

UML slovníček

Diagram aktivit

- K zachycení business procesů (AS-IS a variant TO-BE) v rámci hledání business řešení
- Grafický záznam procesů je rychleji pochopitelný než textový popis
- Vizualizuje procesy
- Každý proces může být složen z dílčích sub-procesů

Typy uzlů:





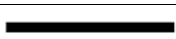
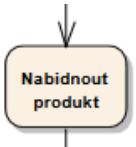

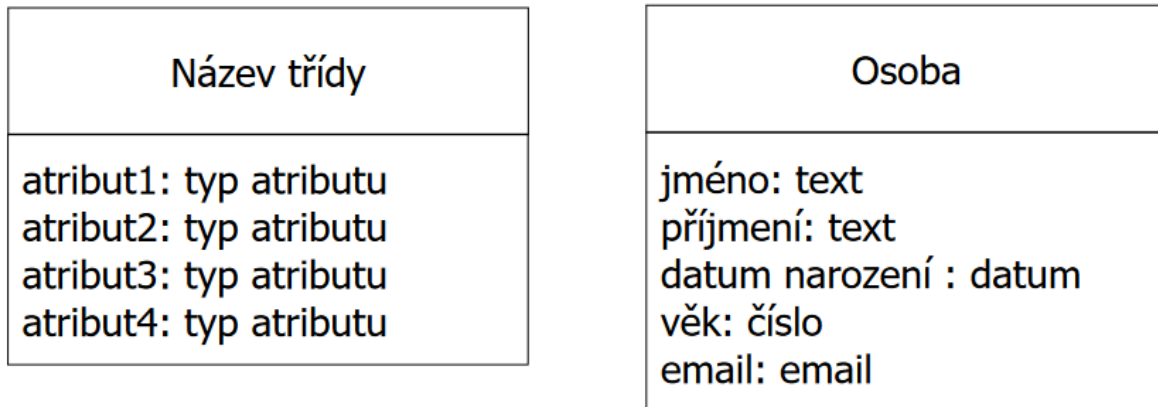
Startovní	
Konec aktivity	
Konec cesty (aktivita nekončí)	
Rozhodnutí / sloučení	
Rozvětvení / spojení	
Aktivita	
Objektový uzel (vstup / výstup aktivity)	

Diagram tříd

- Evidence business entit
 - o Definice entit se kterými řešení bude pracovat
 - o Definice atributů entit
- Evidence vztahů business entit
- Pojmenování podstatnými jmény
- Nepoužívat zkratky

UML notace třídy:



Vazby tříd:

Význam násobnosti: minimum..maximum	
0..1	nula nebo 1
1	právě 1
0..*	nula či více
*	nula či více
1..*	1 či více
1..6	1 až 6

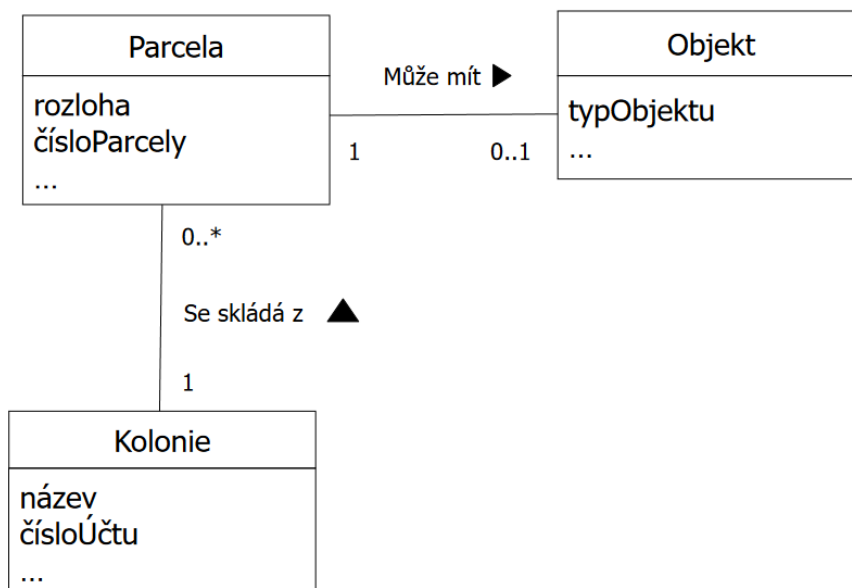


Diagram stavů

- Slouží k modelování životního cyklu entit
- Základní artefakty digramu:
 - o Začátek
 - o Konec (nemusí mít vždy)
 - o Stavy (podstav, nadstav)
 - o Přechody

