

IV101 – Modelování

Jiří Barnat

Modelování obecně

dr. N. Sidorova, Lecture on Process Modelling

<http://www.win.tue.nl/~sidorova/pm/>

Proč vytvářet modely

Obecná motivace

- Snaha lépe **porozumět**, **popsat** nebo **předpovědět** jak věci fungují v reálném světě s pomocí zkoumání zjednodušené reprezentace daného objektu či fenoménu.

Technologické důvody

- Změna formátu nebo vytvoření reprezentace pro další práci s modelovaným systémem.
- Existují jevy, které nelze vyjádřit v daném formátu, je nutné se s nimi nějak vypořádat.

Kdy je model užitečný?

- Pokud s jeho pomocí získáme nové poznatky.

Co je to model

Model

- je pohled na (možná imaginární) realitu.
- je reprezentace některých aspektů reality.
- je zjednodušení reality.
- je formalizovaná a zjednodušená reprezentace reality, kterou lze **snadno** studovat.
- opomíjí nezajímavé detaily a naopak vyzdvihuje důležitá fakta, která napomáhají pochopení studované reality jako celku.

Proces modelování

- Identifikace faktů, které mohou být zanedbány, a faktů které naopak musí být zahrnuty.
- Získání modelu, který je jednoduchý a přitom dostatečně přesný vzhledem ke studovaným aspektům reality.

Základní kroky při řešení problému modelováním

1. Získat jasnou představu o problému.
2. Vybrat, co možná nejvíce vhodné modelovací médium a vytvořit odpovídající model.
3. Vyřešit problém s využitím modelu.
4. Přenést řešení problému zpět na původní rovinu.
5. Ověřit řešení, je dostatečné i pro problém v původním znění?

Pozor

- Problémy s vlastním procesem modelování mohou ve výsledku potlačit původně řešené problémy, nebo učinit nalezená řešení neplatnými pro původní problém.

Procesy modelu

Procesy

- Základní stavební kameny modelu.

Co je to proces

- Jakákoliv množina aktivit organizovaná za společným účelem.
- Aktivity procesu mohou být přirozeně se vyskytující, ale i uměle dodané/navržené.
- Aktivity typicky probíhají v čase a prostoru, konzumují/produkují nějaké zdroje.
- Proveditelnost dané akce procesu může být podmíněna konkrétními vstupními podmínkami.
- Proces jako celek interaguje s jinými (sub)procesy v celku.
- Process jako celek může/nemusí mít interní stav.

Validace modelu a verifikace

Validace modelu

- Je model v dostatečném souladu s realitou, tak aby výsledky práce s modelem byly přenositelné do původní roviny?

Verifikace modelu

- Vykazuje skutečně model ty vlastnosti, které o něm předpokládáme?

Verifikace validního modelu \sim ověřování modelované reality.

Modelování systémů

Souvislost modelování a distribuovaných systémů

Distribuované systémy

- Distribuovaný systém je tvořen souborem vzájemně oddělitelných (identifikovatelných) procesů, které spolu komunikují výměnnou zpráv.

Modelování

- Identifikace procesů, komunikačních kanálů a komunikačních zpráv nezbytných pro fungování systému jako celku.
- Pokud systém vyžaduje pro svůj vznik nebo existenci komunikaci s procesy mimo sebe sama, jedná se o **otevřený** systém. V opačném případě se jedná o **uzavřený** systém.
- Modelováním prostředí vytváříme z otevřeného systému uzavřený distribuovaný systém.

Jak můžeme vytvořit model

Obecný vzor tvorby modelu sleduje základní pravidlo

1. Co je požadováno
2. Jak se toho docílí

Postup řešení

- Víme, co chceme (jaké vlastnosti má systém splňovat), ale nemáme jasnou představu o tom, jak systém má fungovat.
- Bližší představu o tom, jak má systém fungovat, lze získat analýzou případů použití (use cases).

Případy použití

Tvořeny

- seznamem událostí přicházejících z vnějšku systému,
- odpovídajícími očekávanými reakcemi systému,
- scénářem chování systému ilustrujícím klíčové charakteristiky procesů a jejich funkcionality.

Míra detailů a způsob vyjádření

- by měly být taková, aby bylo možné případ použití validovat.

Případy použití a Workflow diagramy

Jak postupovat při budování případu použití

- Identifikace hlavních funkčních bodů.
- Identifikace participujících subjektů.
- Zakreslení povolených a explicitně zakázaných scénářů (posloupností akcí) pomocí Message Sequence Chart nebo podobných formalismů (UML).

