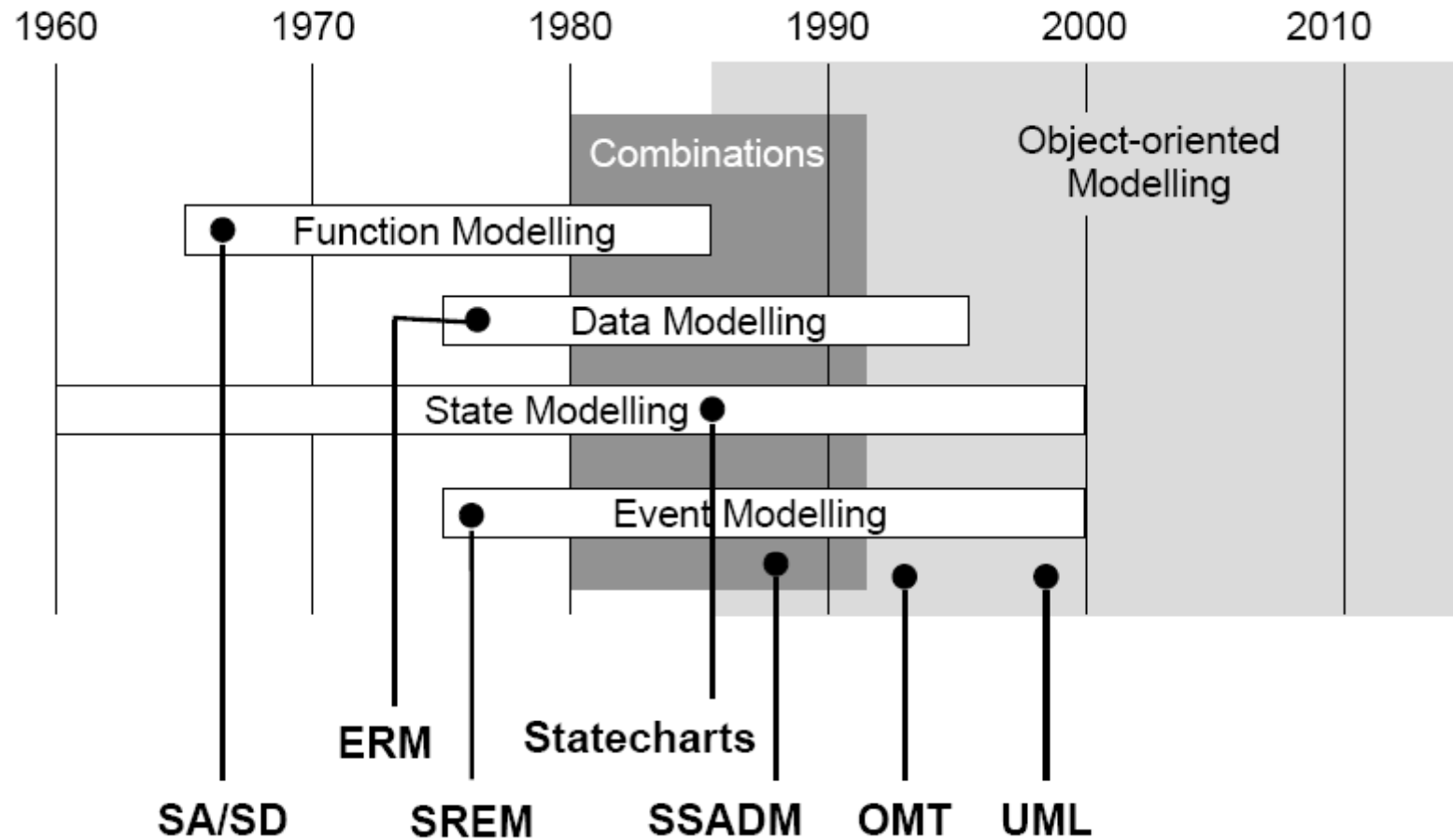


UML - Unified Modelling Language

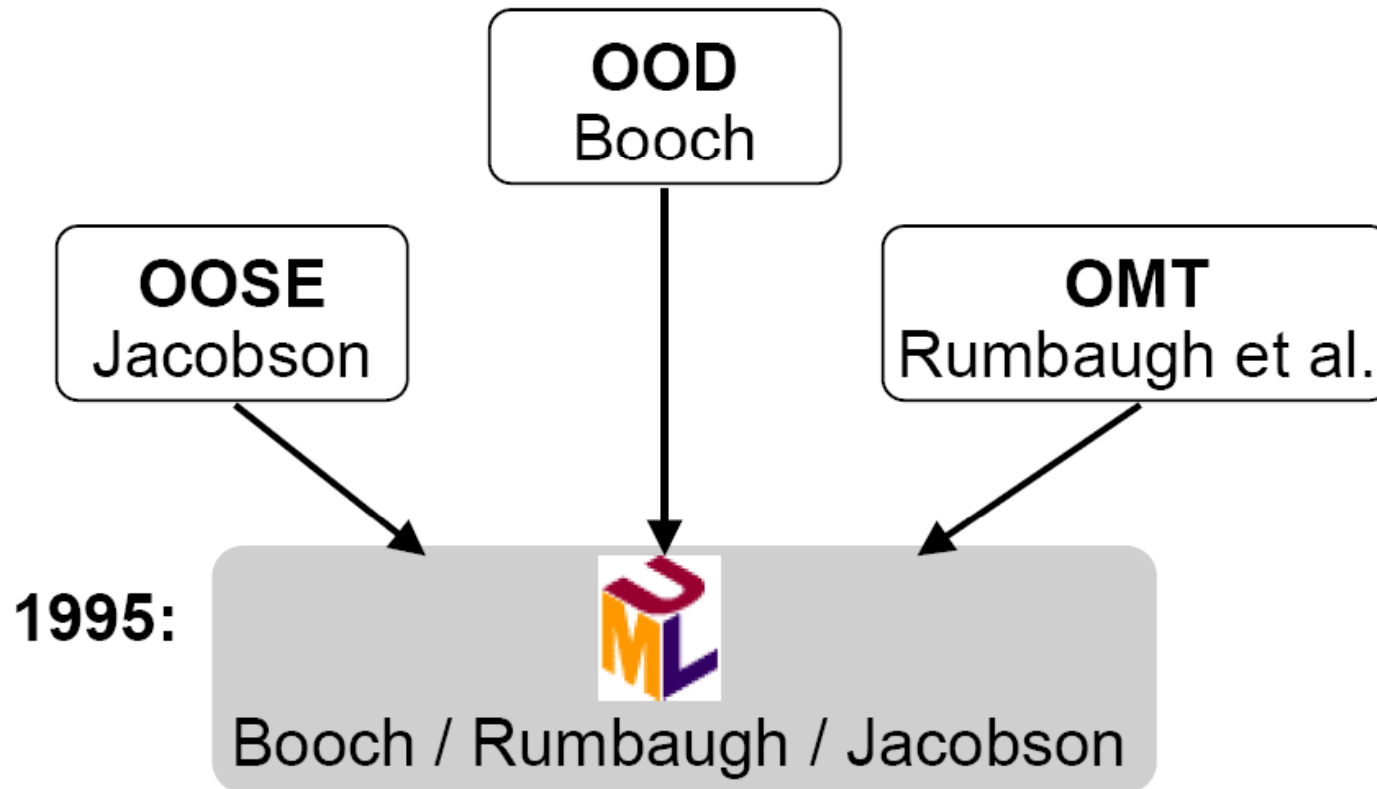
Historický vývoj



Co je UML

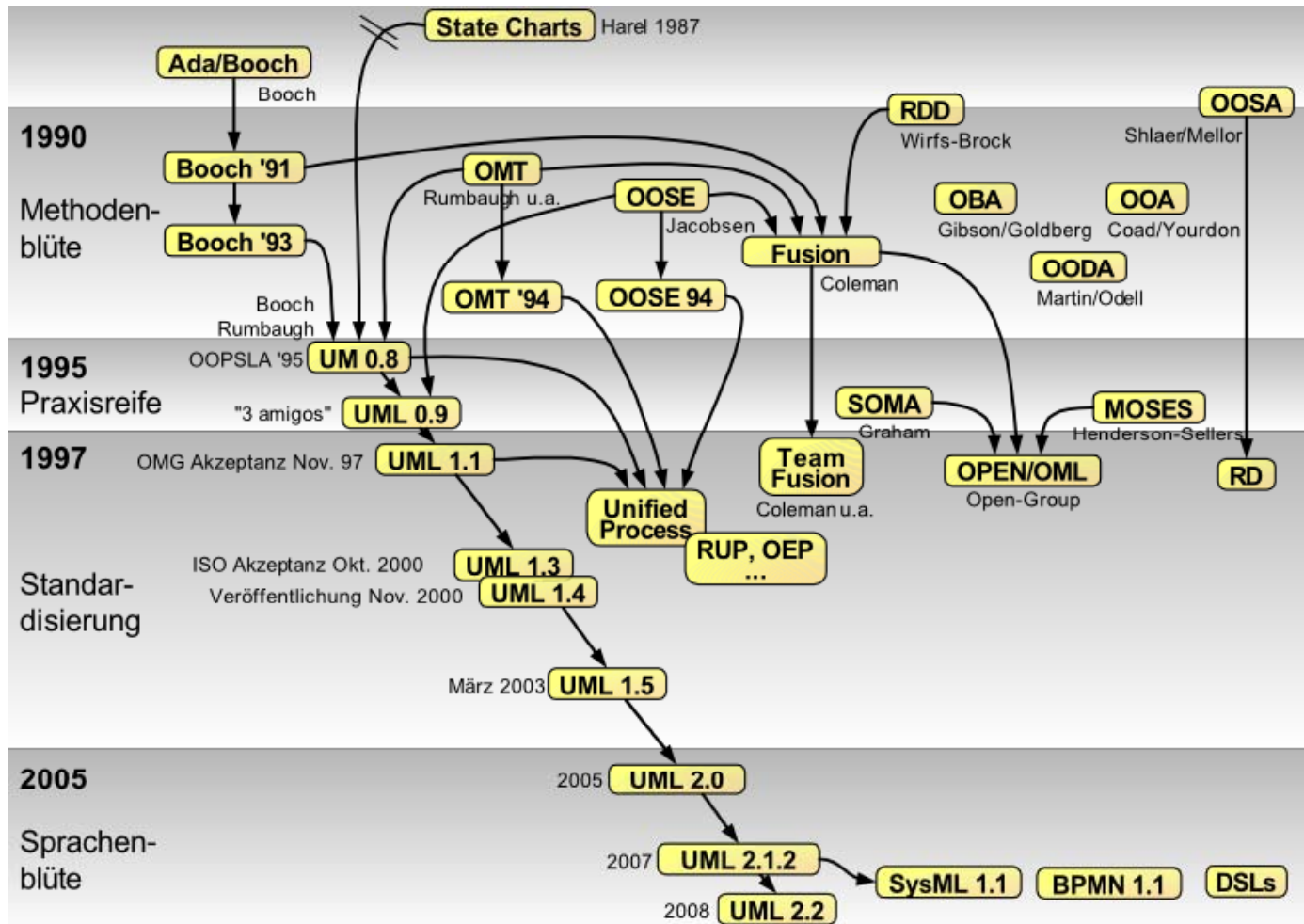


- UML - Unified Modelling Language
- UML kombinuje vše nejlepší z
 - konceptů datového modelování (entitně-relační diagramy)
 - modelování výrobních procesů (toky práce)
 - modelování objektů
 - modelování komponent
- UML je standardní jazyk pro zobrazení, specifikaci, konstrukci a dokumentaci artefaktů systémů s převážně softwarovou charakteristikou
- Může být použit při všech procesech životního cyklu vývoje a pro různé technologie implementace.





Historie UML





UML umožňuje (mimo jiné):

- zobrazit hranice systému & jeho hlavních funkcí pomocí případů užití (use cases) a účastníků (actors)
- ilustrovat realizaci případů užití pomocí diagramů interakcí
- reprezentovat statickou strukturu systému pomocí diagramů tříd
- modelovat chování objektů pomocí stavově-přechodových diagramů
- odhalit fyzickou implementační architekturu pomocí diagramů zapojení komponent
- rozšířit funkcionalitu pomocí stereotypů



Příklad: „Univerzita“ - zadání

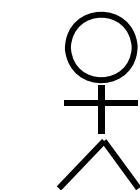
Univerzita chce registrační systém převést na počítače

- proděkan pro studijní plány stanoví kurikulum pro semestr. Jeden kurz může být nabídnut vícekrát jako „Nabídnutý kurz“.
- studenti volí 4 primární kurzy a 2 volitelné kurzy