Typy uzlů:

Začátek	
Konec	
Stav	
Přechod	

Příklad diagramu:

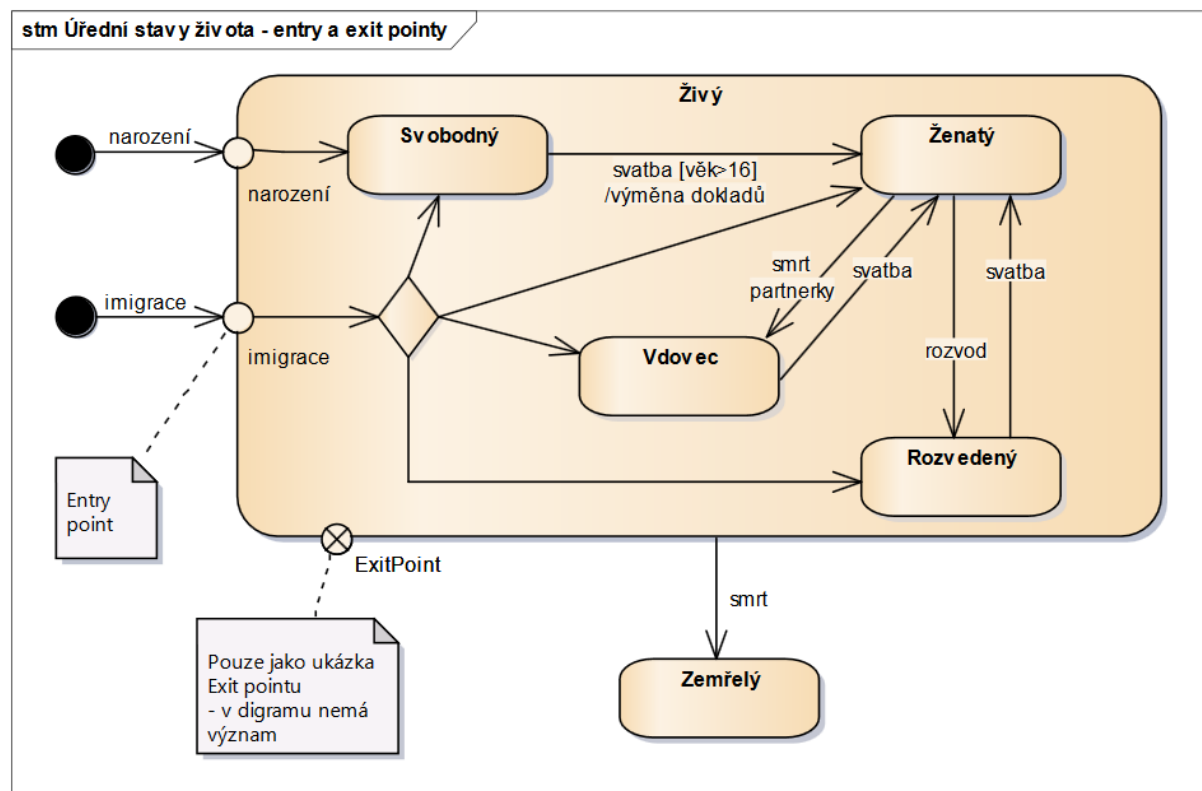
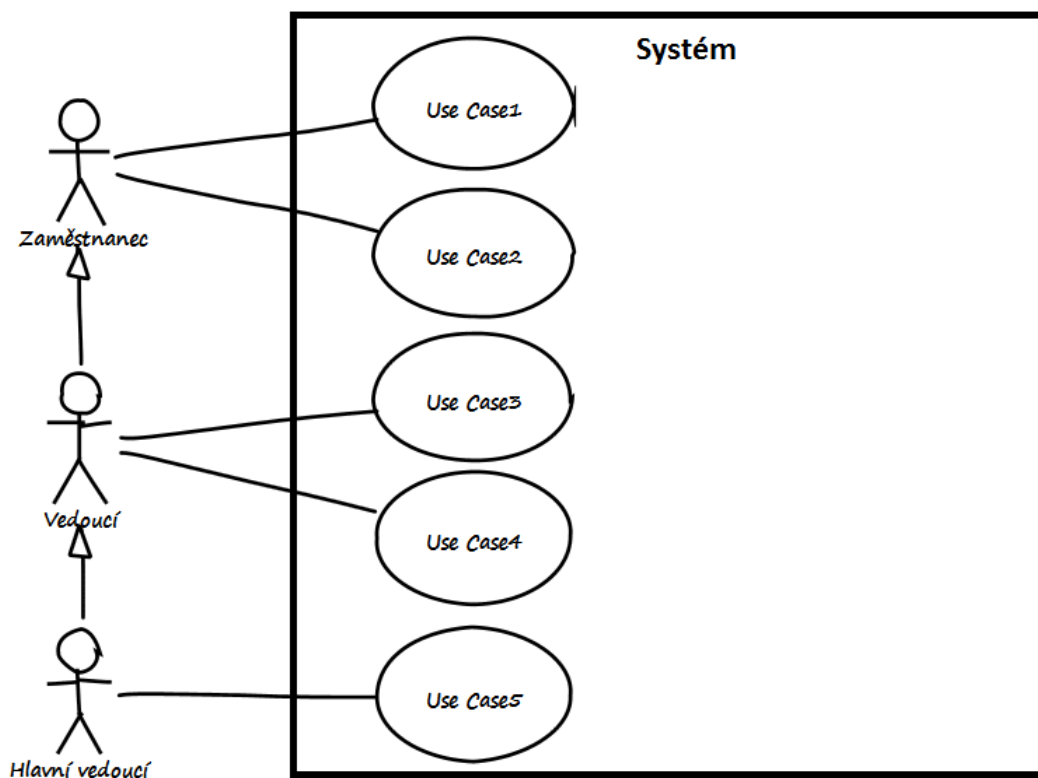


