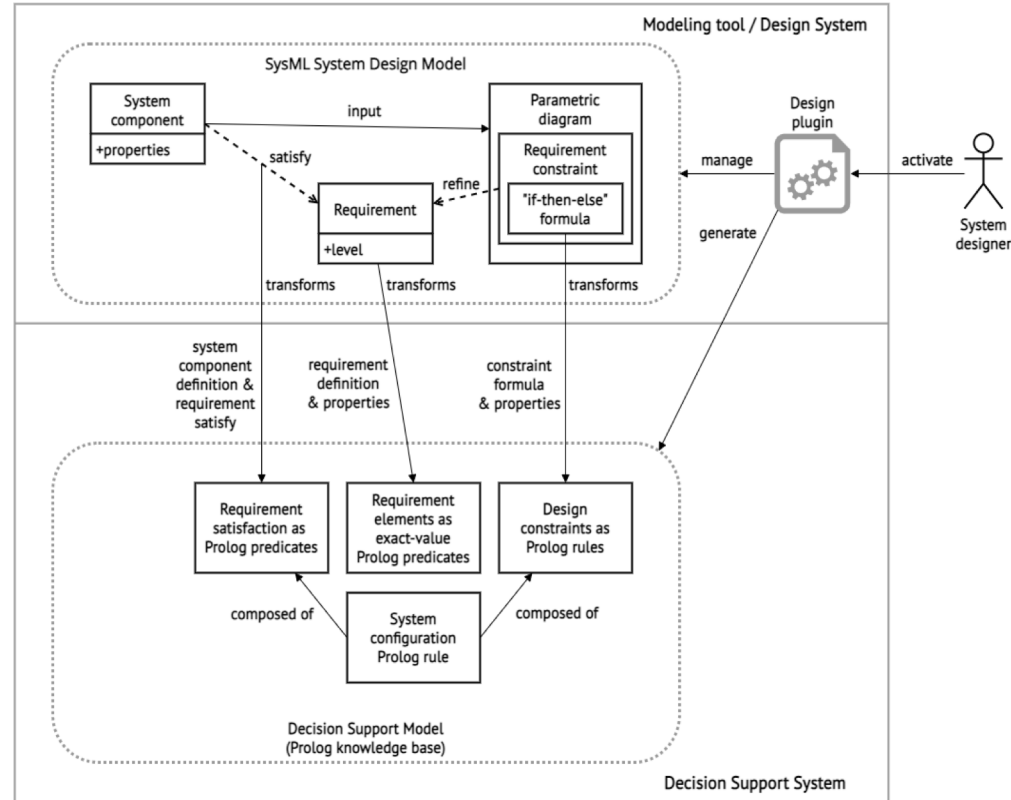


SysML to Prolog entity transformation (one-way)¹



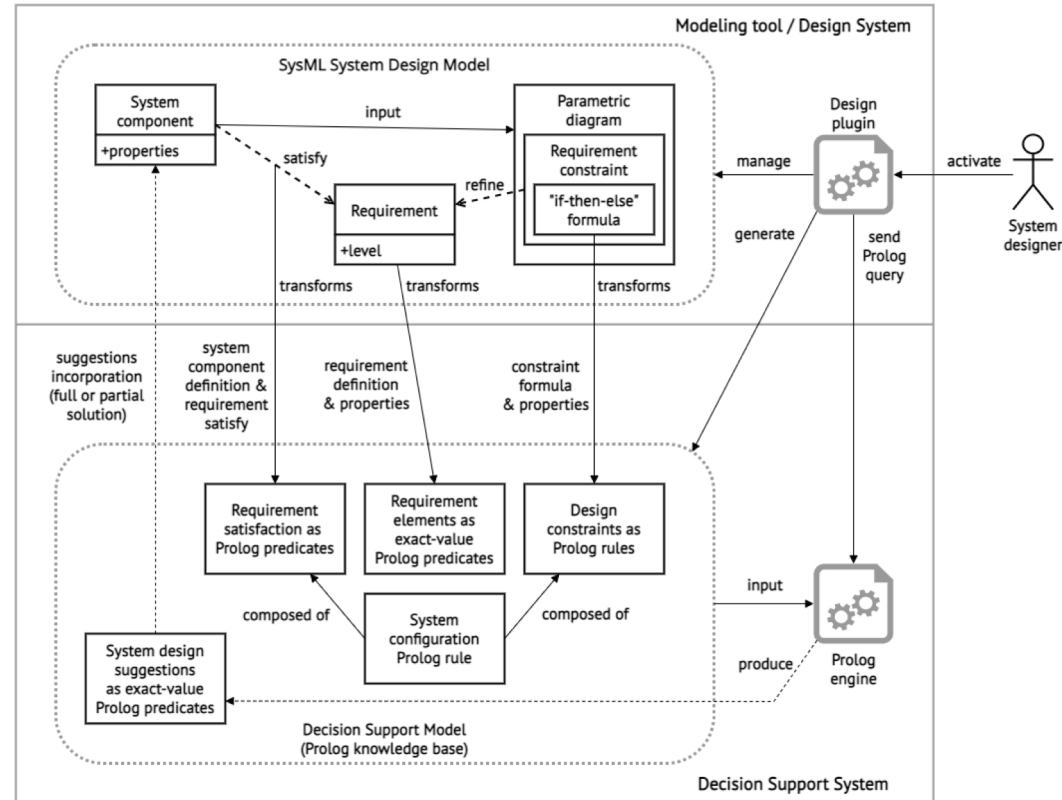
Transforming SysML model to decision model (II)

- Plugin generates decision model
 - It is a Prolog knowledge base
 - Stems directly from SysML model
- Requirements, constraints and requirements-components associations
 - Are transformed into Prolog predicates/rules
 - Predicates/rules are combined into general system configuration rule