- ihned poté, co se student zapíše do semestru, je účetní systém informován, že studentovi lze vystavit účet za semestr
- studenti mohou používat systém pro přidání/odhlášení kurzů po určitou dobu po zápisu.
- učitelé používají systém pro získání seznamu uchazečů zapsaných do jejich kurzů
- uživatelé registračního systému mají přidělené heslo, které je použito pro validaci při přihlášení do systému.



Příklad: „Univerzita“ - účastníci

Účastník (Actor) je někdo nebo něco, jež musí spolupracovat s (budoucím) vyvíjeným systémem



Proděkan



Profesor



Student



Účetní systém



Příklad: „Univerzita“ – případ užití

Případ užití je vzorem (předlohou) chování, které systém vykazuje

Každý případ užití je posloupnost souvisejících transakcí prováděných účastníkem a systémem během jejich dialogu.

Účastníky zkoumáme, abychom určili jejich potřeby

proděkan - údržba kurikula

profesor - žádá rozpis (seznam studentů,...)

student - údržba vlastního rozvrhu

účetní systém - získání podkladů pro účet po zápisu

Udržuj kurikulum

Žádej rozpis kurzu

Udržuj rozvrh



Dokumentace případů užití

Pro každý případ užití je vytvořen dokument obsahující tok událostí. Je napsán z pohledu účastníka.

Detaily, které systém musí poskytnout účastníkovi při provádění případu použití

Typické obsahy

- Začátek a konec případu použití
- Normální tok událostí
- Alternativní tok událostí
- Výjimečný tok událostí