Diagram případů užití

- Vyjadřuje, kdo bude jakým způsobem používat systém = uživatele systému a jejich práva
- Modelování případů užití, je způsob zachycení pouze funkčních požadavků.
- Příklad užití popisuje jednu systémem podporovanou aktivitu obvykle jednoho účastníka v jednu chvíli.
- Pojmenovává funkce systému z pohledu účastníků. Tedy název odpovídá na otázku: „Co SYSTÉM umožňuje uživateli?“
- Název aktivity / use case má vždy slovesnou vazbu! (Zadat objednávku, Zjistit stav objednávky)
- Každý případ užití popisuje pouze jednu ze základních aktivit v systému (Vytvořit, zobrazit, upravit nebo smazat.)

Příklad diagramu



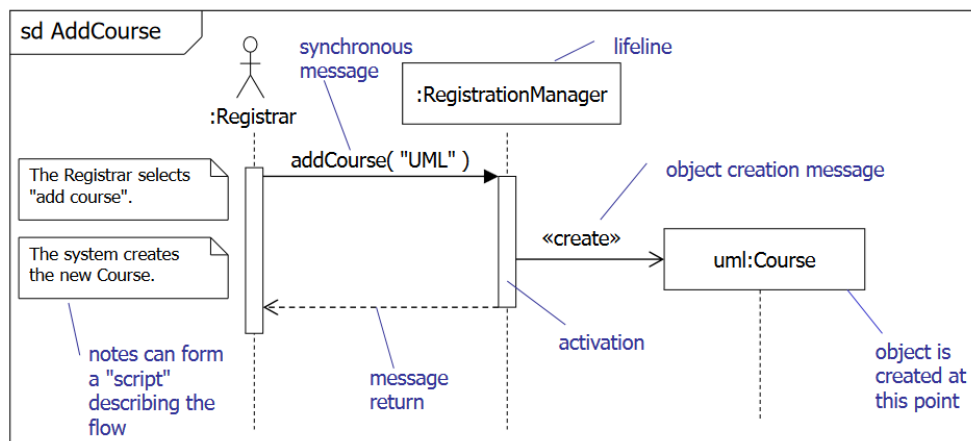
Sekvenční diagram

- kladou důraz na časové hledisko
- ukazují:
 - o kdo (instance tříd) se podílí na realizaci
 - o jaké zprávy a s jakými parametry si objekty vyměňují
 - o v jakém pořadí