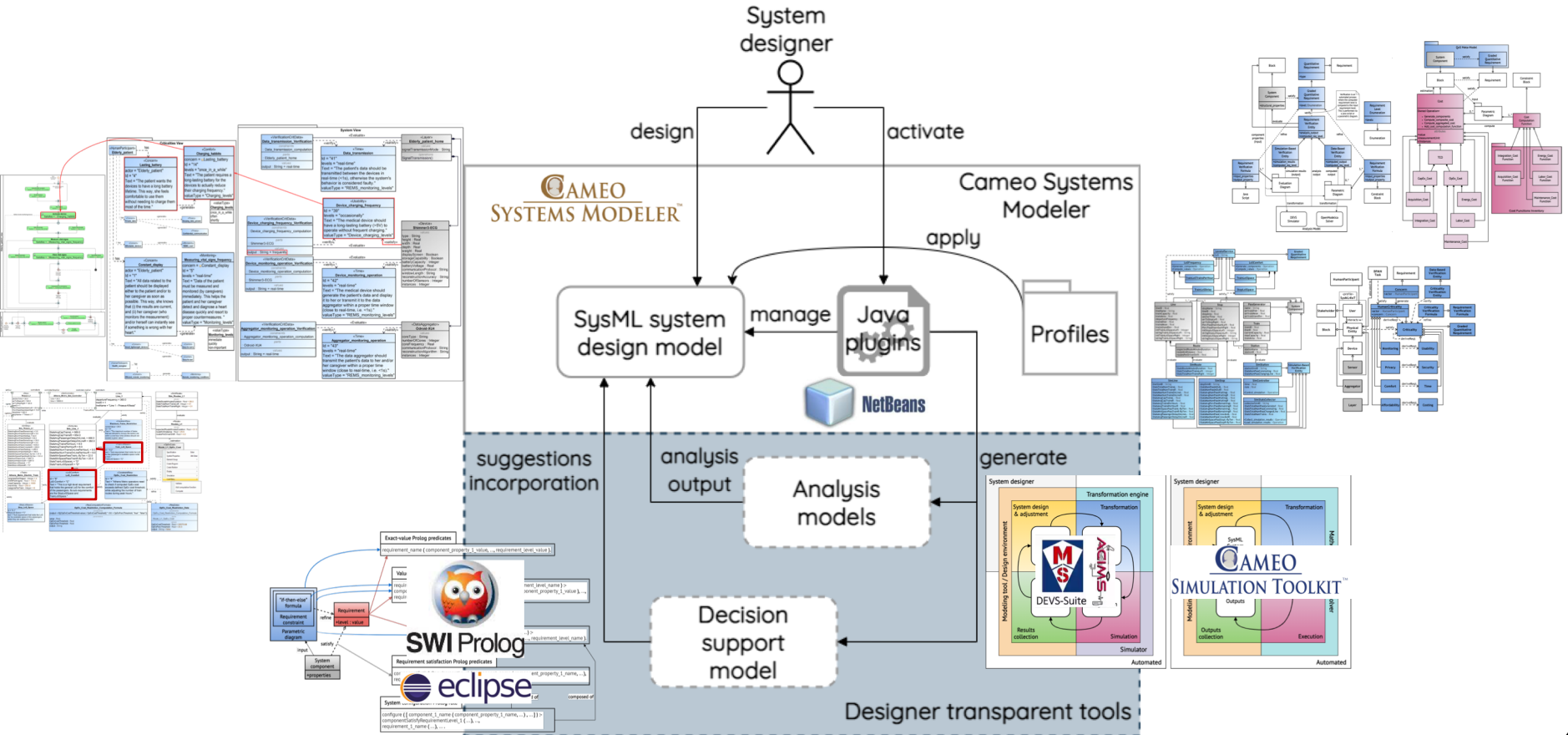


Transforming SysML model to decision model (III)

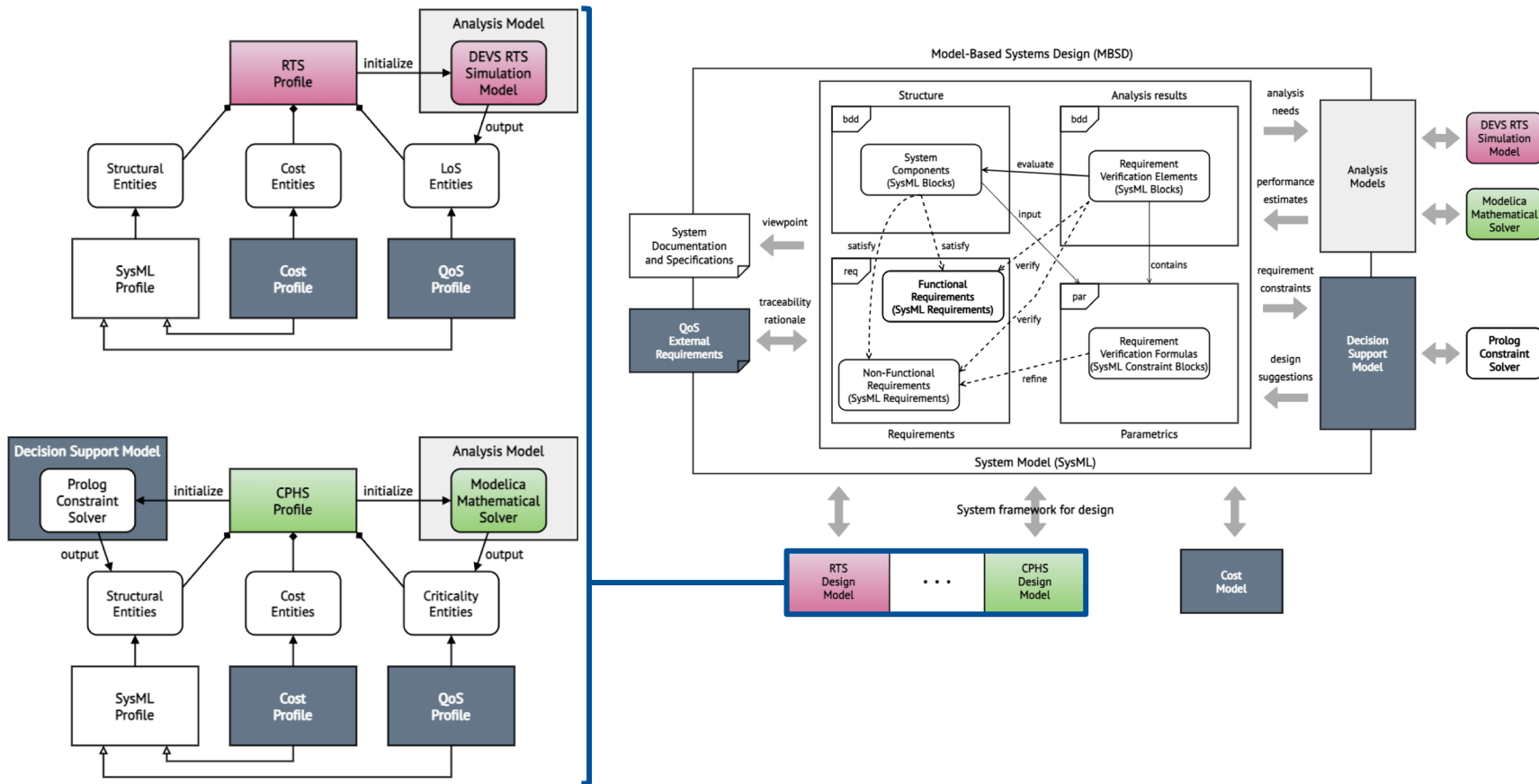
- Plugin sends Prolog query
 - “Asks” for feasible system configuration solution
- Decision model is input to external tool
- System design suggestions are output
 - Exact-value Prolog predicates
- If feasible solution is found
 - Full or partial
 - Resulting suggestions incorporated into SysML model
 - I.e. values fed back into components' properties