Workflow diagramy

- Schéma fungování systému
- Vybudováno tak, aby bylo konzistentní s identifikovanými případy použití
- Základ pro modelování procesů v systému

Message Sequence Charts

MSC

- Grafické znázornění jistých omezení na možná chování distribuovaného systému.
- Zachycuje množinu povolených běhů systému.

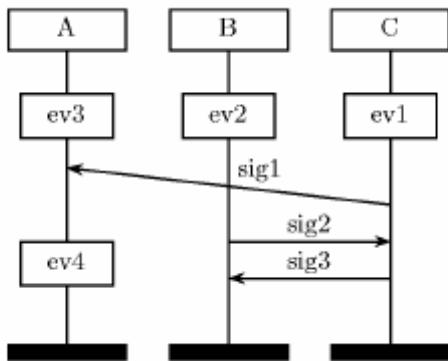
Čím je tvořeno

- Vertikální čáry reprezentují jednotlivé instance procesů.
- Šipky mezi čarami reprezentují signály/zprávy.
- Boxy na čarách reprezentují interní akce procesů.

Uspořádání událostí

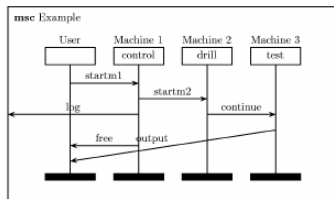
- Odeslání signálu musí předcházet jeho přijetí.
- Událostí v jedné instanci jednoho procesu jsou uspořádány směrem shora dolů.

Message Sequence Chart – příklad

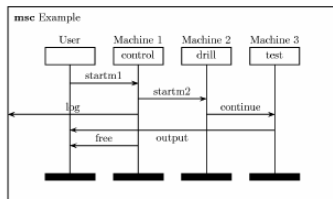


| | |
|-------|----|
| ev3 | @A |
| ev1 | @C |
| ev2 | @B |
| sig1/ | @C |
| sig2/ | @B |
| /sig2 | @C |
| sig3/ | @C |
| /sig1 | @A |
| ev4 | @A |
| /sig3 | @B |

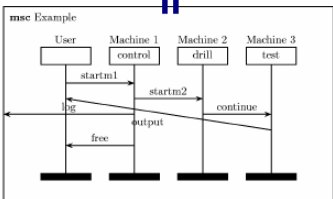
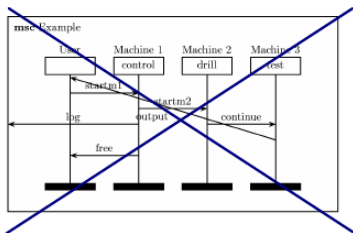
Message Sequence Chart – příklad



≠



||



Události v systému

Charakterizace dle typu

- Externí události.
 - Podněty přicházející z vnějšku systému.
- Časové události.
 - Způsobené dosažením jistého časového okamžiku.
 - Periodické události, časové limity.
- Stavové události.
 - Události iniciované nabytím určitého stavu.
 - .
- Systémové události.
 - Logování, test integrity dat, zálohování a pod.
 - Většinou se nemodelují, neboť nepřispívají k primárním funkcím systému.

Čím je dána událost

Při modelování událostí systému je potřeba mít jasno o následujících aspektech události

- Jak se pozná, že událost nastala.
- Kdo/Co je zdrojem dané události.
- Jaká aktivita v systému je svázaná se zpracováním dané události.
- Jaký je (pokud je) dopad na zbytek systému, tj. v co vyústí zpracování této události.
- Komu jsou případná výstupní data či výstupní událost určeny.

Meziprocesní komunikace

Souběžné procesy

- Procesy nazýváme souběžné, pokud dochází k překrytí časových intervalů existence těchto procesů.

Meziprocesní komunikace

- Přenos informace mezi souběžnými procesy.
- Samotný výskyt komunikace nese jistou informaci.

Realizace

- Komunikace přes sdílené proměnné.
- Komunikace posíláním zpráv.

Komunikace přes sdílené proměnné

Vlastnosti

- Asynchronní událost (nesynchronizuje pisaře a čtenáře).
- Bezčasovost akce
 - Samotná změna je bezčasová.
 - Čtenář není informován o změně hodnoty dané proměnné. Neví jak dlouho je daná proměnná nastavena na aktuální hodnotu.
- Přenos informace je bezporuchový, není vhodné na modelování komunikačních protokolů.

