

---

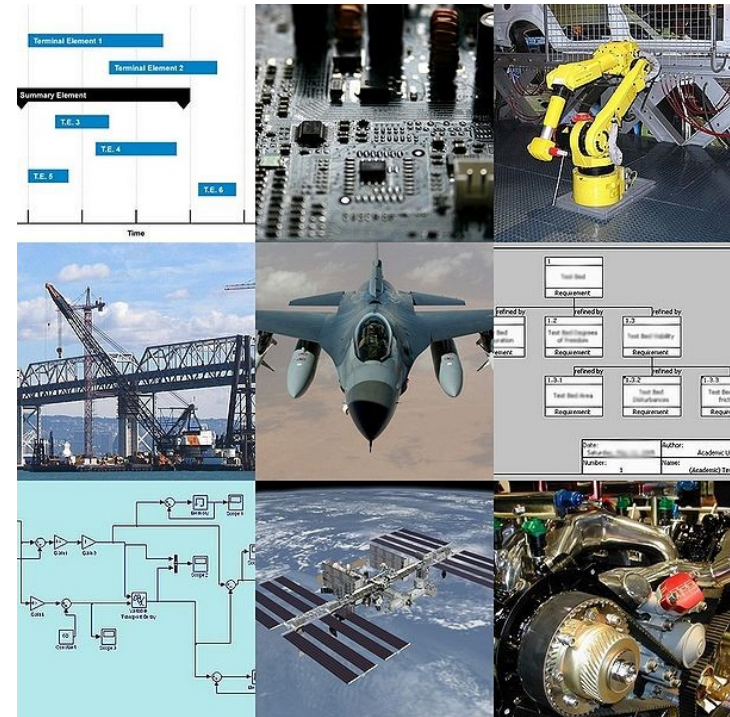
# SysML

## (System Modeling Language)

### Osnova:

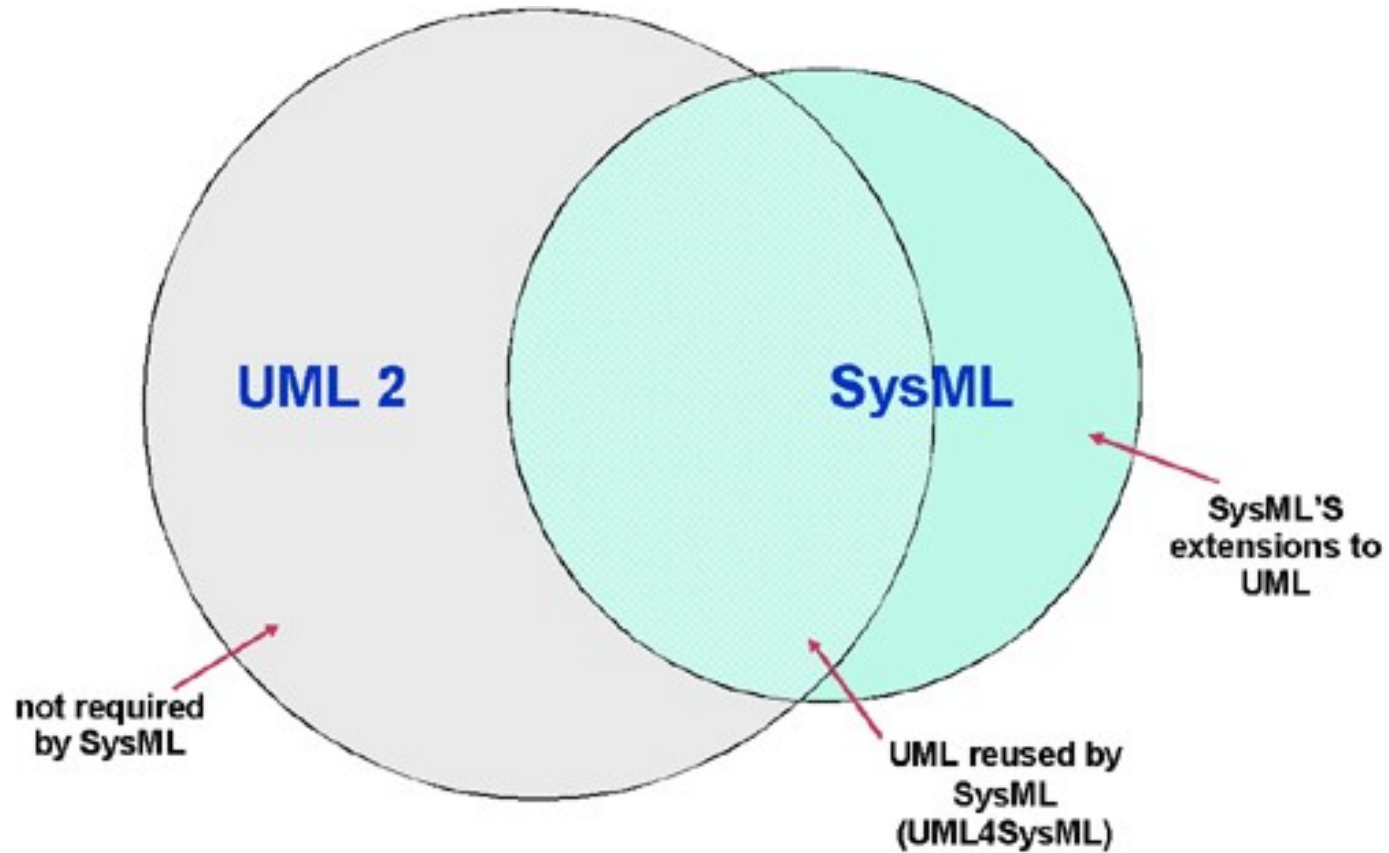
- Motivace: proč další modelovací jazyk?
- SysML vs. UML
- SysML v příkladech
- Zhodnocení a závěr

- SysML je grafický modelovací jazyk vycházející z jazyka UML
- UML je zaměřený na návrh SW systémů (od mobilních zařízení po enterprise řešení)
- SysML je zaměřený na systémové inženýrství (splňuje specifikaci SE RFP – Systems Engineering Request for Proposal)
  - Embedded systems
  - Real-time systems
  - Complex engineering projects



- Vznik UML:
  - 1996: První verze UML 0.9
  - 1997: UML přebírá OMG (stává se standardem)
  - 2005: UML 2
- Vznik SysML:
  - 2003: založeno sdružení SysML Partners
  - 2005: SysML 1.0a předložena OMG (Object Management Group). Od té doby OMG udržuje OMG SysML, včetně open source licence. Poslední verze je OMG SysML 1.1
- SysML je definováno jako profil UML
  - SysML je nedefinováno pomocí UML
- Profil UML
  - Mechanismus UML jak přizpůsobit modely pro konkrétní doménu (leťectví, zdravotnictví, finance, ...) a/nebo platformu (J2EE, .NET, ...)
  - Děje se to prostřednictvím stereotypů, tagů a omezení aplikovaných na konkrétní UML elementy (třídy, atributy, aktivity apod.)