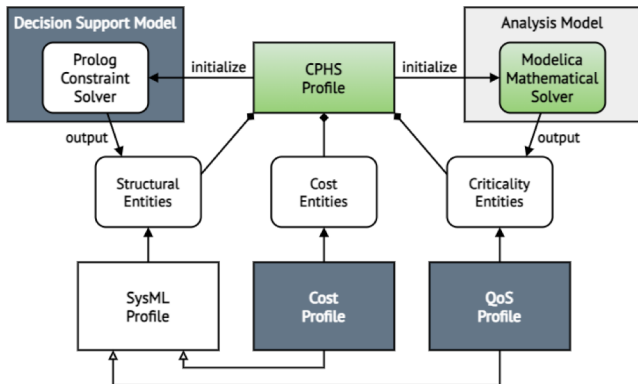
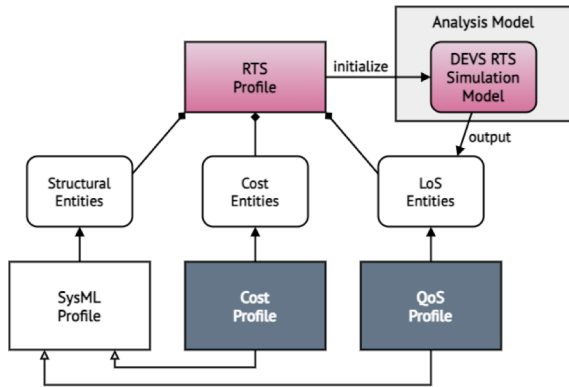
Implementation



QoS-aware MBSD SysML in specific domains



QoS-aware MBSD SysML in specific domains



- Apply MBSD to specific domains
 - via domain-specific SysML profiles
- Domain SysML profile
 - contains basic domain stereotypes
 - describe structural entities
- Integrate QoS and Cost profiles
- Analysis models
 - produce results to satisfy (or not) requirements
- Logical solver
 - support decision-making process



Demonstration & Evaluation LoS-aware RTS operation design

Railway Transportation Systems

- Railway Transportation Systems (RTSs)
 - Comprise of diverse sub-systems / components and are characterized as SoSs
 - Can be modeled as a system composed of stations, lines, stops, trains, commuting passengers
- LoS represents
 - quality of the transportation services
 - from the passengers' point of view
 - quantified in six classes
 - F (worst situation) to A (best service quality)
- LoS is focused on passenger comfort
 - available space for passengers in Stops (platforms) / Trains
 - departure frequency and on-time arrival of Trains

LoS	Category	Passenger space (m^2)
A	Free movement	≥ 1.21
B	Restricted movement	0.93 - 1.21
C	Personal comfort	0.65 - 0.93
D	Occasional contact with others	0.27 - 0.65
E	Contact with others	0.19 - 0.27
F	Frequent contact with others	< 0.19

RTS: Athens Metro

- Study passenger comfort on trains/stations during peak hours¹
- Costs are directly connected to comfort-related metrics
 - e.g., more space → more frequent trains → more operational cost

⇒ Explore how to achieve certain level for passenger comfort²

- Estimate passenger comfort during peak hours
- Improvement comfort under certain operational budget constraints

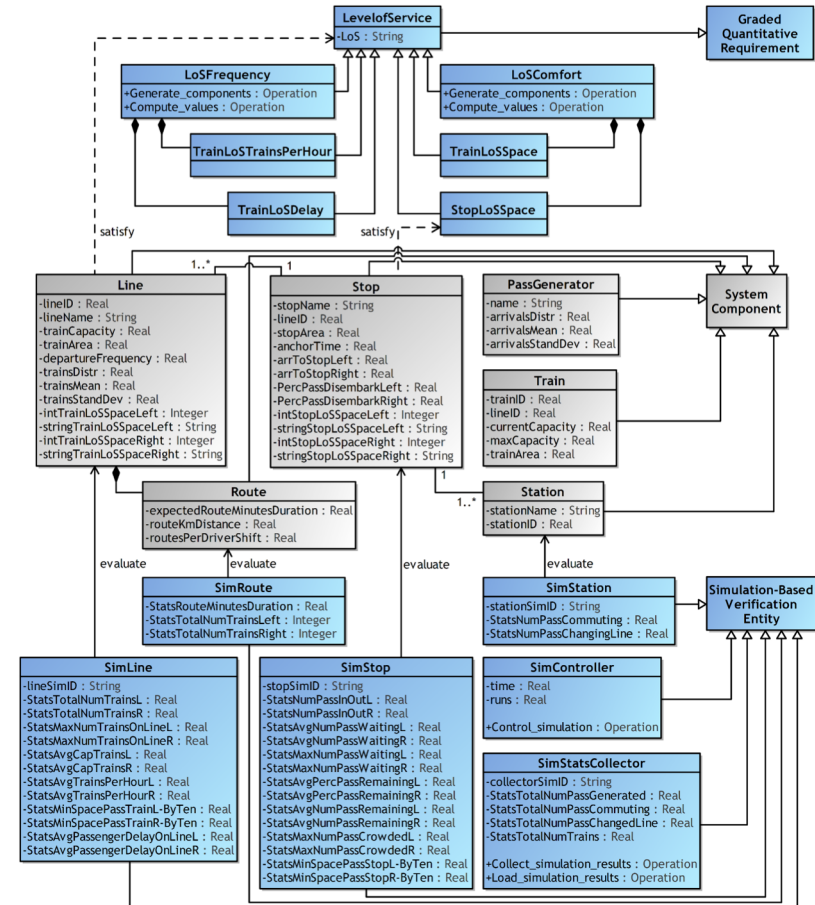
1.Manual, H. C. (2010). HCM2010. Transportation Research Board, National Research Council, Washington, DC, 1207.

2.Kotronis, C., Nikolaidou, M., Kapos, G. D., Tsadimas, A., Dalakas, V., & Anagnostopoulos, D. (2020).

Employing SysML to model and explore levels-of-service: The case of passenger comfort in railway transportation systems. Systems Engineering, 23(1), 82-99.

Target

- Model RTS
 - Structural entities like lines, stops, trains
 - LoS-related requirements
 - Associated to lines, stops
 - Simulation entities to store analysis results
- Compute LoS
 - Using simulations results
 - Verify LoS requirements
- Compute OpEx cost
- Explore potential LoS improvements under budgetary constraints



RTS Engineers Viewpoint

1. Set different LoS bounds (upper and lower) for different LoS requirements
2. Describe verification formulas

LoS Bounds

LoS Comfort

1 - Introduction

2 - TrainLoSSpace

3 - StopLoSSpace

Description

The **StopLosSpace** requirement has lower and upper bounds that enclose the Level of Service categories (from A to F)