Zprávy mezi instancemi

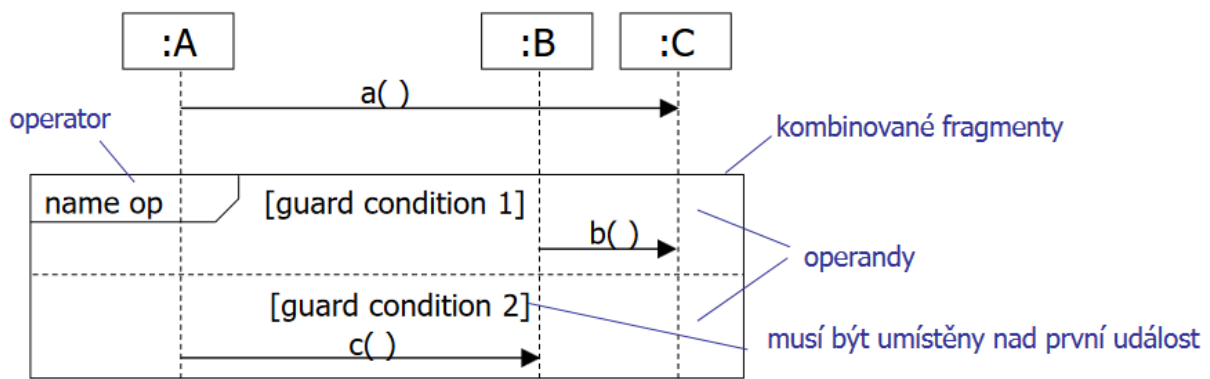
sender → receiver/target	typ	sémantika
	synchronní	odesílatel čeká na odpověď
	asynchronní	odesílatel nečeká na odpověď
	návrat	returning from a synchronous operation call návrat ze synchronního volání
	konstrukce	odesílatel vyvolá cíl
	destrukce	odesílatel destruuje cíl

Syntaxe



Kombinované fragmenty

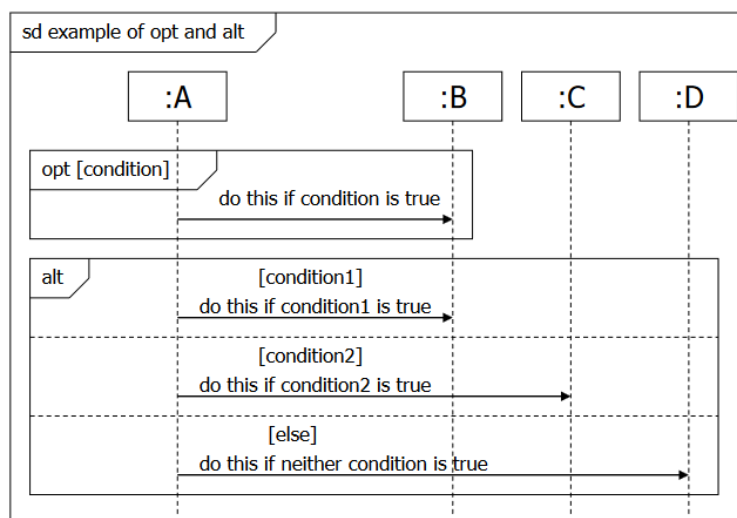
- Operator říká jak (režim) budou operandy spouštěny
- Guard condition – podmínka spuštění dané ho operandu/ů



Operátory

operator	long name	sémantika
opt	Option	if ...then...
alt	Alternatives	switch...case...
loop	Loop	loop min, max [podmínka] Iteruje minimálně <i>min times</i> , pokračuje do <i>max times</i> dokud podmínka splněna
break	Break	zbytek je přeskočen
ref	Reference	odkaz na jinou interakci= volání procedury