# UML vs. SysML (pokr.)



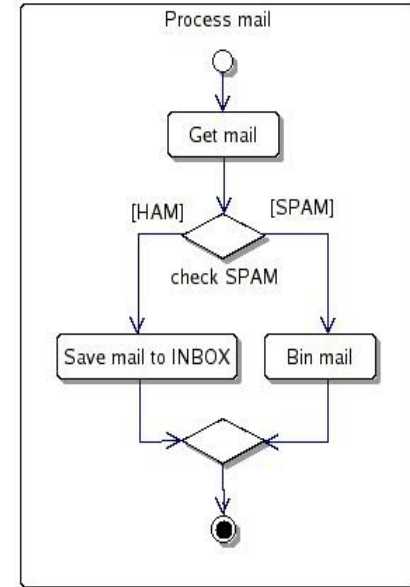
# Výhody SysML



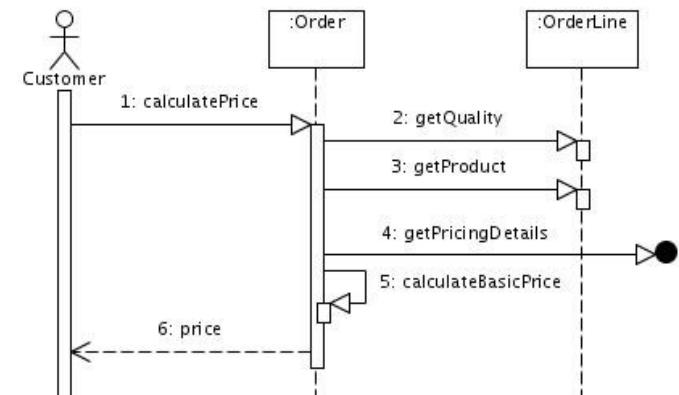
- Je menší a jednodušší než UML (menší počet diagramů a pojmů obecně).
- Redukuje orientaci na SW a přidává modelování HW požadavků, výkonu apod.
- Přidává obecné alokační tabulky, např. pro alokaci požadavků, funkcí, struktury, a tím ulehčuje automatizovanou verifikaci a validaci.

SysML	UML
Activity Diagram	Activity Diagram
Block Definition Diagram	Class Diagram
Internal Block Diagram	Composite Structure Diagram
Package Diagram	Package Diagram
Parametric Diagram	N/A
Requirement Diagram	N/A
Sequence Diagram	Sequence Diagram
State Machine Diagram	State Machine Diagram
Use Case Diagram	Use Case Diagram
Allocation Tables	N/A
N/A	Component Diagram
N/A	Communication Diagram
N/A	Deployment Diagram
N/A	Interaction Overview Diagram
N/A	Object Diagram
N/A	Timing Diagram

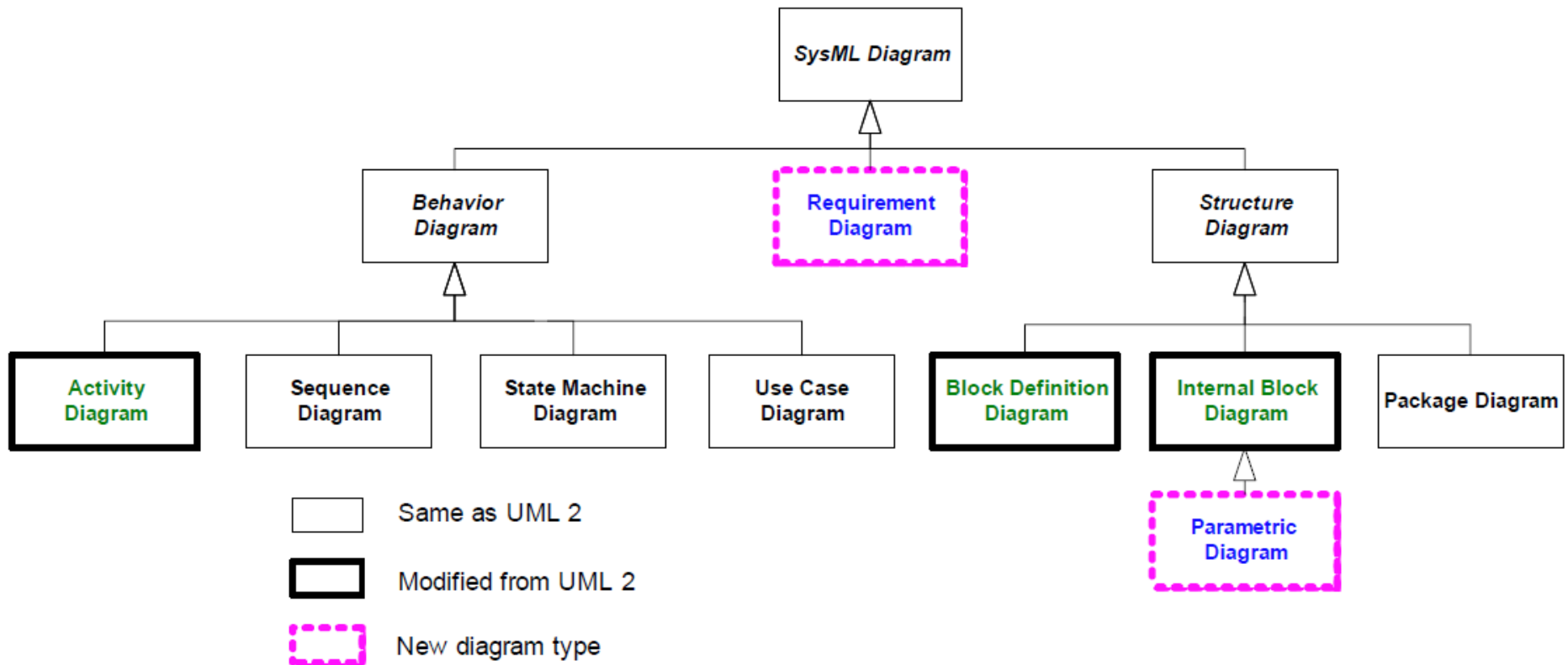
UML Activity Diagram:



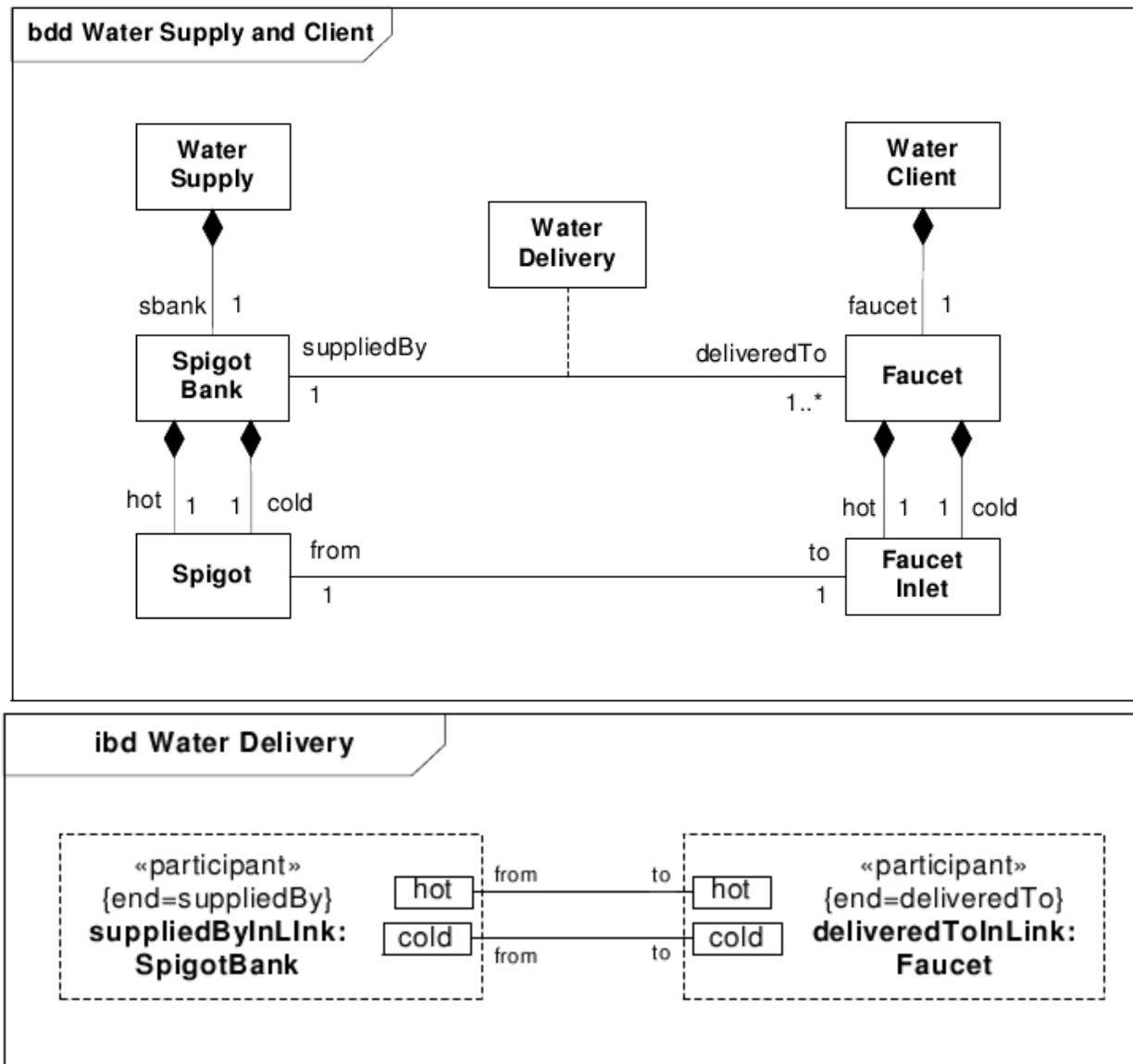
UML Sequence Diagram:



# Členění SysML diagramů



# Block Definition and Internal Block Diagrams





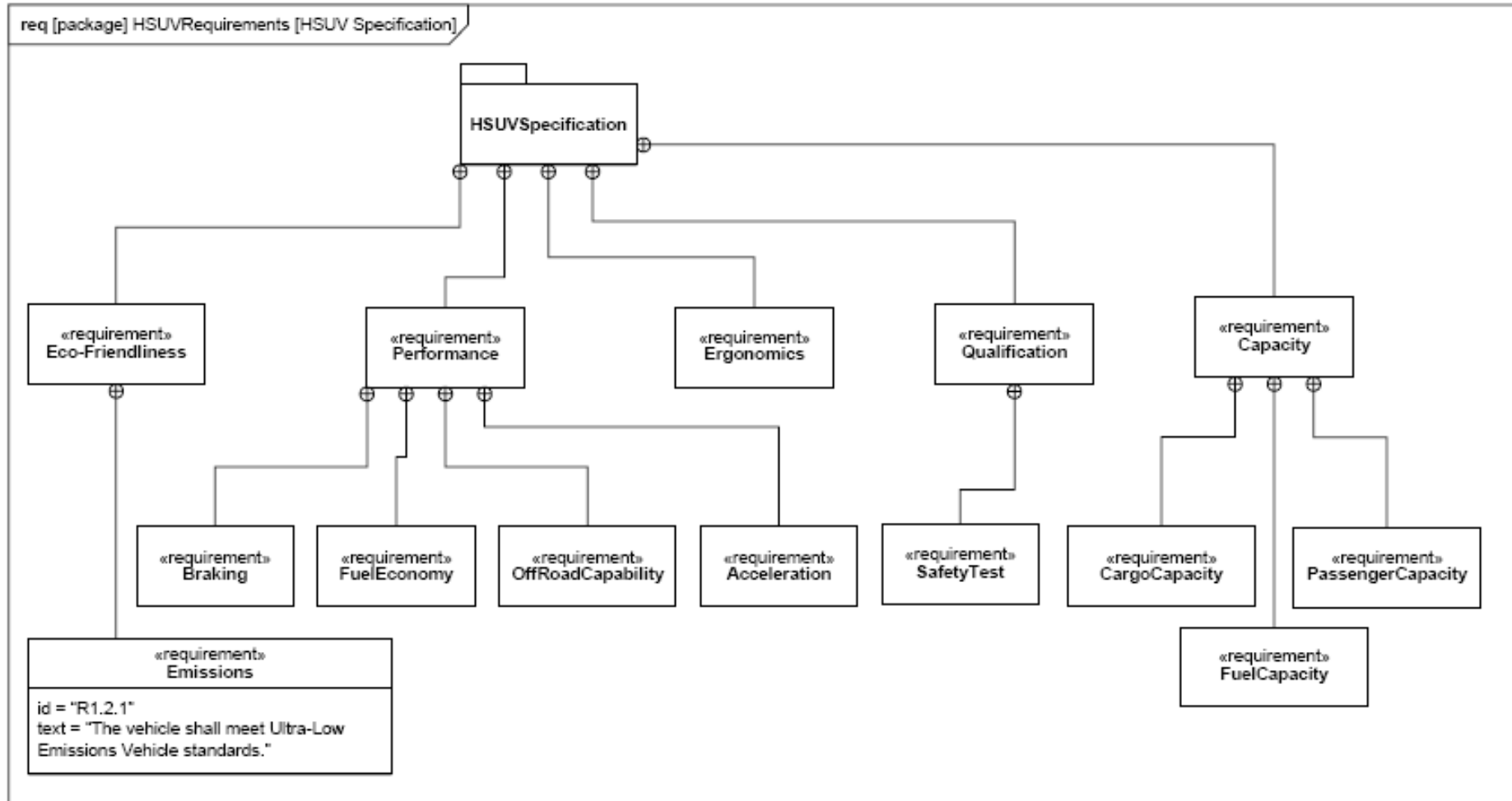
# Requirement Diagram

- *Diagram požadavků*
- Ukazuje systémové požadavky a jejich vztahy s jinými elementy
- Obsahuje případy užití (use-case), požadavky (requirement), balíky (package), testovací případy (test-case) a zdůvodnění (rationale)
- Výše uvedené elementy jsou propojeny omezujícími vztahy (deriveReq, refine, verify, trace, ...)
- Vhodný pro analýzu požadavků (requirement analysis).