Komunikace posíláním zpráv

Charakterizace dle typu komunikace

- Synchronní předávání zpráv.
 - Odesílatel nedokončí posílání zprávy dříve, než ji adresát přijme.
- Asynchronní předávání zpráv.
 - Odesílatel dokončí posílání zprávy dříve, než ji adresát přijme, a může poslat několik dalších zpráv, než adresát přijme tu první.
 - Komunikační buffer.

Charakterizace dle specifikace adresáta

- Konkrétně určený bod v konkrétním procesu.
- Libovolný bod v konkrétním procesu.
- Libovolný proces.
- Vícenásobný adresát (všesměrové vysílání).

Abstrakce a ekvivalence

Pozorování a problém

Pozorování

- Reálné systémy nelze modelovat do všech detailů.
 - Nástroje pro práci s modely (model checkery) nejsou schopny takto velké systémy zpracovat.
 - Časová, personální i technická náročnost.
- K vybudování modelu se používá abstrakce.

Problém validního modelu

- Zachovává model při použití dané abstrakce požadované vlastnosti?

Ekvivalence modelů

Porovnání modelů

- Pro účely smysluplného porovnání modelů, je třeba na modely nahlížet unifikovaným způsobem.
- LTS – přechodové systémy s návěštím

Ekvivalence přechodových systémů

- Stopová ekvivalence
 - Maximální stopy (nelze je prodloužit).
 - Na systémech bez konečných stop zachovává vlastnosti specifikované pomocí Lineární Temporální Logiky (LTL).
- Bisimulace
 - Zachovává vlastnosti specifikované pomocí CTL*

Abstrakce

Abstrakce

- Snaha o nahrazení konkrétního detailu obecnějším popisem.
- Jestliže nějaká událost mění abstrahovaný detail, v modelu tato událost nezpůsobí žádnou změnu nebo pouze obecnou změnu.

Příklad

- Konkrétně: $x = 7$ Abstraktně $x \geq 0$
- $x = x + 3$
- Konkrétně: $x = 10$ Abstraktně $x \geq 0$
- $x = x * 2$
- Konkrétně: $x = 20$ Abstraktně $x \geq 0$
- $x = x/3$
- Konkrétně: $x = 6$ Abstraktně $x \geq 0$

Abstrakce a nedeterminismus

Abstrakce může vynutit nedeterministické chování modelu, neboť vzhledem k obecné reprezentaci, nemusí být výsledek akce dán jednoznačně.

Příklad

- Konkrétně: $x = 7$ Abstraktně $x \geq 0$
- $x = x + 3$
- Konkrétně: $x = 10$ Abstraktně $x \geq 0$
- $x = x - 5$
- Konkrétně: $x = 5$ Abstraktně $x \geq 0$ nebo $x < 0$
- $x = x * (-1)$
- Konkrétně: $x = -5$ Abstraktně $x \leq 0$ nebo $x > 0$

Abstrakce a aproximace

Aproximace chování

- Různým způsobem abstrahování můžeme obdržet model, jehož množina možných chování pouze “aproximuje” (není totožná) množinu chování reálného systémů.

Over-aproximace

- Množina chování modelu je **nadmnožinou** množiny chování původního systému.
- Chybový běh v modelu nevynucuje přítomnost chybového běhu v systému. (False negative.)

Under-aproximace

- Množina chování modelu je **podmnožinou** množiny chování původního systému.
- Nepřítomnost chybového běhu v modelu nevylučuje přítomnost chybového běhu v systému. (False positive.)

Typy abstrakcí

Predikátové abstrakce

- Konkrétní hodnoty proměnných v modelu jsou nahrazeny valuací predikátů nad těmito proměnnými ($x \geq 0$, je příklad predikátové abstrakce).

Abstrakce od reálného času

- Má-li náš model schopnost zachytit reálný čas události, můžeme modelovat události tak, že nastávají pouze v okamžicích odpovídajících celým jednotkám času.

Abstrakce od náhodnosti uspořádáním

- Abstrahujeme od možnosti náhodného pořadí událostí, a namodelujeme omezení na pořadí jejich výskytů.

Abstrakce od náhodnosti shlukováním

- Abstrahujeme od možnosti náhodného pořadí událostí tím, že namodelujeme tyto události jako jednu souhrnnou událost.