Bounds

LoS A

LowerUpper

0.00.19

LoS B

LowerUpper

0.190.28

LoS C

LowerUpper

0.280.65

LoS D

LowerUpper

0.650.93

LoS E

LowerUpper

0.931.21

LoS F

LowerUpper

1.212.0

Default Values

Reset Values

Menu

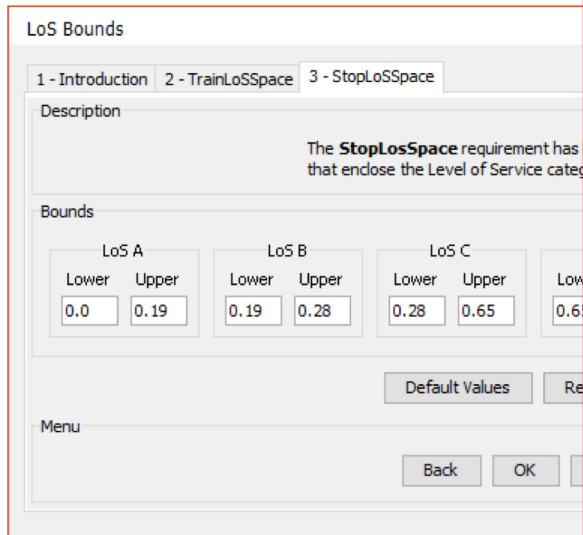
Back

OK

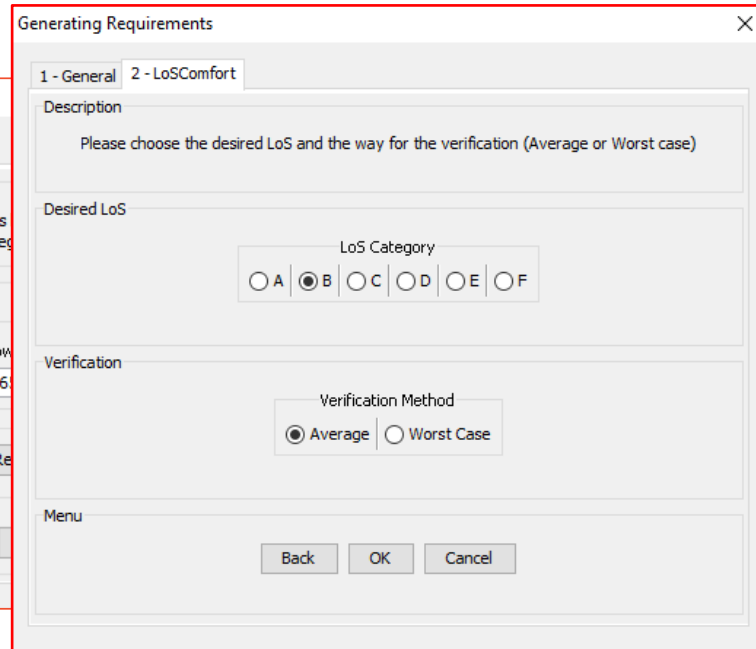
Cancel

RTS Engineers Viewpoint

1. Set different LoS bounds (upper and lower) for different LoS requirements
2. Describe verification formulas



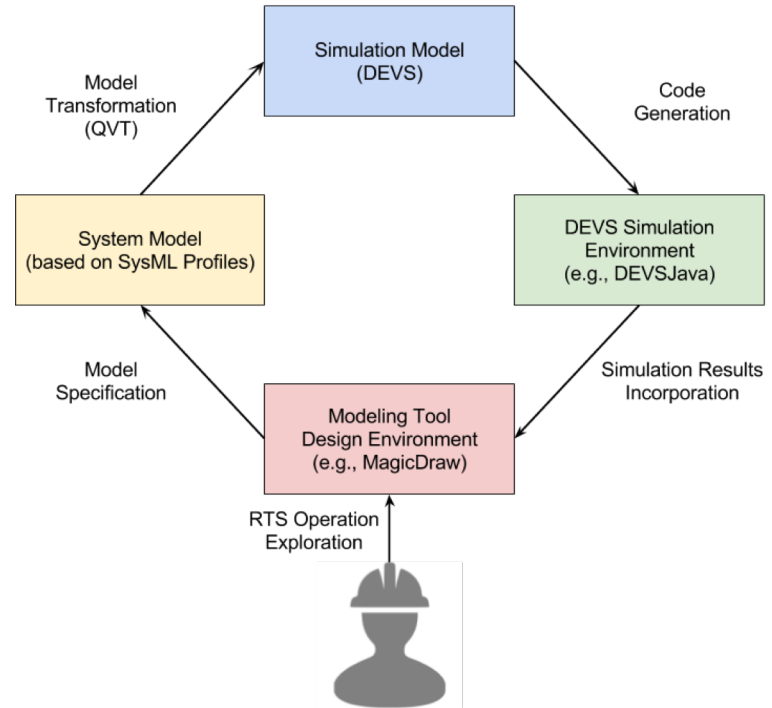
The screenshot shows the 'LoS Bounds' dialog box with three tabs: '1 - Introduction', '2 - TrainLoSSpace', and '3 - StopLoSSpace'. The '3 - StopLoSSpace' tab is selected. The 'Description' field contains the text: 'The **StopLoSSpace** requirement has that enclose the Level of Service category'. Below the description, there are three sections for 'LoS A', 'LoS B', and 'LoS C', each with 'Lower' and 'Upper' input fields. The values are: LoS A (Lower: 0.0, Upper: 0.19), LoS B (Lower: 0.19, Upper: 0.28), and LoS C (Lower: 0.28, Upper: 0.65). There are also 'Default Values' and 'Reset' buttons. At the bottom, there is a 'Menu' section with 'Back' and 'OK' buttons.



The screenshot shows the 'Generating Requirements' dialog box with two tabs: '1 - General' and '2 - LoSComfort'. The '2 - LoSComfort' tab is selected. The 'Description' field contains the text: 'Please choose the desired LoS and the way for the verification (Average or Worst case)'. Below the description, there is a 'Desired LoS' section with a 'LoS Category' row of radio buttons: A, B, C, D, E, F. Radio button B is selected. Below this, there is a 'Verification' section with a 'Verification Method' row of radio buttons: Average, Worst Case. Radio button Average is selected. At the bottom, there is a 'Menu' section with 'Back', 'OK', and 'Cancel' buttons.

RTS Analysis Framework

- DEVS4SysML profile* is used for simulation purposes
 - DEVSJava is employed as a simulation platform
 - QVT is utilized for model transformation
 - Simulation results are integrated back into the system model
- RTS simulation libraries were developed for DEVSJava



Automated Activities

- Simulation instantiation
 - Simulation results incorporation (simulation entities)
 - LoS Comfort Computation
 - LoS Comfort Requirement Verification
-
- Activities are started by the designer through Cameo Interface