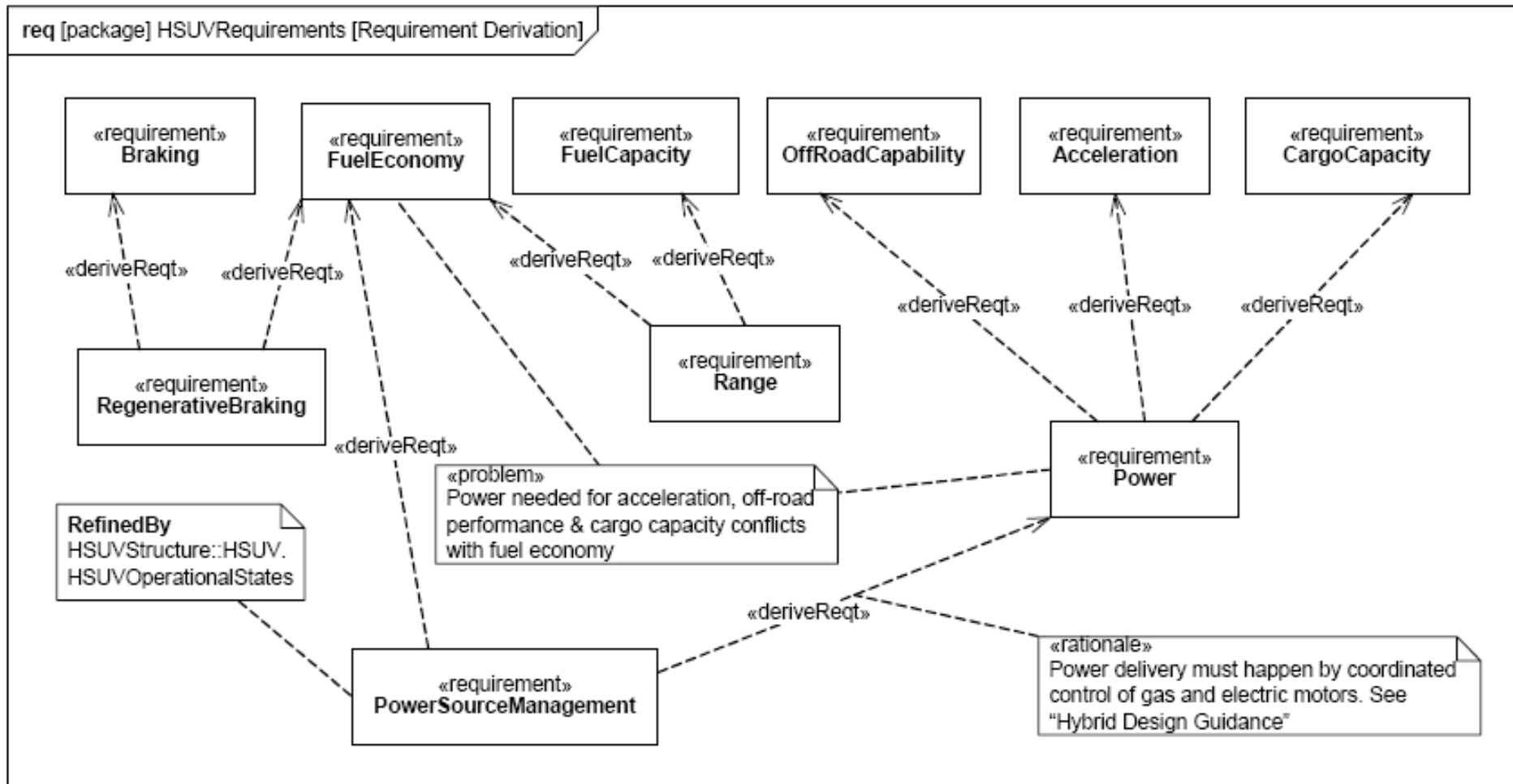
3 February  
Project status:  
100% complete