Větvení pomocí opt a alt



Iterace s loop

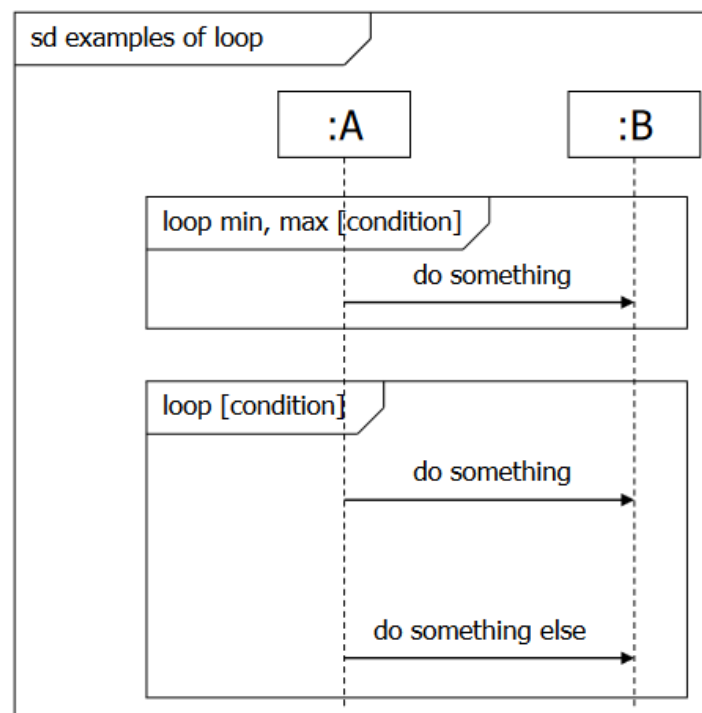
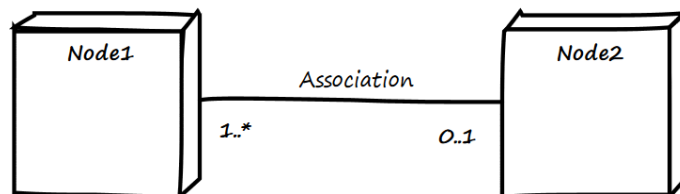