LoS Computation Formulas

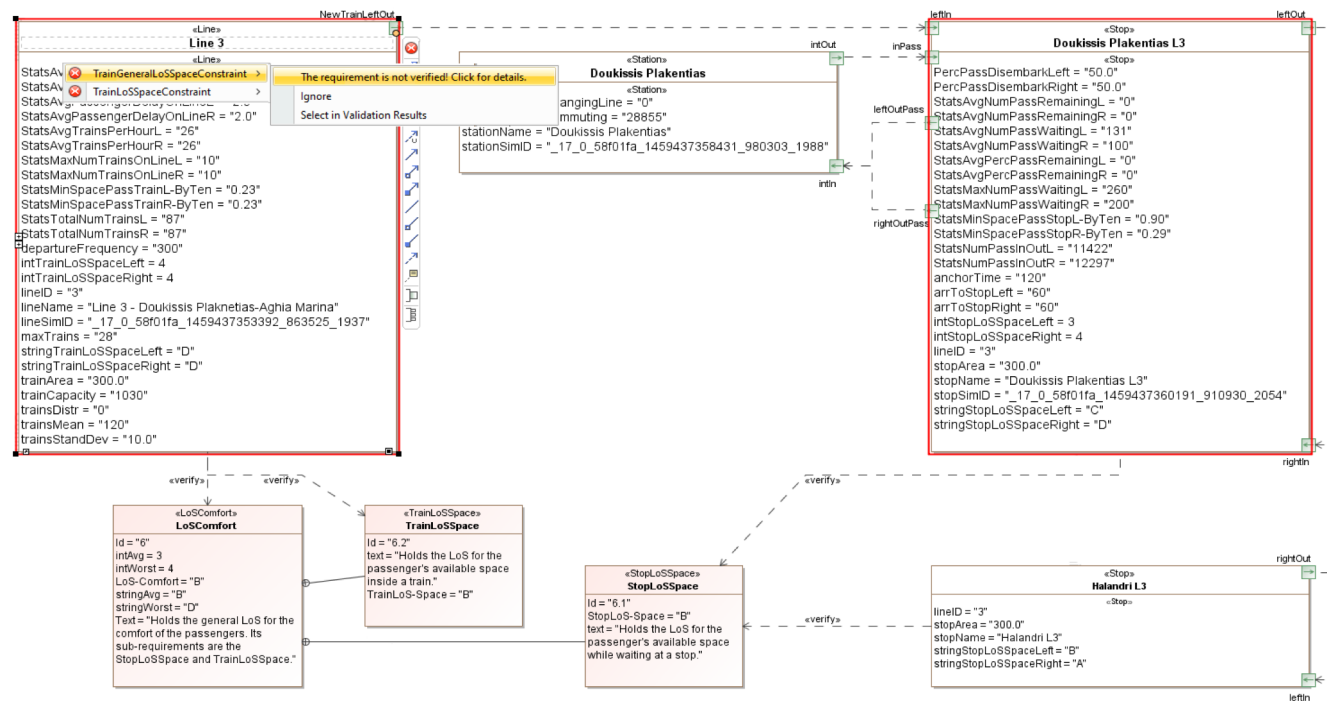
Average space LoS at Stop (ALS):

$$ALS(S, d) = \sum_t LoS(S, d, t) / N$$

Average space LoS at Train (ALT):

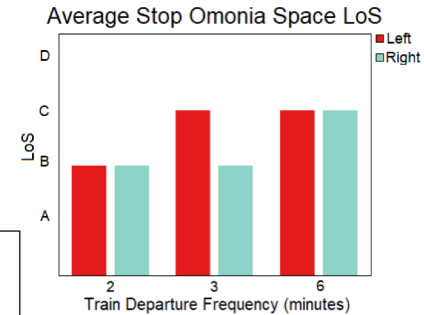
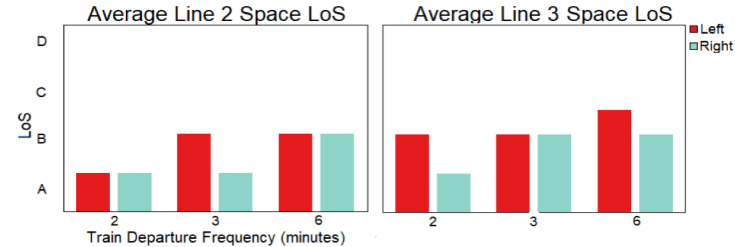
$$ALT(T, d) = \sum_t LoS(T, d, t) / N$$

3. checks if desired LoS is verified by model entities
4. evaluates environment's automated feedback



Results and Experience obtained¹

- Assisted the Metro operator to identify Train frequency to
 - balance QoS and OpEx
- The operator provided passenger data, the configuration of network and cost estimation formulas
- The operator appreciated the fact the all data where accessible through a single environment, still skeptical for using SysML



Athens Metro Line 1		Routes (L + R)	Peak hours OpEx (E)	Peak hours OpEx increase (%)	Daily OpEx (E)	Daily OpEx increase (%)	Avg LoS	Accepted solution
Line 1 trains frequency (minutes)	7	56	35575.68	–	160494.03	–	D	×
	6	62	38827.15	9.1	164213.1	9.14	D	×
	5	78	49551.84	39.2	174243.98	15.8	C	✓
	3	128	81315.8	128.5	205632.99	36.7	B	×

1.Kotronis, C., Nikolaidou, M., Kapos, G. D., Tsadimas, A., Dalakas, V., & Anagnostopoulos, D. (2020).

Employing SysML to model and explore levels-of-service: The case of passenger comfort in railway transportation systems. Systems Engineering, 23(1), 82-99.

Profiles and Tutorial

- Profiles and plugins are available at

<https://bitbucket.org/ckotronis-phd/plugins/src/main/>

- A demonstration is available at [RTS_demo](#)



Demonstration & Evaluation Criticality-aware CPHS design

CPHS application domain - REMS

⇒ Human participants affect the operation of CPHS

- To become willing participants they should participate in system design
- Their concerns should be taken into account
- Multi-view MBSD facilitates
 - Definition of human concerns
 - Transformation of human concerns into design requirements and their correlation to system components
 - Communicating to human-participants how their concerns are treated
- Remote Elderly Monitoring System (REMS)
 - Part of EmBioT research project to develop wearables to monitor patient's heart condition
 - REMS solution designer should assist patients choose appropriate REMS home configuration taking into account their concerns

Criticality concept

- Indicates the severity for the human participant if the system does not accommodate his/her concern
- Assigned a level of importance by the participant
- Categorized as human or system
 - human → restrict participants' activities, defined by patient
 - system → restrict system components, extracted from human criticalities by the system designer