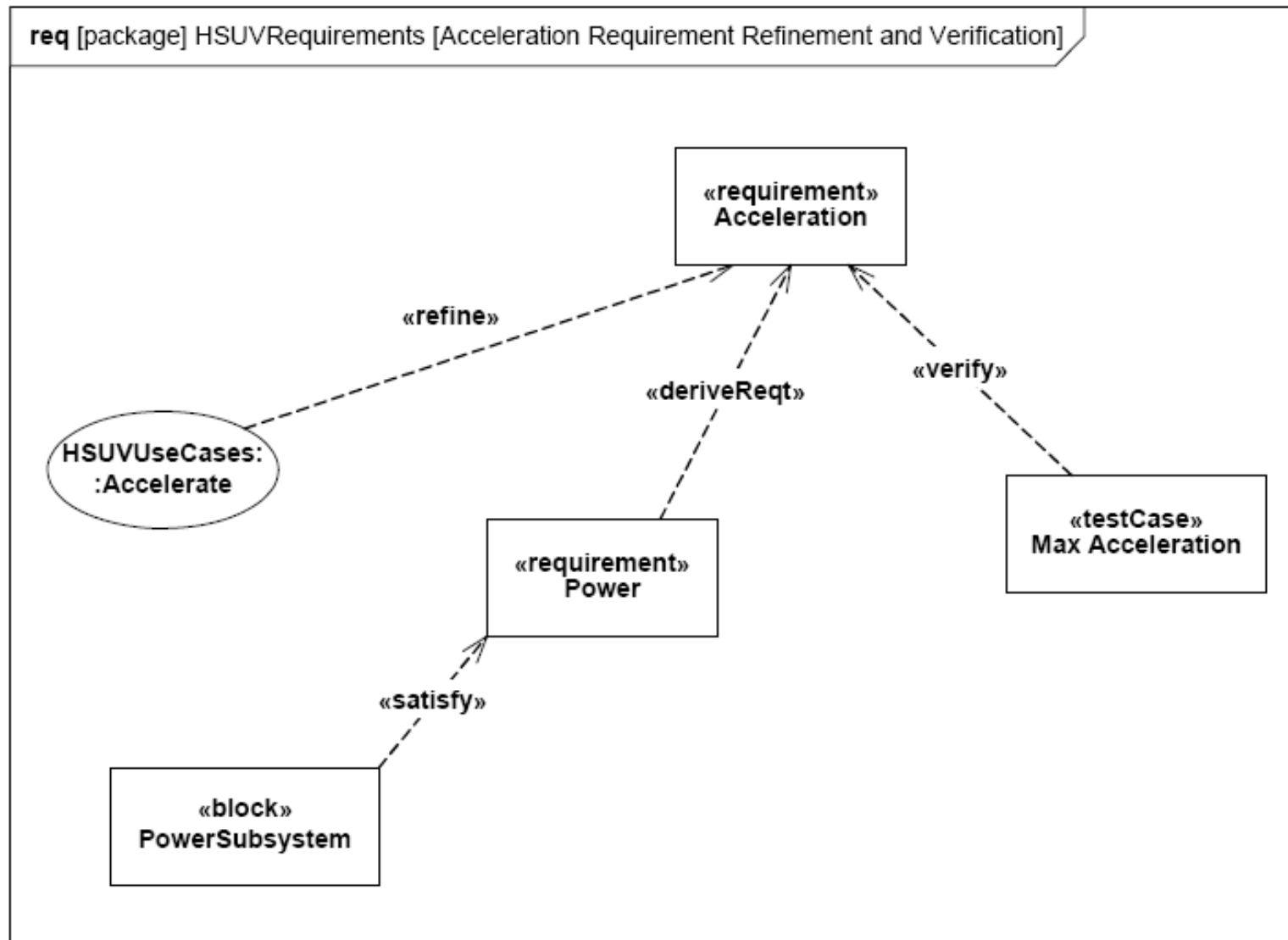
# Requirement Diagram – příklad 1



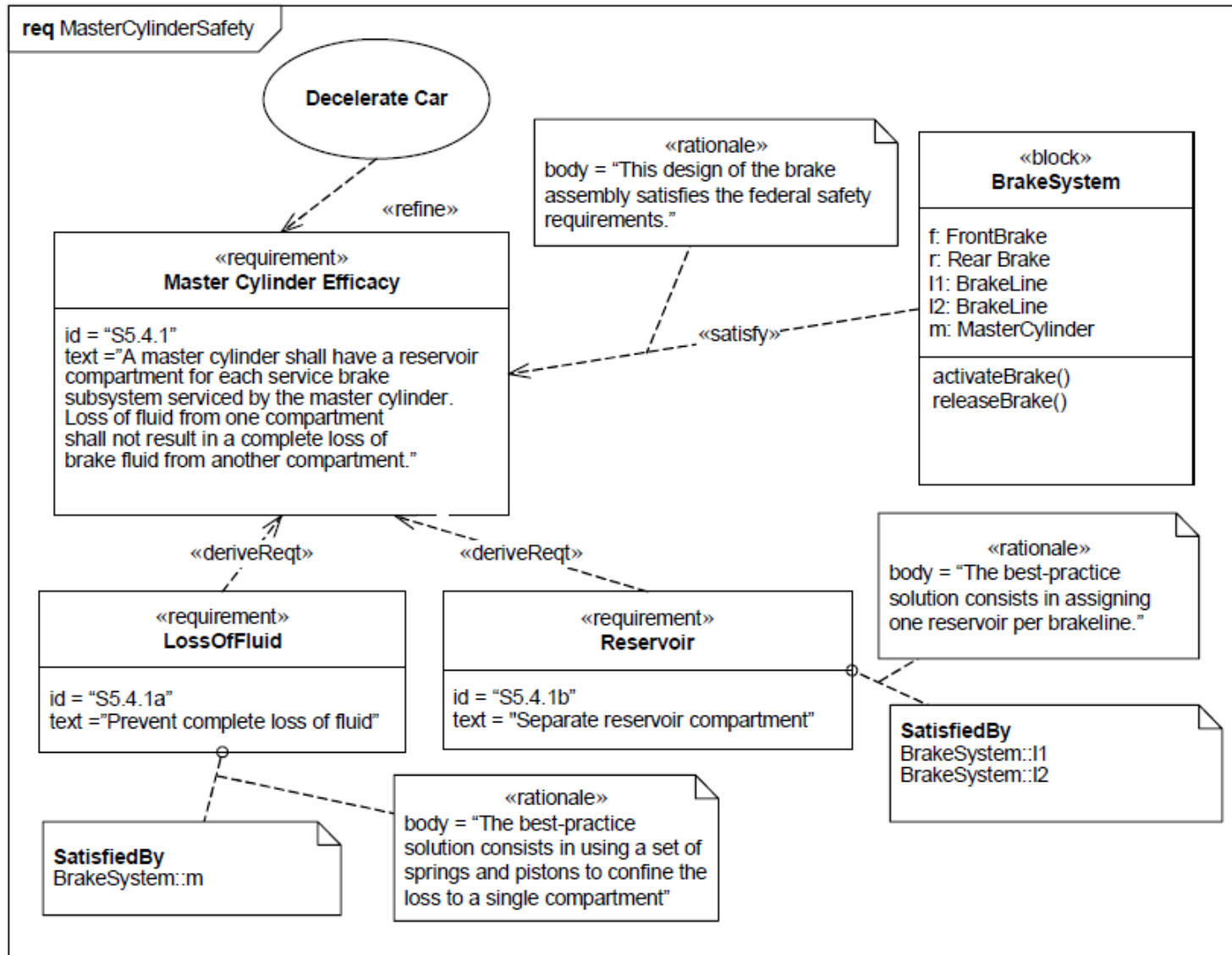
# Requirement Diagram – příklad 1 (pokr.)



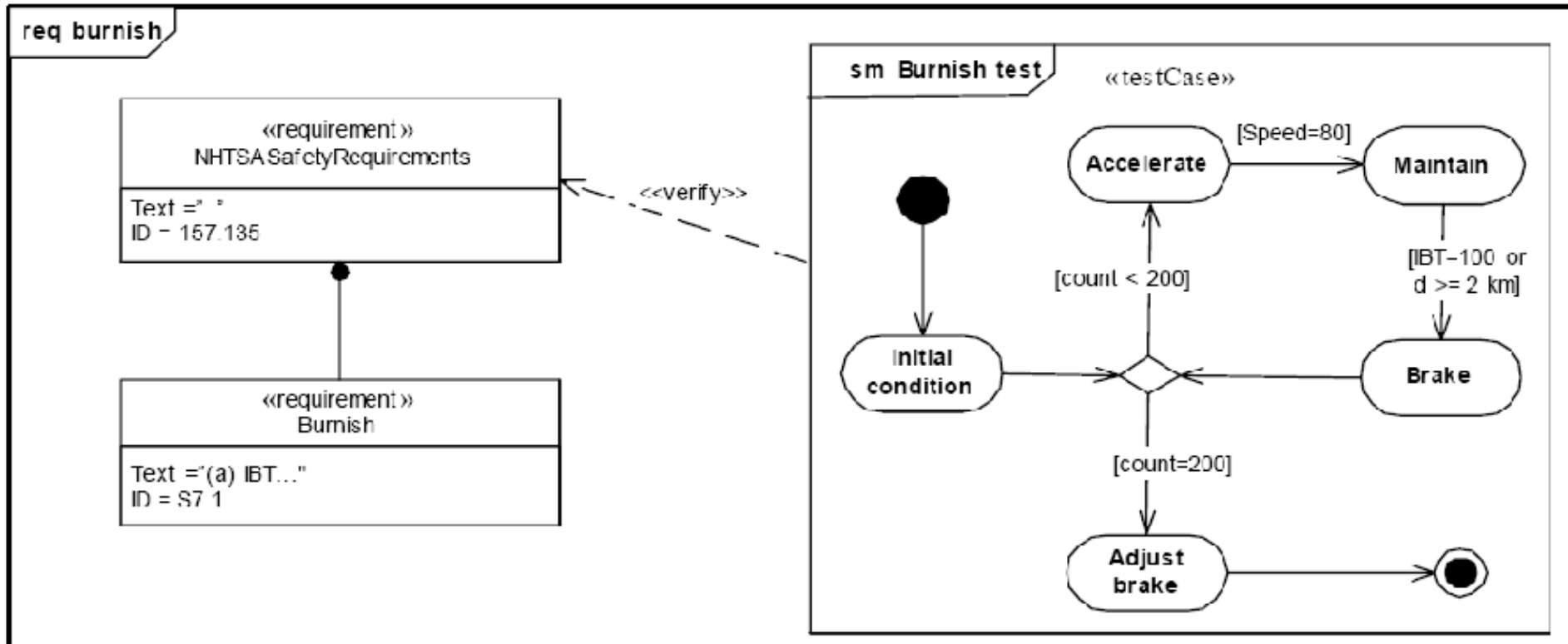
# Requirement Diagram – příklad 1 (pokr.)