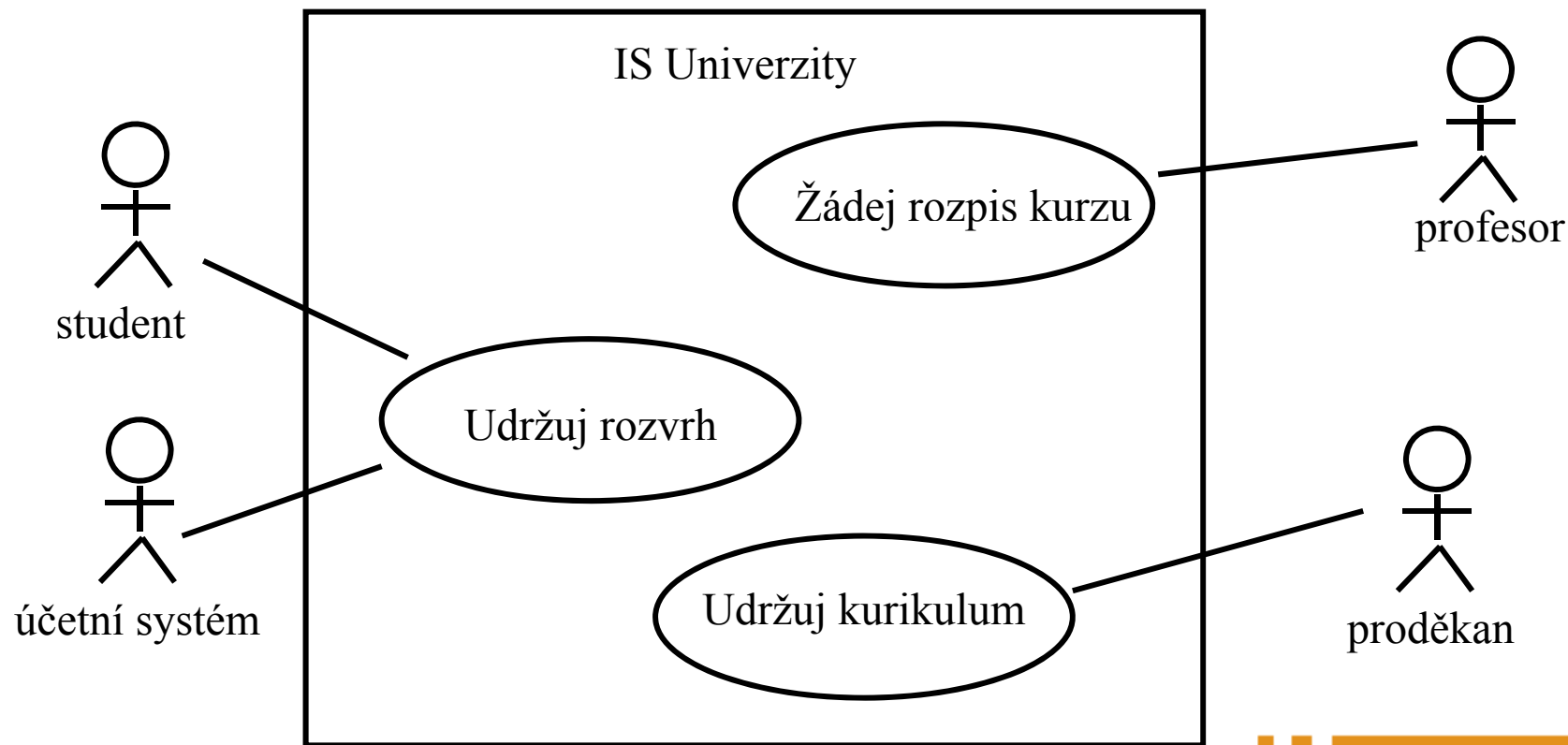
Příklad: „Univerzita“ – tok událostí při údržbě kurikula

- Tento případ užití začíná v okamžiku, kdy se proděkan přihlásí do Registračního systému a zadá heslo. Systém ověří platnost hesla (E1) a vyzve proděkana k výběru stávajícího nebo budoucího semestru (E2). Proděkan zadá požadovaný semestr. Systém vyzve k volbě požadované činnosti: ADD, DELETE, REVIEW, nebo QUIT.
- Pokud byla zvolena činnost ADD, pak se provede dílčí tok S-1: Přidej kurz.
- Pokud byla zvolena činnost DELETE, pak se provede dílčí tok S-2: Zruš kurz.
- Pokud byla zvolena činnost REVIEW, pak se provede dílčí tok S-3: Prohlédni kurikulum.
- Pokud byla zvolena činnost QUIT, případ užití končí.
- ...



Příklad: „Univerzita“ – diagram případu užití

Diagramy případů užití jsou tvořeny, aby znázornily vztahy mezi účastníky a případy užití.

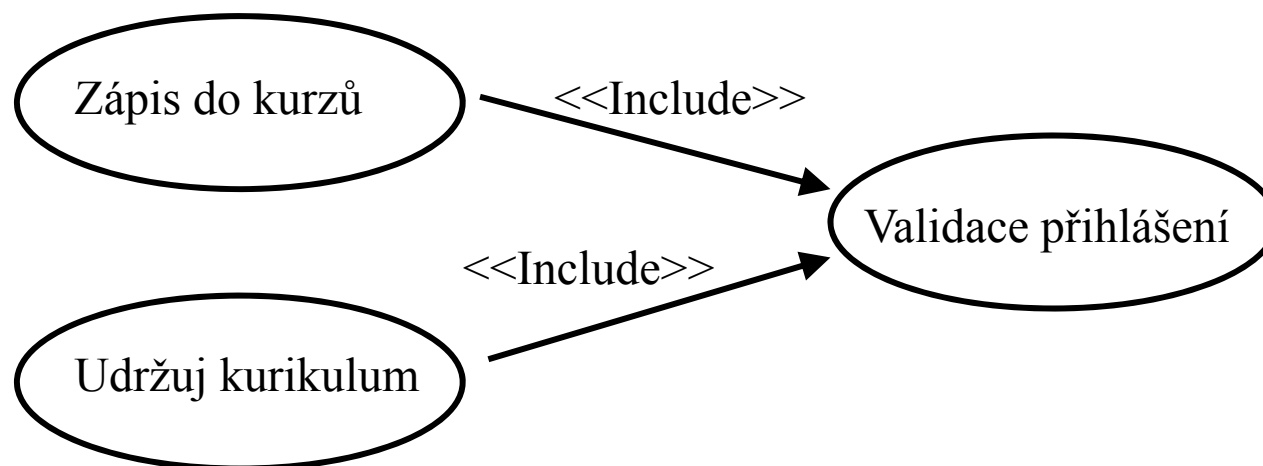




Příklad: „Univerzita“ – vztahy „Include“ a „Extend“

Při tvorbě dokumentace případů užití mohou být nalezeny další vztahy

- vztah „Include“ ukazuje chování, které je společné dvěma a více případům užití
- vztah „Extend“ ukazuje volitelné chování





- Diagram případů užití předkládá vnější pohled na systém.
- Interakční diagramy popisují, jak jsou případy užití realizovány jako interakce mezi skupinami objektů.
- Dva typy interakčních diagramů:
 - diagramy posloupností (sequence d.)
 - diagramy komunikace (communication d.)



Diagram posloupností

Diagram posloupností ukazuje interakci mezi objekty uspořádanou do časové posloupnosti.