CPHS design views

⇒ Human view^{1,2}

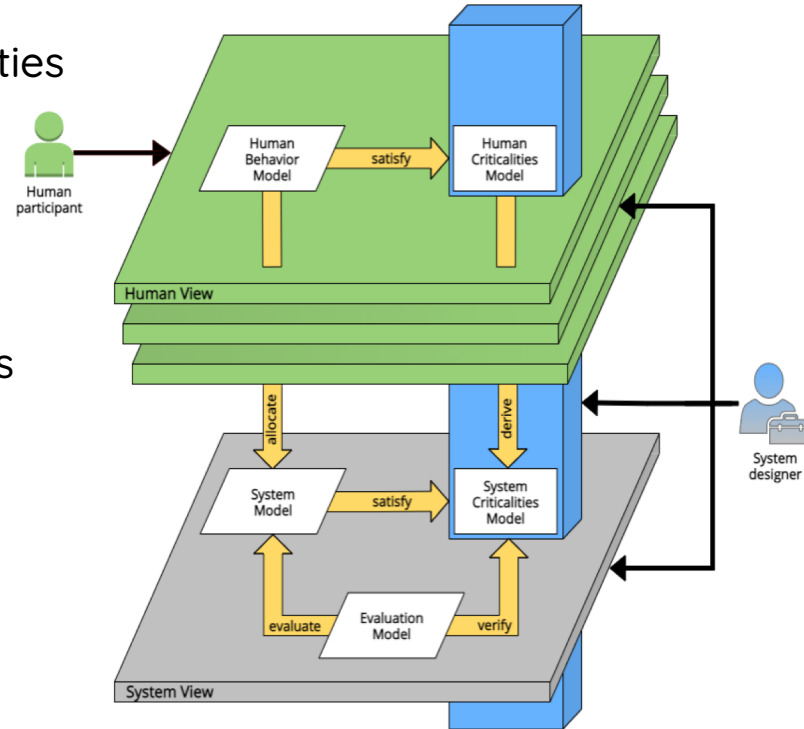
- Human Behavior Model
 - entities related to human behavior like activities

⇒ Criticalities view

- Human Criticalities Model
 - human concerns and criticalities
- System Criticalities Model
 - system criticalities, derived from human ones

⇒ System view

- System Model
 - system structure, defined by designer
- Evaluation Model
 - verify system criticalities



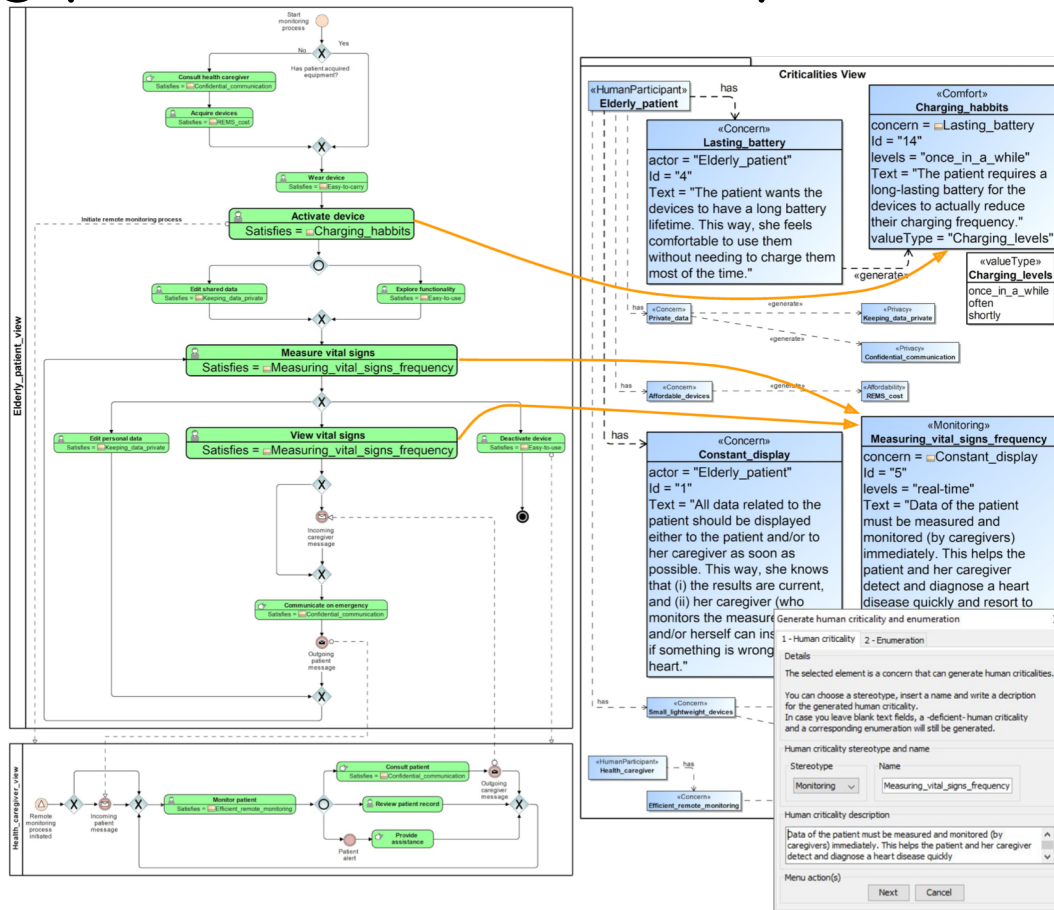
1.Kotronis, C., Routis, I., Tsadimas, A., Nikolaidou, M., & Anagnostopoulos, D. (2019, July).

A model-based approach for the design of cyber-physical human systems emphasizing human concerns. In 2019 IEEE ICIOT (pp. 100-107). IEEE.

1.Kotronis, C., Routis, I., Tsadimas, A., Kapos, G. D., Dalakas, V., Nikolaidou, M., & Anagnostopoulos, D. (2021).

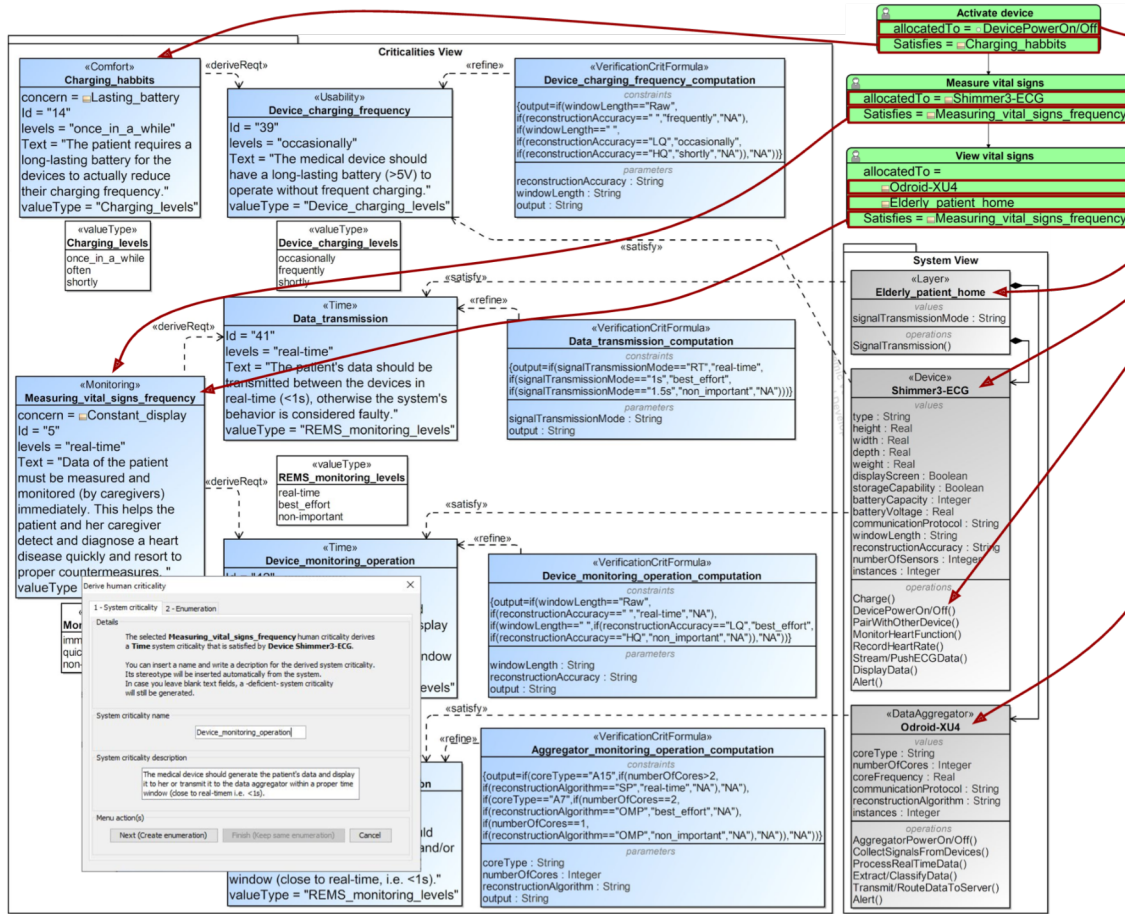
Engaging human participants in CPHS design: A model-based approach. ACM Transactions on Internet of Things Journal. submitted for publication.