# Requirement Diagram – příklad 2

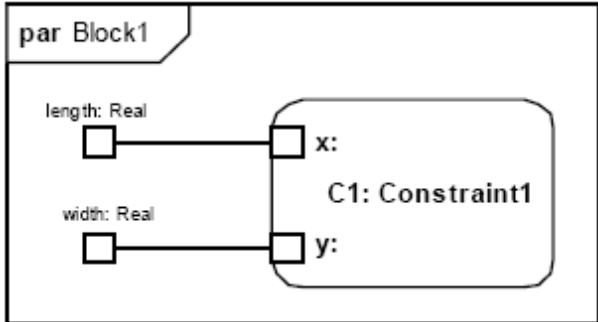
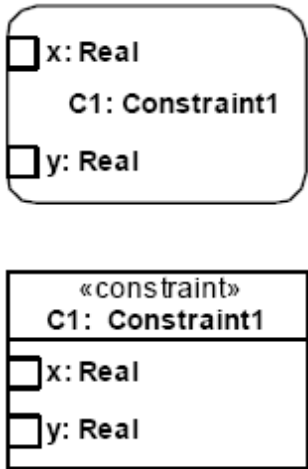


# Requirement Diagram – příklad 2 (pokr.)



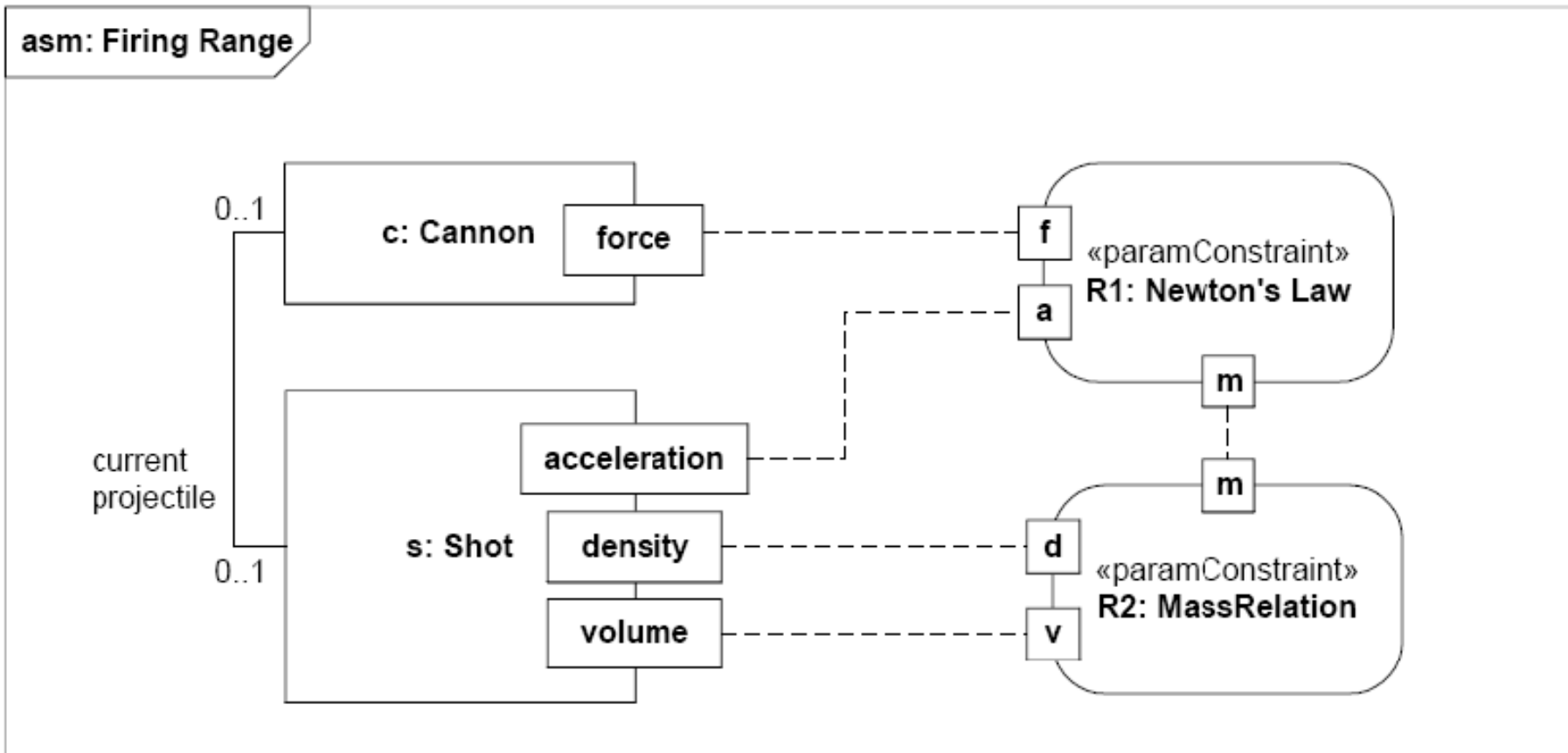
- *Diagram požadavků*
- Ukazuje parametrická omezení mezi elementy.
- Vhodný pro výkonnostní a kvantitativní analýzu.
- Je definovaný jako omezená forma diagramu interních bloků.
- Obsahuje interní bloky a jejich vnitřní vlastnosti + omezující vlastnosti a jejich parametry.
- Všechny vlastnosti (kromě samotných omezení) musí být ohraničeny buďto přímo omezujícím parametrem, nebo nepřímo ostatními vlastnostmi a omezeními.
- Přináší constraint-logic paradigm do object-oriented paradigm.