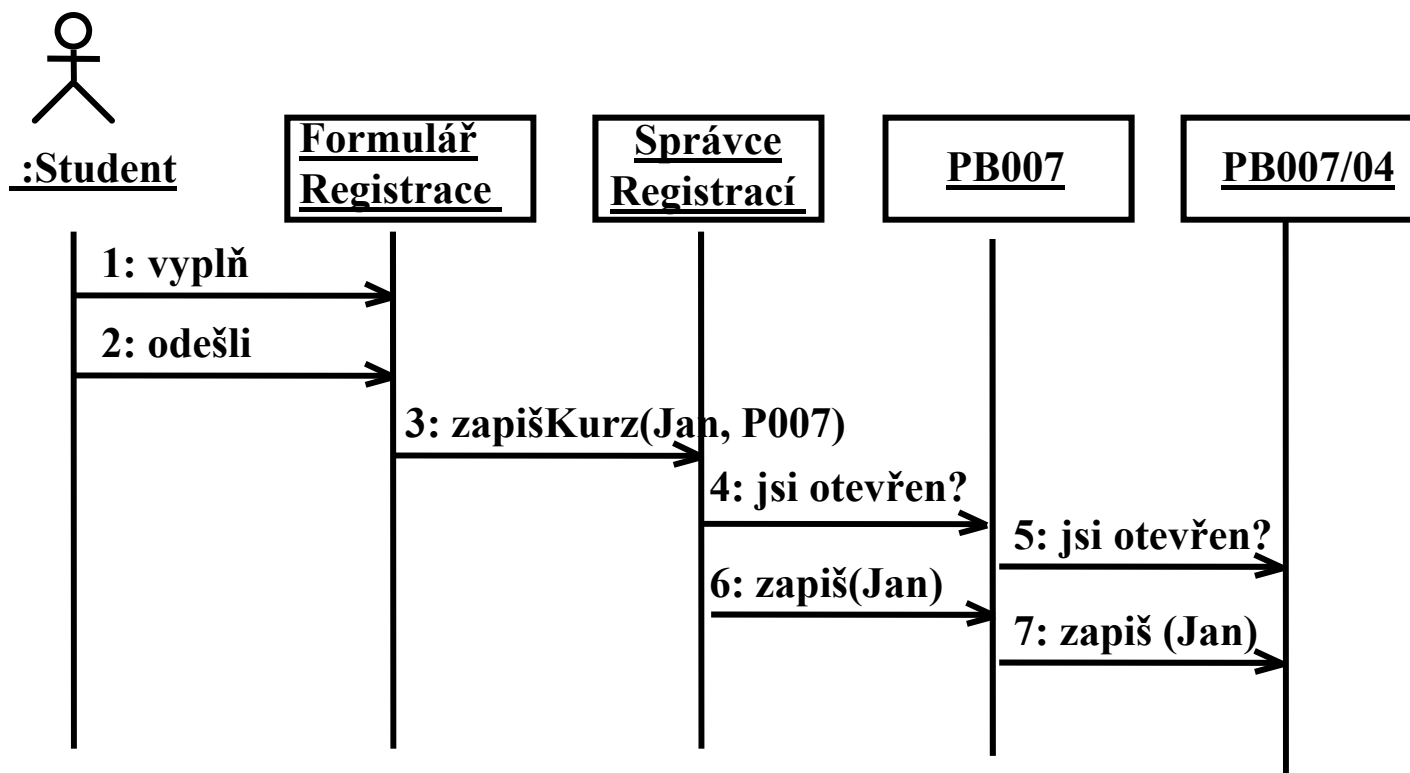
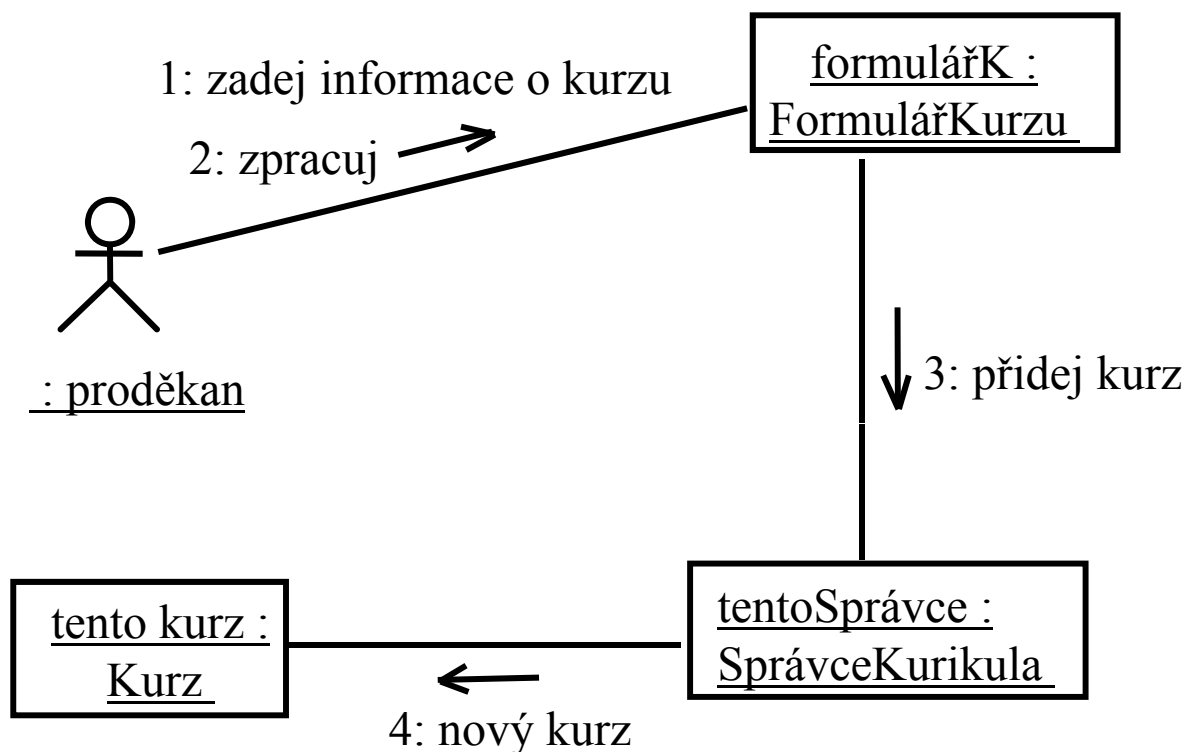




Diagram komunikace (spolupráce)

Diagram komunikace ukazuje interakce objektů a jejich propojení mezi sebou.



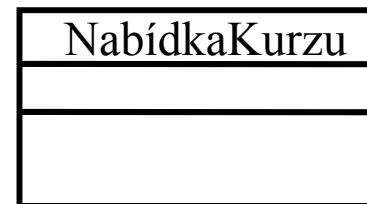
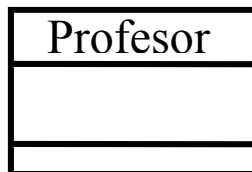
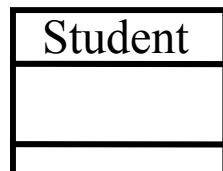
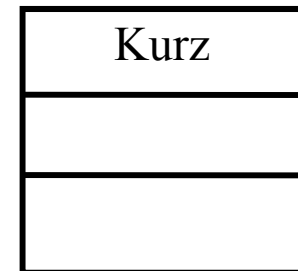


- Diagram tříd ukazuje existenci tříd a jejich vztahů v logickém pohledu na systém
- UML modelovací prvky používané v diagramech tříd:
 - třídy a jejich struktura a chování
 - asociace, agregace, závislost a vztahy dědění
 - příznaky (ukazatele) násobnosti a navigace
 - jména rolí



- Třída je kolekce objektů se shodnou strukturou, shodným chováním, shodnými vztahy a shodnou sémantikou.
- Třídy jsou nalezeny přezkoumáním objektů v diagramech posloupností a diagramech komunikace. Třída je zakreslena jako obdélník se třemi odděleními
- Třídy by měly být pojmenovány s použitím slovníku předmětné oblasti.
- Měly by být vytvořeny standardy pro volbu jmen.
 - např. všechny třídy pro jméno použijí podstatné jméno v jednotném čísle začínající velkým písmenem