Mapping patient activities to patient concerns



Performed by patient with the assistance of the solution designer

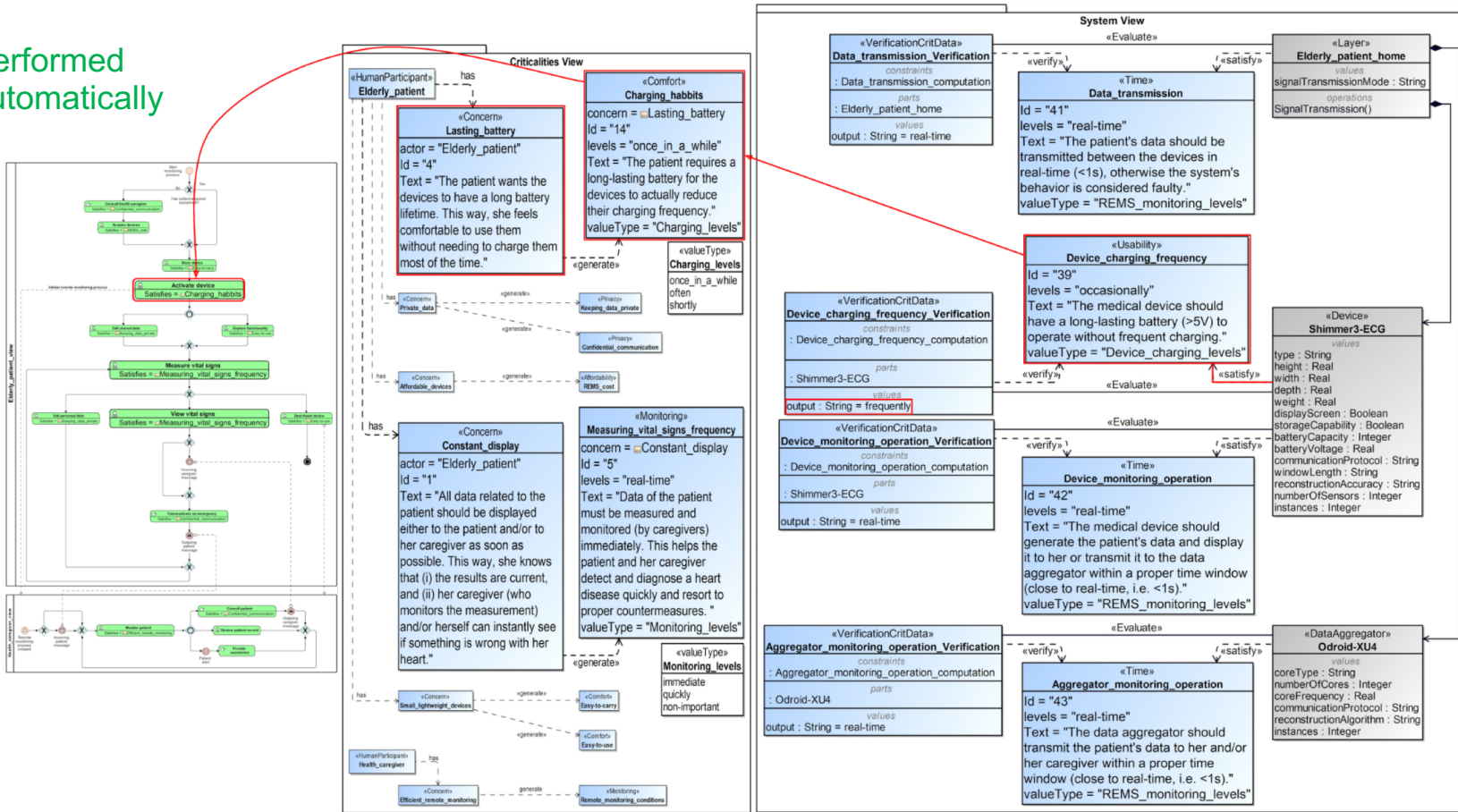
Derivation of system criticalities from human ones



Performed by
the solution
designer in a
semi-
automated
fashion

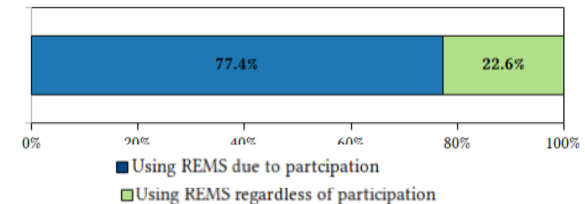
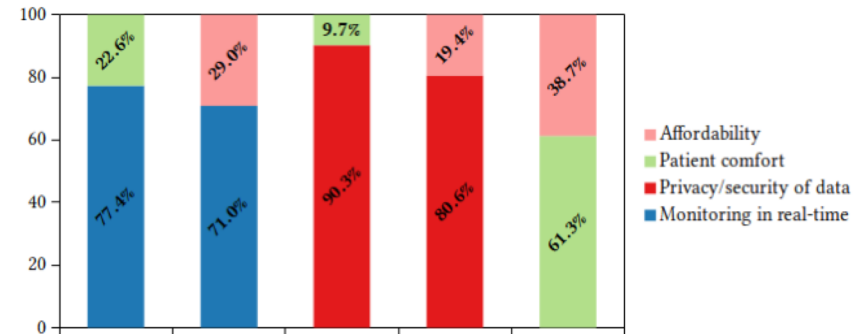
Verifying criticalities across views