# Parametric Diagram – příklad syntaxe

Element Name	Concrete Syntax Example	Metamodel Reference
<b>ParametricDiagram</b>		SysML::Constraint-Blocks::ConstraintBlock SysML::Blocks::Block
<b>ConstraintProperty</b>		SysML::ConstraintBlocks::ConstraintProperty



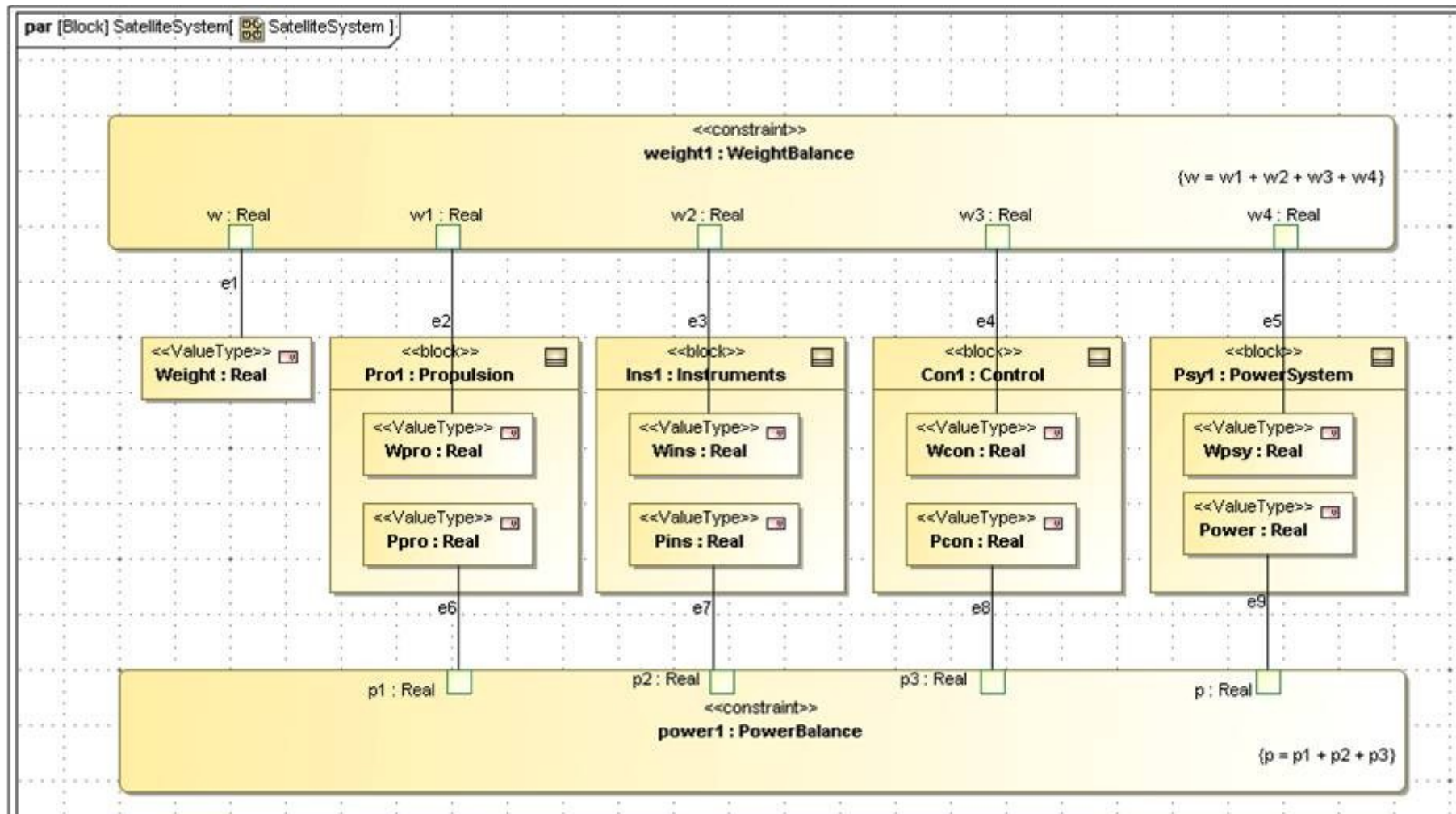
# Parametric Diagram – příklad 1



## Druhý Newtonův zákon:

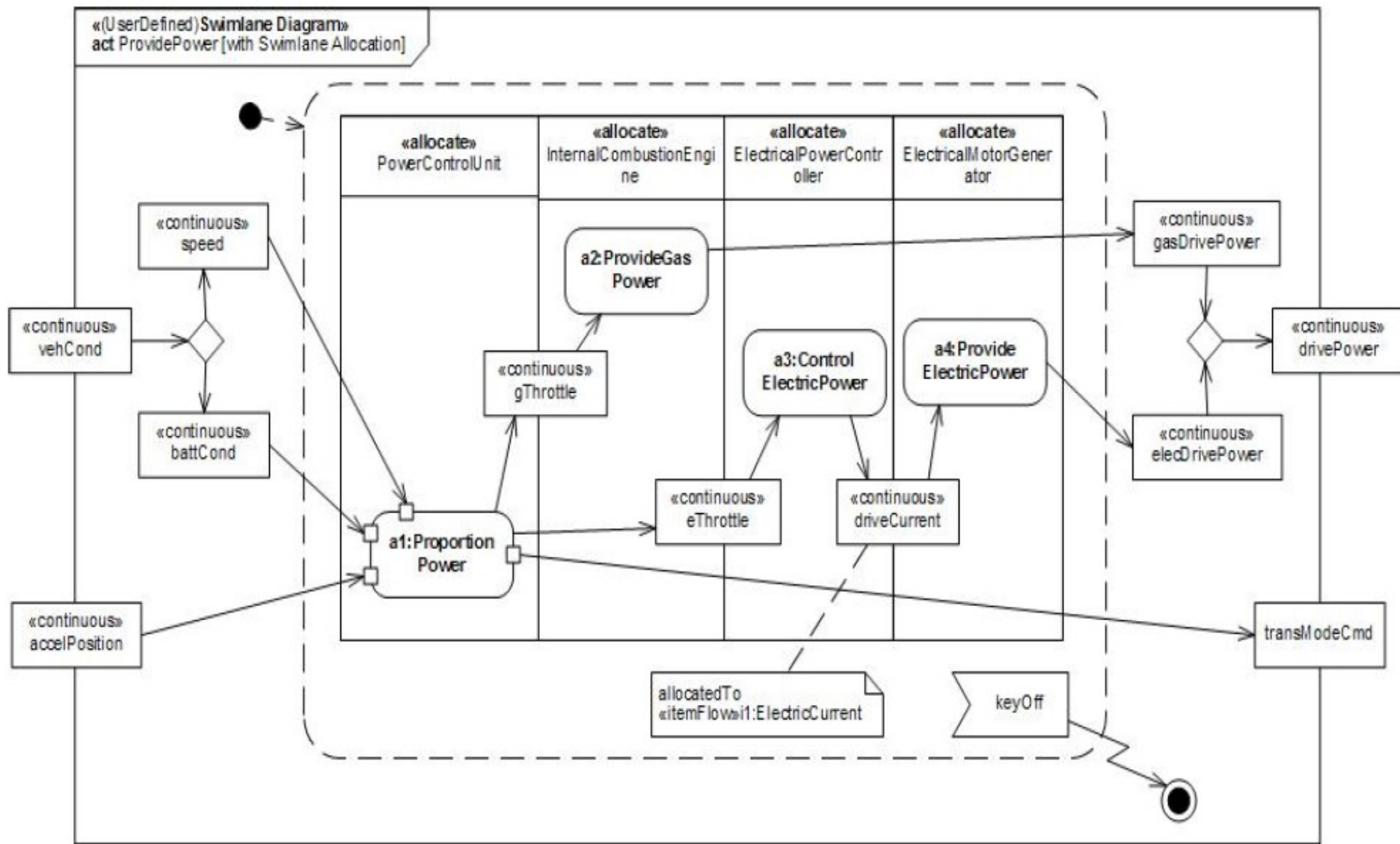
Jestliže na těleso působí **síla**, pak se těleso pohybuje se **zrychlením**, které je přímo úměrné působící síle a nepřímo úměrné **hmotnosti** tělesa