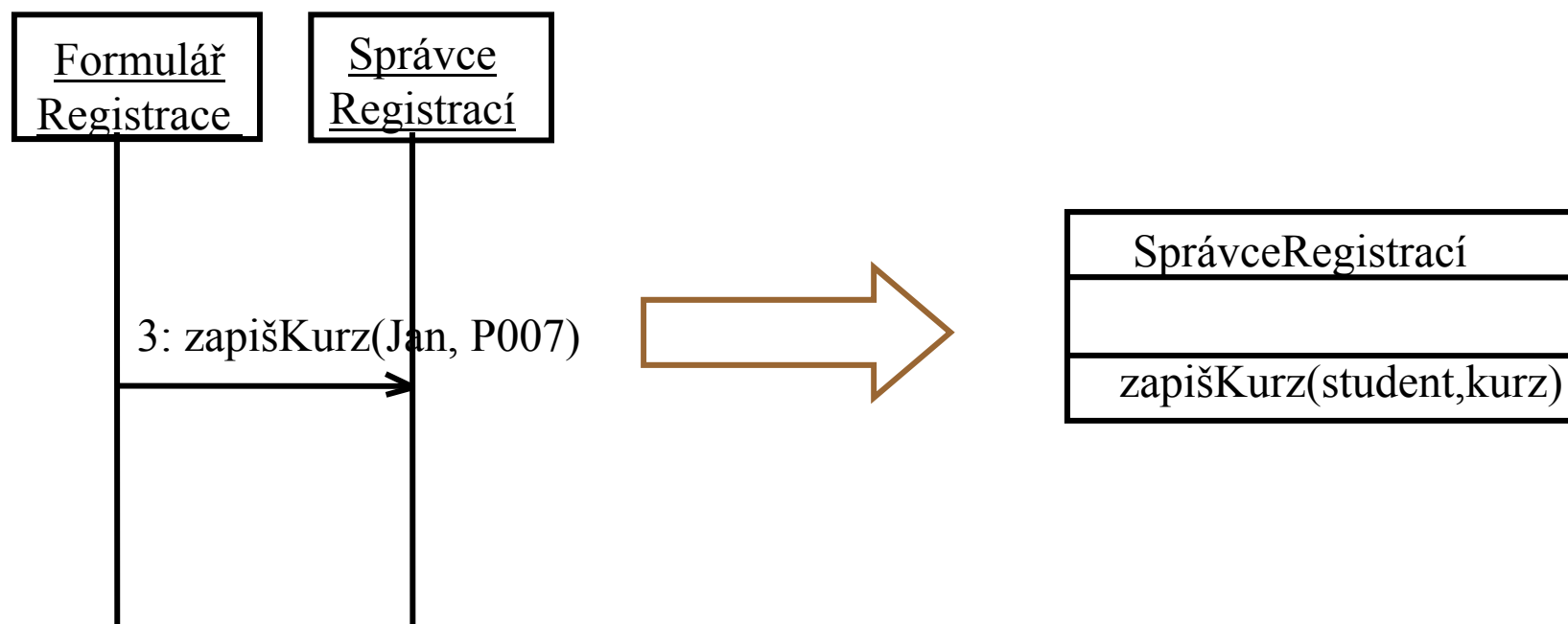
Třídy





Operace

- Chování třídy je reprezentováno jejími operacemi.
- Operace mohou být nalezeny po přezkoumání diagramů interakcí.



Atributy



- struktura třídy je reprezentována jejími atributy
- atributy mohou být nalezeny přezkoumáním definice třídy, požadavků na problémy a použitím znalostí z předmětné oblasti

Každá nabídka kurzu
obsahuje
číslo, místo a čas



NabídkaKurzu
číslo
místo
čas

Třídy



FormulářRegistrace

PlánovacíAlgoritmus

SprávceRegistrací
zapišKurz(student,kurz)

Kurz
název
početKreditů
otevři()
zapišStudenta(student)

Student
jméno
vedoucí

Profesor
jméno
zařazení

NabídkaKurzu
místo
otevři()
zapišStudenta(student)



- Vtahy poskytují cestu pro komunikaci mezi objekty
- Diagramy posloupností a/nebo komunikace zkoumáme, abychom určili, jaká spojení mezi objekty musí existovat pro splnění požadovaného chování - pokud dva objekty potřebují spolu „mluvit“, pak mezi nimi musí být vazba
- Tři typy vztahů jsou:
 - Asociace
 - Agregace
 - Závislost



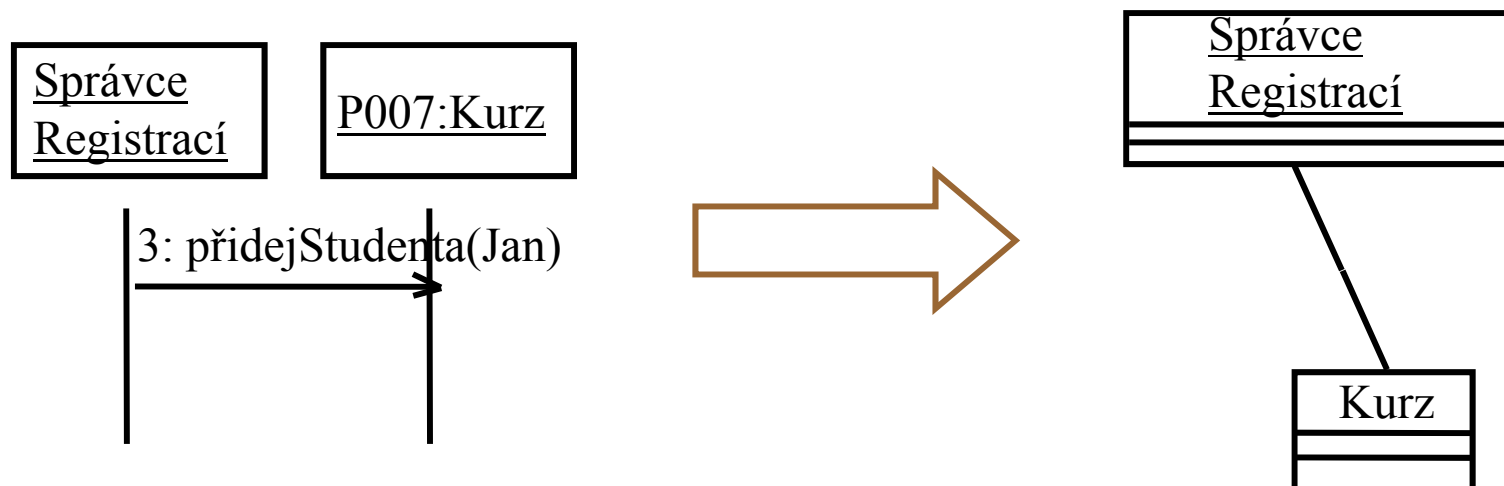
- Asociace je obousměrné propojení mezi třídami
 - Asociace se znázorňuje jako čára propojující vztažné třídy
- Agregace je silnější forma vztahu, jedná se o vztah mezi celkem a jeho částmi
 - Agregace se znázorňuje jako čára propojující vztažné třídy, značka diamant je umístěna u třídy, která představuje celek
- Vztah závislosti je slabší formou vztahu mezi klientem a poskytovatelem, kde klient nemá žádnou sémantickou znalost o poskytovateli
 - Závislost je ukázána jako čárkovaná čára se šipkou ukazující od klienta k poskytovateli