Performed automatically



Results and Experience obtained

- Four criticality categories were identified
- Right balance between them is not trivial
 - Patients have the change to decide what is more important for them by “co-designing” the system
 - Solution designers are provided with a comprehensive manner to transform concerns into system design requirements and automatically verify them
- Patients participation in system design is important
 - affected decision to use and operate a personalized REMS



Profiles and Tutorial

- Profiles and plugins are available at

<https://bitbucket.org/ckotronis-phd/plugins/src/main/>

- A demonstration is available at [REMS demo](#)



Discussion

Implementation Limitations

- MBSD using SysML is been tested in practice
 - Using tools such as Cameo System Modeler, profiles and plugins
- Integration
 - profiles and plugins
 - external tools
- SysML capabilities
 - Difficulties arise when trying to “work” with your models (active SysML models)
 - Similar to UML and the extraction of executable code

Integration

- All external tools integrated in our framework, provided a Java API to be integrated in Cameo Java Plugins
 - We basically needed ways to a) provide them with data, b) instantiate them and c) integrate results into the SysML models
- The application of more than one profiles is not that easy in practice
 - This becomes more difficult when you are not the one that created them
 - All object-oriented features (class specialization for instance) is always more efficient in theory
- Versioning is always an issue
 - Affecting all the tools integrated and also MagicDraw product suite

SysML Capabilities

- SysML is defined as a graphical language
 - Depiction of information was the main concern
 - However to study design alternatives one needs to "evaluate" models
- Computations are essential to evaluate design solutions
 - In SysML they can be described using parametric diagrams
 - To obtain the result you need external tools (Cameo Modelica plugin)
 - Quicker to integrate them in plugins
 - Known in advance, not defined by the designer

SysML Capabilities

- Analysis results must be integrated in the SysML model
 - History of experiments should be kept
 - This should be facilitated by a specific diagram

In our case we introduced specific stereotypes for the purpose
- SysML standard is not fully supported by modeling tools
 - Details do matter (computation, inheritance, profile definition)
 - SysML support the main reason for choosing MagicDraw and Cameo Suite

Good practices

- Provide a specific order of steps to engineers to apply an MBSD framework
 - Profiles are needed to restrict SysML entities for specific domains
- Define specific generic stereotypes to manage QoS requirements
 - Introduce discrete entity to Integrate the results of analysis tools
- Abstract specific parameters related to specific viewpoint (engineering task) from system entities
 - Cost entities containing information to compute cost of ownership
- Automate tasks for the designer
 - Automated computations were fully appreciated
 - Validate models in all stages

Reality Check

- When applying an MBDS approach in a real case you need to answer specific questions
 - When complex systems are involved, it is complicated!
 - Iterative approach improving design solution in each iteration
 - More than one alternatives should be studied, thus it is important to store and access intermediate results
- Stakeholders of the case are interested in the results and not your approach/ framework
 - Flexibility and customization is needed (integrating and creating SysML profiles and external tools)

Reality Check

- Analysis is always tricky (whether concerning an existing system or not)
 - Designers should really know/study the system and explore its behavior using stress testing or simulation
 - Custom solutions are most probably needed

Addressing obstacles in SysML adoption

Did we manage to hide design complexity?

- QoS requirements are easier defined and explored by engineers
 - Single point of interaction using a specific environment/tool (Cameo Suite)
 - Hidden exhaustive QoS computation details
 - Automated QoS requirement verification

Custom code is needed

Addressing obstacles in SysML adoption

Did we convinced system designers to adopt SysML??

- Establishment of an MBSD approach
 - Simplify usage
 - provision of domain-specific languages DSLs (profiles), hide detailed, low level information, automated integration of external tools
 - Utilize existing experience
 - integrate all information in the system model, provision of libraries

However

- The integration of profiles and tools created by others remains an issue hindering extensive use
- In both case studies engineers asked for our support to use the approach and tools provided

SysML v.2

- Under development by OMG
 - Major change based on KernelML
 - Actually an executable language providing also a graphical interface
 - Expressions and computations are natively supported (parametric diagrams become obsolete)
- Requirement entity
 - Describing requirements with quantitative parameters
 - Computing parameter values using formulas
 - Provide features for their verification (under consultation)
- Still no clear support for integration of analysis results back into system model
 - The concept of a SysML analysis model is integrated
 - Feedback of analysis results to SysML model is needed



Future work

⇒ Ethical concerns as requirements in MBSD¹

- IEEE standard issued in 2021, OMG announced a working group on Responsible Computing in 2022
- Case study: Managing the environmental footprint of electronic services and datacenters

⇒ QoS-aware Cloud engineering

- Enable providers/customers assess solutions cost-wise
- Support revenue/profit analysis

¹IEEE 7000-2021 - IEEE Standard Model Process for Addressing Ethical Concerns during System Design. 2021. URL: <https://standards.ieee.org/standard/7000-2021.html>.



Thank you for your attention!

Acknowledgments

The research work was supported by EU Projects, QRF (Qatar Research Fund) and HFRI (Hellenic Foundation Research Institute)

Implementation

- Profiles and plugins are available at <https://bitbucket.org/ckotronis-phd/plugins/src/main/>
- A tutorial for using them is available at [QoS MBSD demo](#)
- Contact at kotronis@hua.gr

Recent Publications

1. Kotronis, C., Routis, I., Politi, E., Nikolaidou, M., Dimitrakopoulos, G., Anagnostopoulos, D., ... & Djelouat, H. (2019). Evaluating internet of medical things (IoMT)-based systems from a human-centric perspective. *Internet of Things*, 8, 100125.
2. Kotronis, C., Nikolaidou, M., Kapos, G. D., Tsadimas, A., Dalakas, V., & Anagnostopoulos, D. (2020). Employing SysML to model and explore levels-of-service: The case of passenger comfort in railway transportation systems. *Systems Engineering*, 23(1), 82-99.
3. Kapos G. -D., Tsadimas A., Kotronis C., Dalakas V., Nikolaidou M. and Anagnostopoulos D. A Declarative Approach for Transforming SysML Models to Executable Simulation Models (2021). *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3330-3345, June 2021, doi: 10.1109/TSMC.2019.2922153
4. Nikolaidou, M., Kotronis, C., Routis, I., Politi, E., Dimitrakopoulos, G., Anagnostopoulos, D., ... & Bensaali, F. (2021). Incorporating patient concerns into design requirements for IoMT-based systems: The fall detection case study. *Health informatics journal*, 27(1), 1460458220982640.
5. Kotronis, C., Tsadimas, A., Nikolaidou, M., Michalakelis, C., & Anagnostopoulos, D. (2022). Extending sysml to integrate cost analysis into model-based systems engineering. *IEEE Transaction of Engineering Management*, doi: 10.1109/TEM.2022.3200148.
6. Kotronis, C., Routis, I., Tsadimas, A., Kapos, G. D., Dalakas, V., Nikolaidou, M., & Anagnostopoulos, D. Engaging human participants in CPHS design: A model-based approach. *ACM Transactions on Internet of Things Journal*. Under review.