# Parametric Diagram – příklad 2



- *Alokační tabulky*
- Tabulkový zápis alokací (mapování) elementů. Hojně využíváno právě v systémovém inženýrství.
  - Requirement allocation
  - Functional allocation
  - Structural allocation
  - ...
- Vhodné pro ulehčení automatické verifikace a validace

# Allocation Tables – příklad (d. aktivit)



# Allocation Tables – příklad (pokr.)

table [activity] ProvidePower [Allocation Tree for Provide Power Activities]

type	name	end	relation	end	type	name
activity	a1:ProportionPower	from	allocate	to	block	PowerControlUnit
activity	a2:ProvideGas Power	from	allocate	to	block	InternalCombustionEngine
activity	a3:ControlElectricPower	from	allocate	to	block	ElectricalPowerController
activity	a4:ProvideElectricPower	from	allocate	to	block	ElectricalMotorGenerator
objectNode	driveCurrent	from	allocate	to	itemFlow	i1:ElectricCurrent

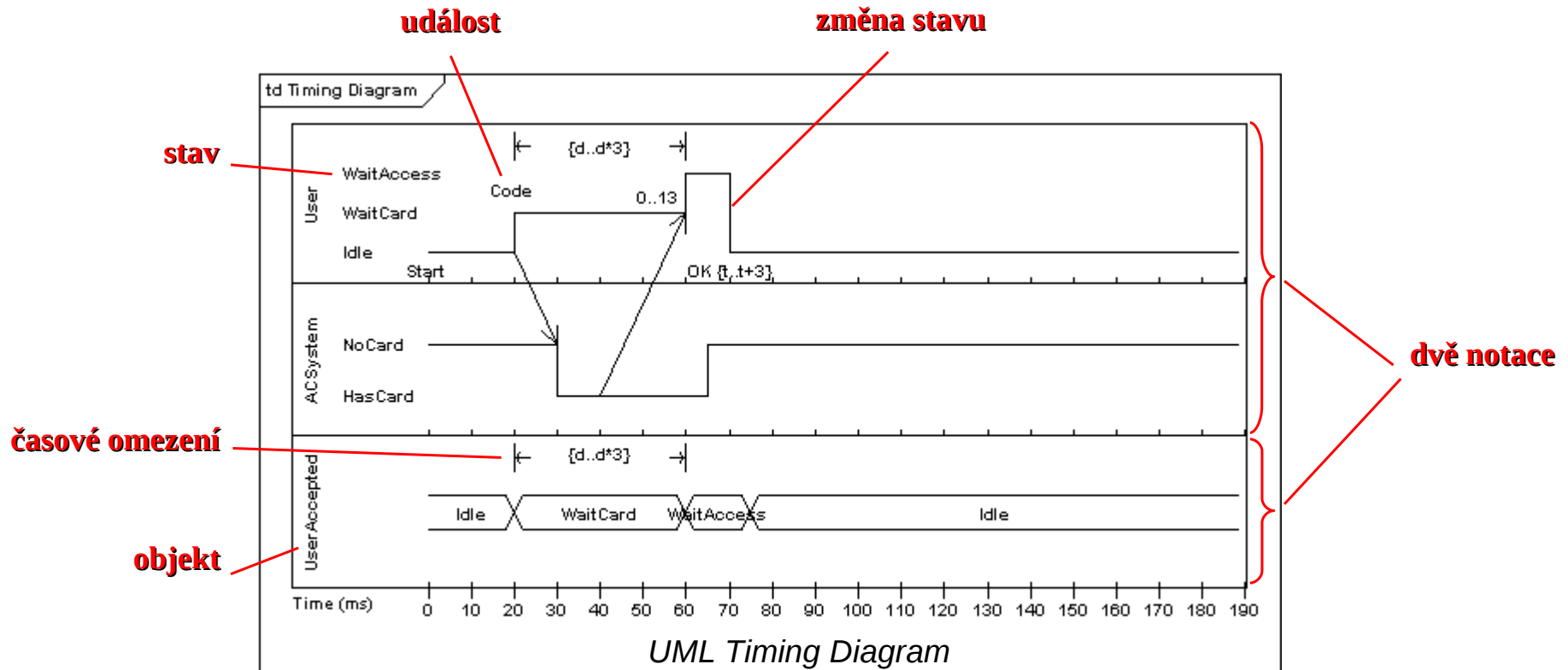
matrix [activity] ProvidePower [Allocation Tree for Provide Power Activities]

Source	Target				
	PowerControlUnit	InternalCombustionEngine	ElectricalPowerController	ElectricalMotorGenerator	I1:ElectricCurrent
A1:ProportionPower	allocate				
A2:ProvideGasPower		allocate			
A3:ControlElectricPower			allocate		
A4:ProvideElectricPower				allocate	
driveCurrent					allocate

- Modely SysML nepracují s pojmy třída a její instance, ale tyto dva pojmy spojují do pojmu *block*.
- Všechny UML diagramy pro práci s instancemi byly ze SysML vypuštěny, přebrané diagramy byly omezeny
  - Např. sekvenční diagram pracuje s bloky. Sémantika se dá přirovnat k modelování anonymních instancí v klasickém UML sekvenčním diagramu.
- Mnozí se shodují: Modelování instancí je nezbytnou součástí modelování systémů a jejich absence v SysML je považována za velký problém.
- V současnosti se nevede ani tak diskuse o tom, jestli instance do SysML přidat, ale spíše jak je přidat.

# Problém absence diagramu časování

- Mnoho projektů řešených pomocí SysML je real-timových
- Návrháři SysML přesto tvrdí, že digramy časování (timing diagrams) nejsou potřeba. Dají se nahradit např. časovými omezeními v diagramu aktivit.
- Vede se diskuse, jestli jsou diagramy časování třeba, nebo ne.



# Lze použít UML a SysML společně?



- Ano, byl to záměr návrhářů SysML.
- V CASE nástrojích by neměl být problém (SysML je UML profil).
- Je zde ale řada otázek:
  - I společné SysML a UML diagramy se drobně liší např. v pojmech, což může vést ke zmatení pojmů a k problémům v komunikaci uvnitř týmu
  - Parametrické diagramy přinášejí prvky constraint-logic paradigm do objektového světa, ale sémantika je definována nejasně.
  - SysML stírá rozdíly mezi třídou a instancí.
  - Neexistuje metodika, která by tuto problematiku seriózně pokrývala.



# CASE systémy s podporou SysML

---



- Enterprise Architect + MDG Technology for SysML
  - Sparx Systems
- MagicDraw + SysML plugin
  - No Magic
- Artisan Studio
  - Artisan Software
- SysML Toolkit (EmbeddedPlus) + Rational Software Modeler/Architect
  - IBM
- Rhapsody
  - IBM/Telelogic
- TAU G2
  - IBM/Telelogic

Děkuji za pozornost