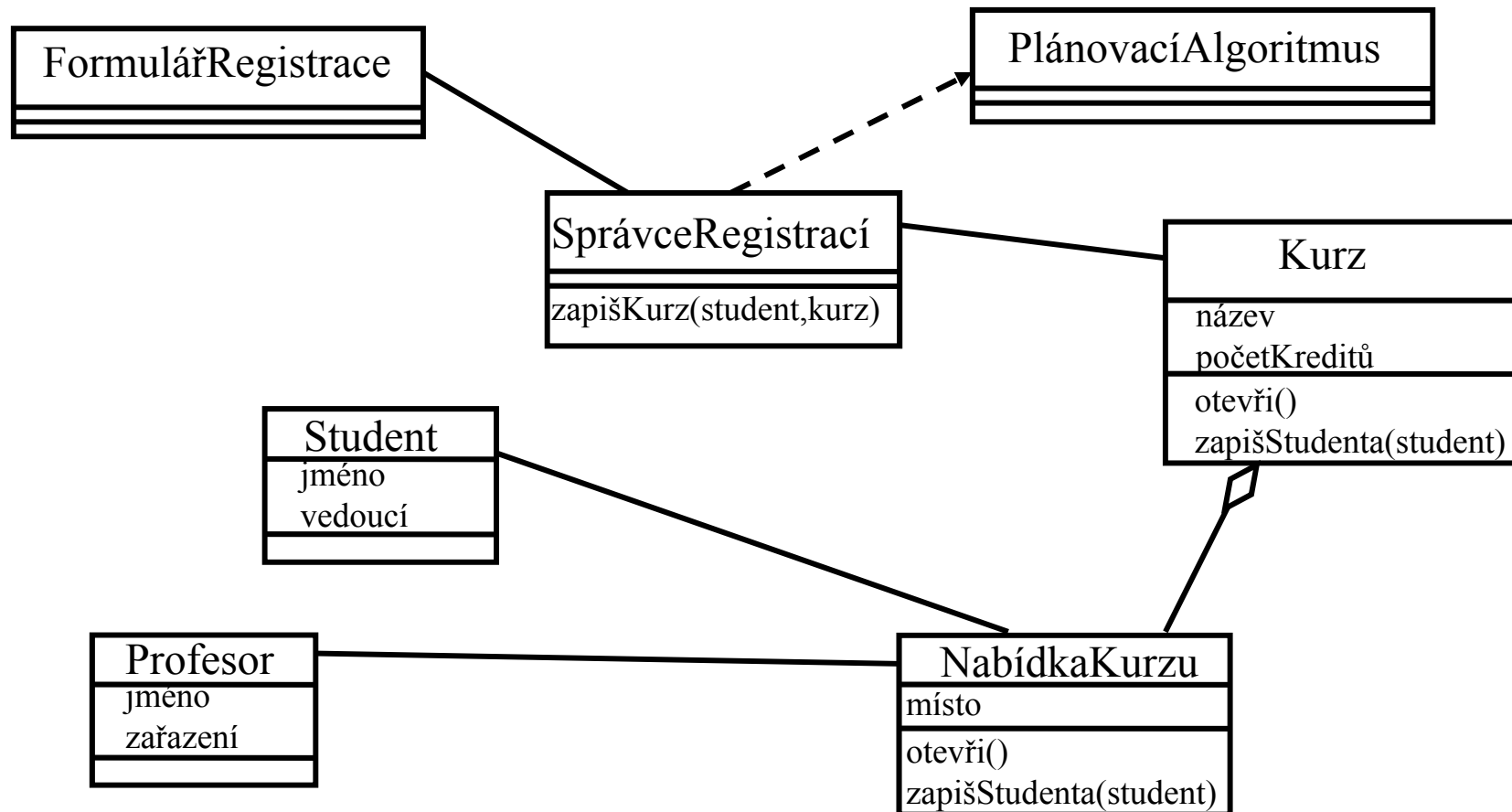
Nalezení vztahů

Vztahy jsou odhaleny po přezkoumání diagramů interakcí.

- Pokud dva objekty musí „hovořit“, pak musí existovat komunikační cesta.



Vztahy





Násobnost definuje, kolik objektů se účastní vztahů.

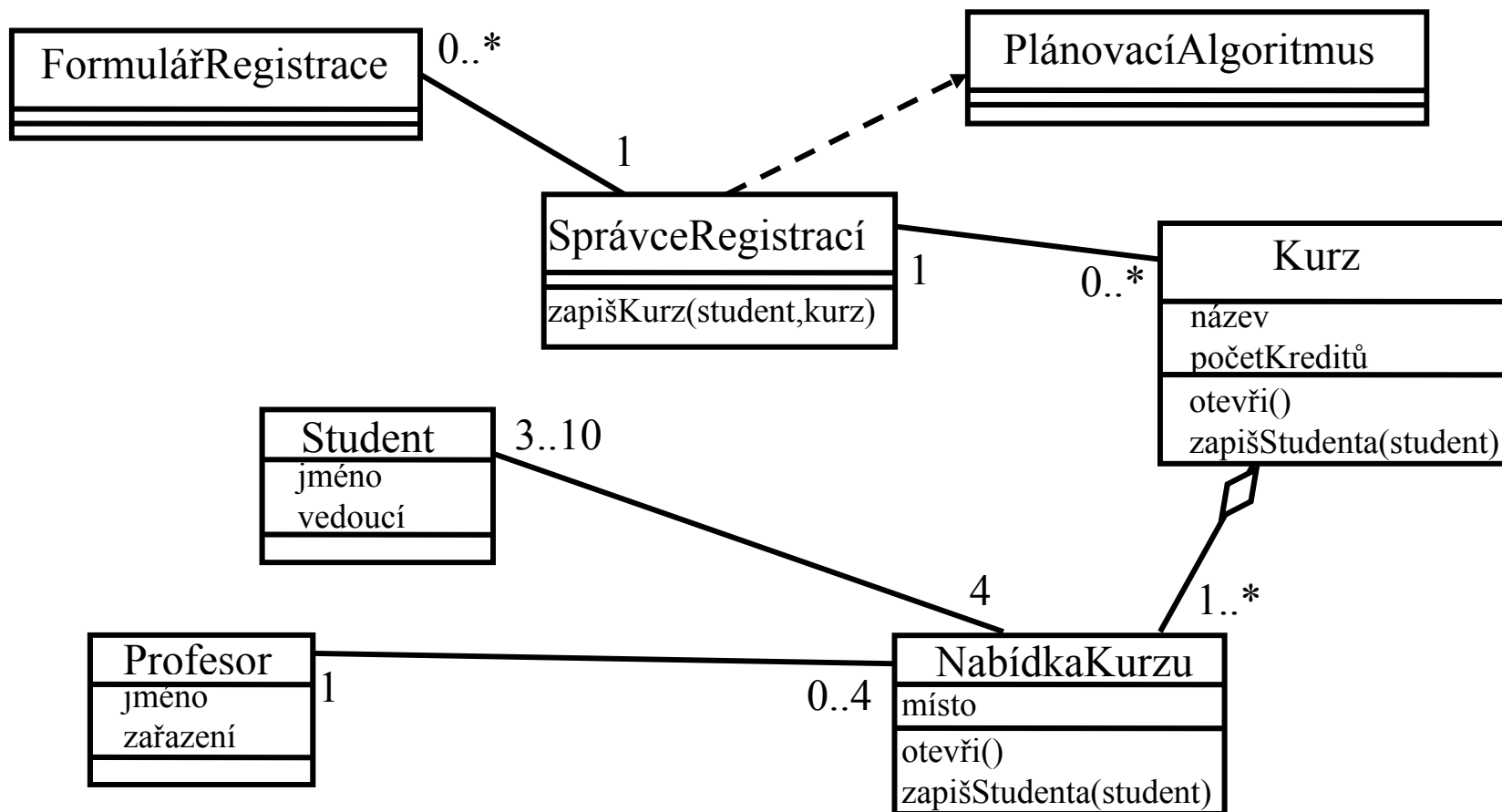
- Násobnost je počet instancí jedné třídy vztažená k JEDNÉ instanci druhé třídy.
- Pro každou asociaci a agregaci musí nalézt dvě násobnosti: pro každý konec vztahu.

Ačkoliv asociace a agregace jsou implicitně obousměrné, často je vhodné omezit navigaci na jeden směr.

V případě omezené navigace přidaná šipka určuje směr navigace.