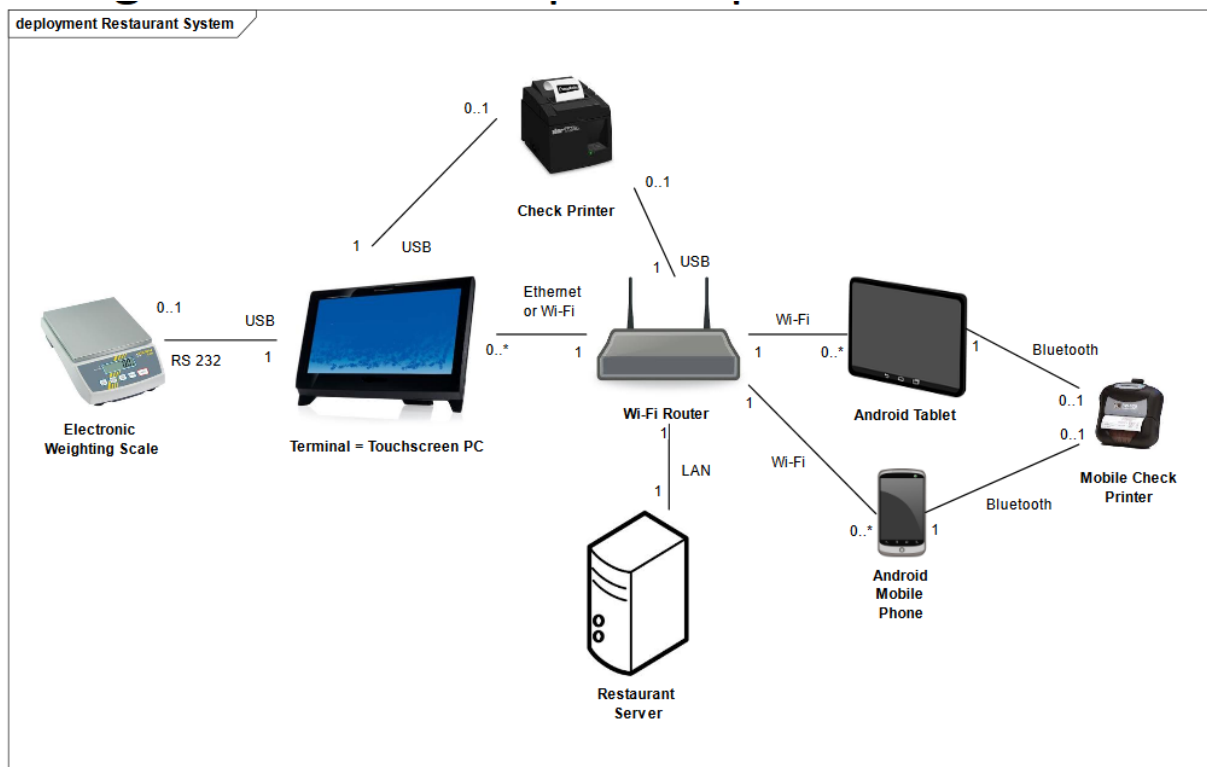
Diagram nasazení

- Popisuje:
 - o HW architekturu systému
 - o Nasazení SW a HW
- Základním elementem je uzel / Node
- Základní vazbou je asociace / Association

Syntaxe



Příklad diagramu nasazení HW



Správně dle UML, ale pro neznalce UML hůře pochopitelné.

Diagram nasazení SW na HW

- Grafický instalační manuál
- Umístění jednotlivých SW částí a popis jejich vztahů

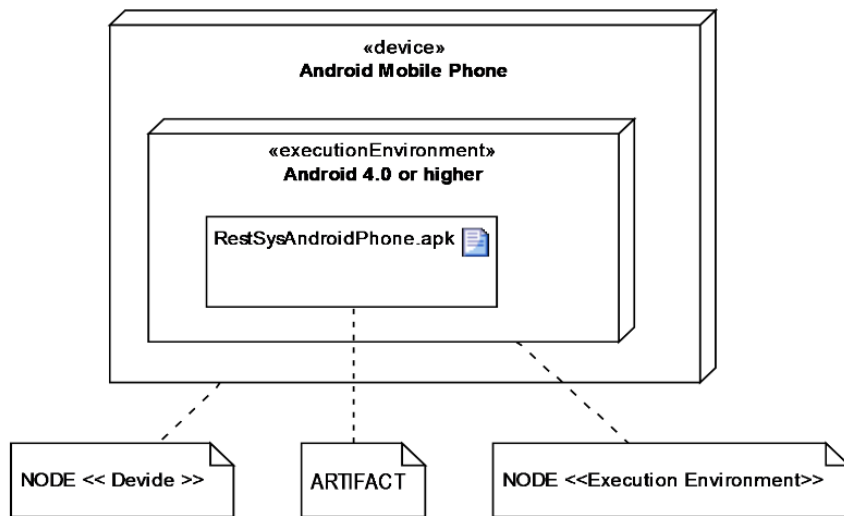
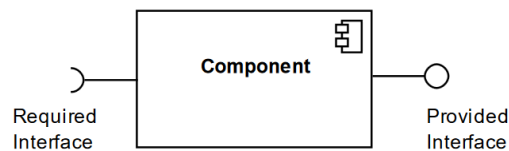


Diagram komponent

- popisuje logickou architekturu systému (tedy rozdělení systému na subsystémy/moduly)
- základním elementem je KOMPONENTA
- chování KOMPONENTY je definováno:
 - o POSKYTOVANÝMI ROZHRANÍMI
 - o POŽADOVANÝMI ROZHRANÍMI
- Komponenta:
 - o zastupuje část modulárního systému
 - o ukrývá (zapouzdřuje) svůj obsah a její chování může být nahrazeno jinou komponentou se stejným chováním
 - o KOMPONENTA je „černá skříňka“
 - o je speciálním případem TŘÍDY
 - o může obsahovat další komponenty
- DIAGRAM KOMPONENT tedy může popisovat logickou architekturu systému

Syntaxe



Příklad diagramu