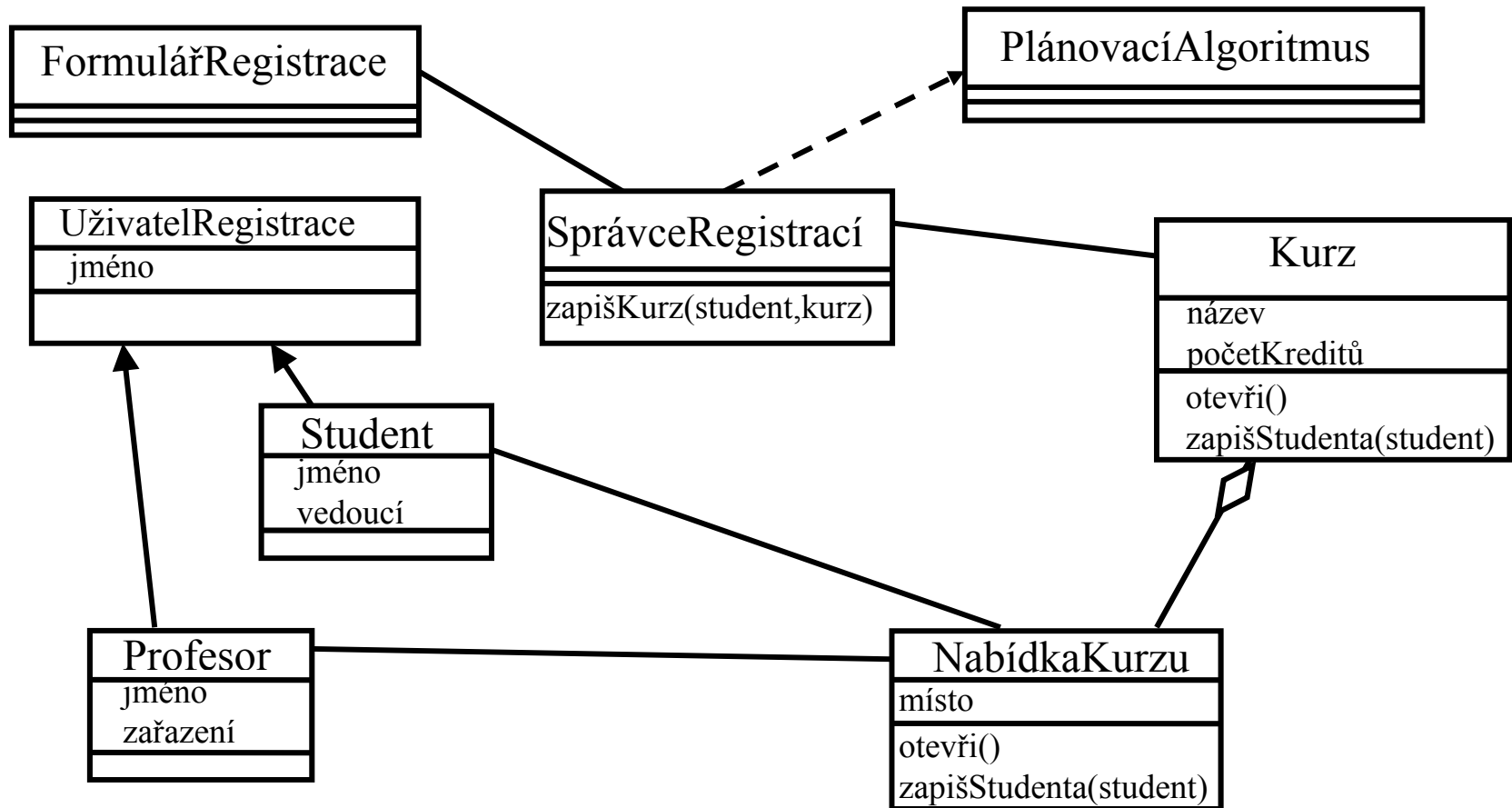
Násobnost a navigace





- Dědičnost je vztah mezi nadtrídou a jejími podtrídami.
- Existují dvě cesty, jak nalézt dědičnost:
 - zobecnění
 - specializace
- Společné atributy, operace a/nebo vztahy jsou ukázány na nejvyšší aplikovatelné úrovni hierarchie.

Dědičnost

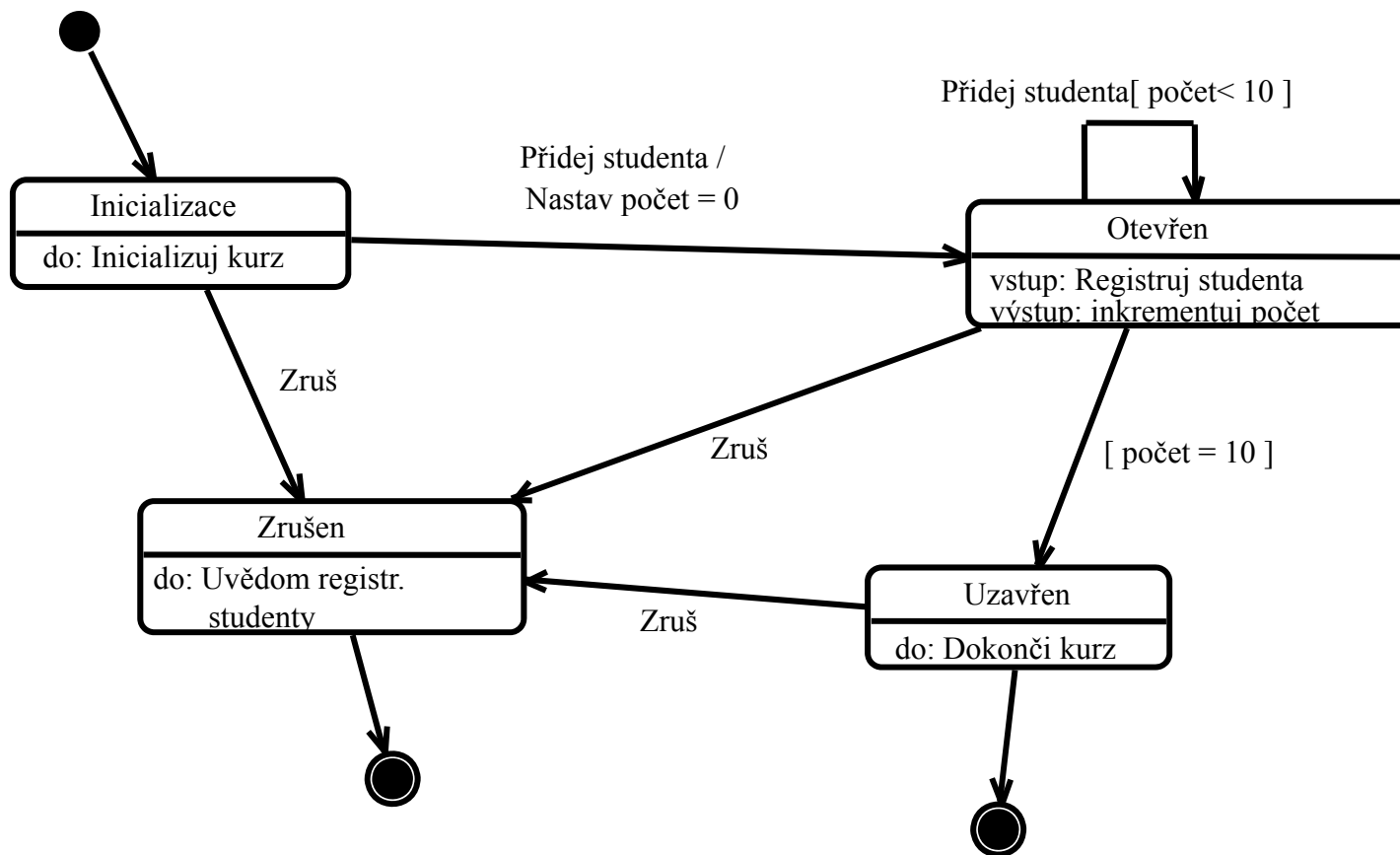




- Stavově-přechodový diagram ukazuje
 - životní historii dané třídy
 - události, které způsobují přechod z jednoho stavu do jiného
 - akce, které jsou výsledkem změny stavu
- Stavově-přechodové diagramy se tvoří pro objekty, které mají významné dynamické chování



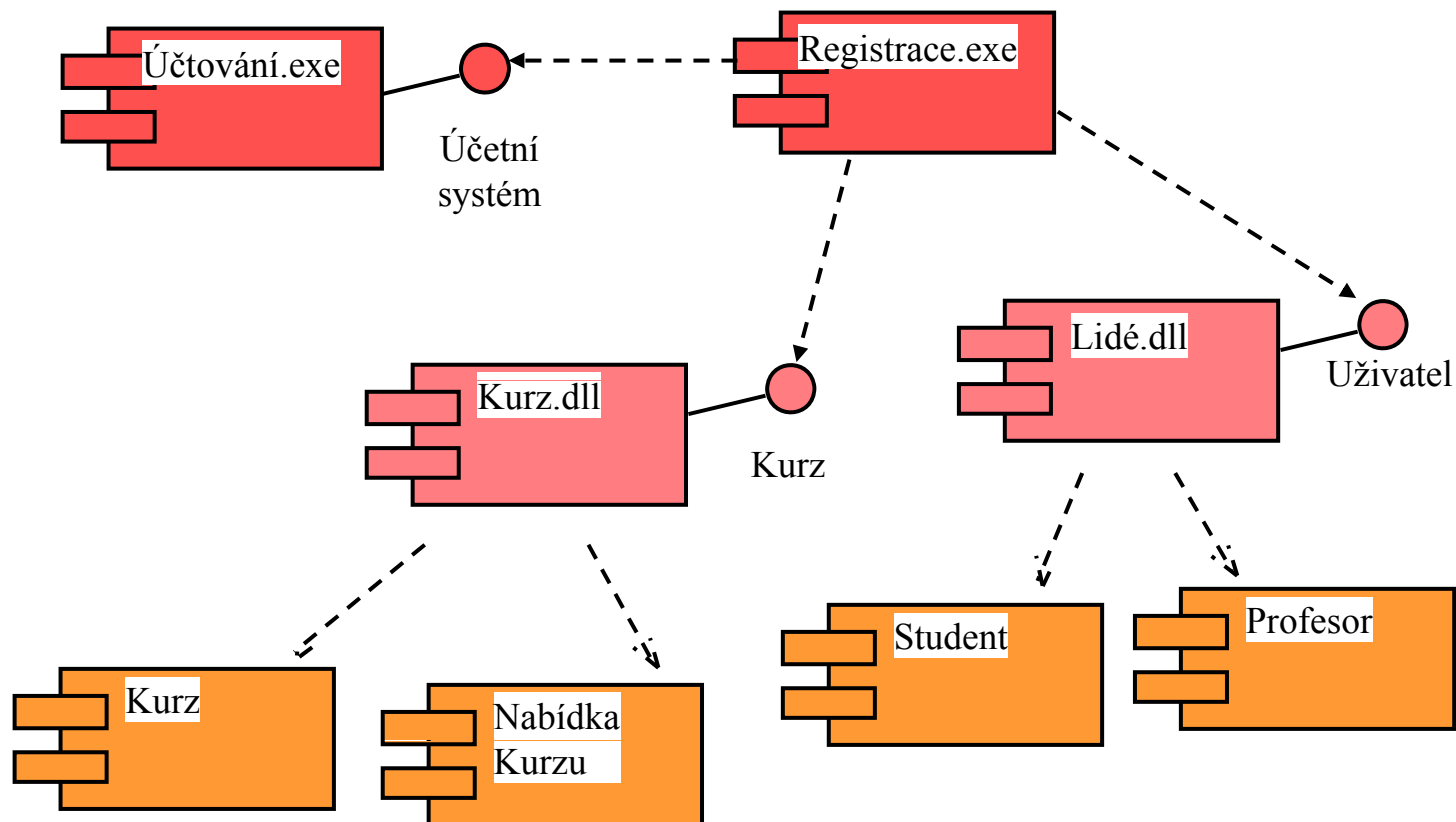
Stavově-přechodový diagram





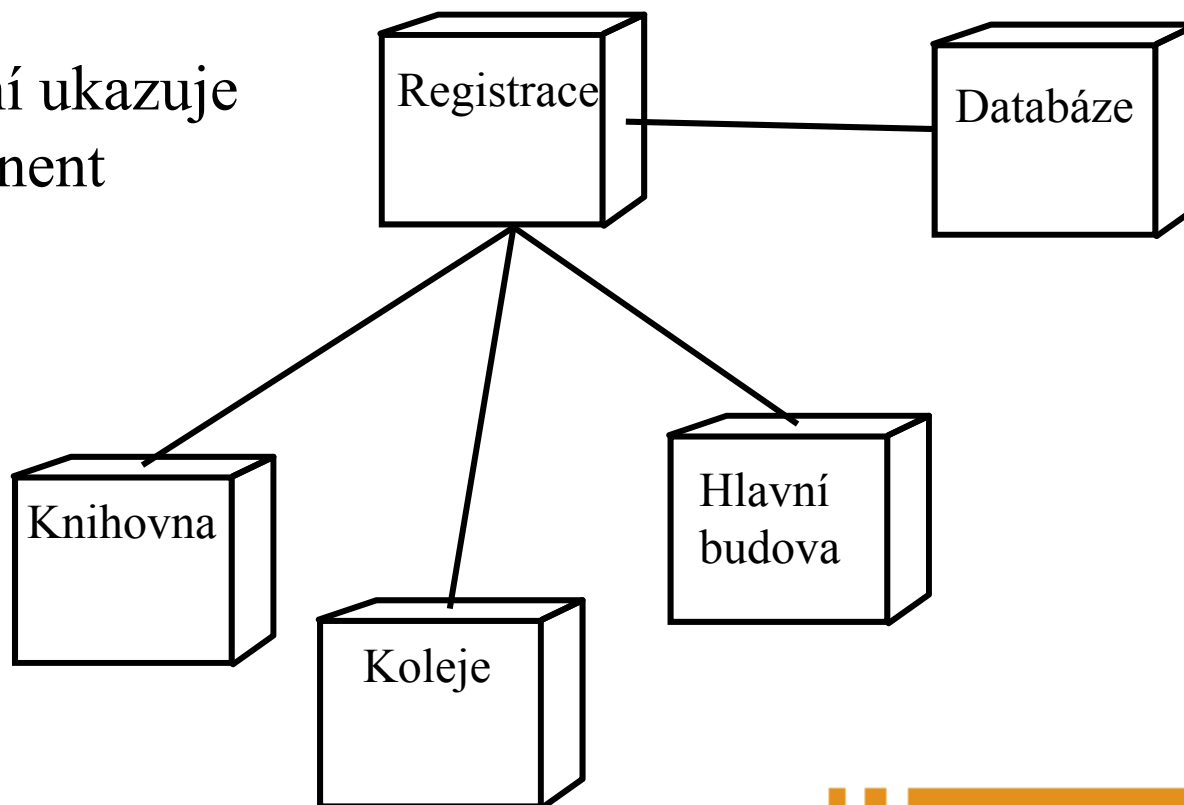
- Diagramy komponent znázorňují uspořádání a závislosti mezi softwarovými komponentami
- Komponenta může být
 - komponenta se zdrojovým kódem
 - komponenta běhu programu (pracovní soubor, ...)
 - proveditelná komponenta (dll, exe)

Diagram komponent

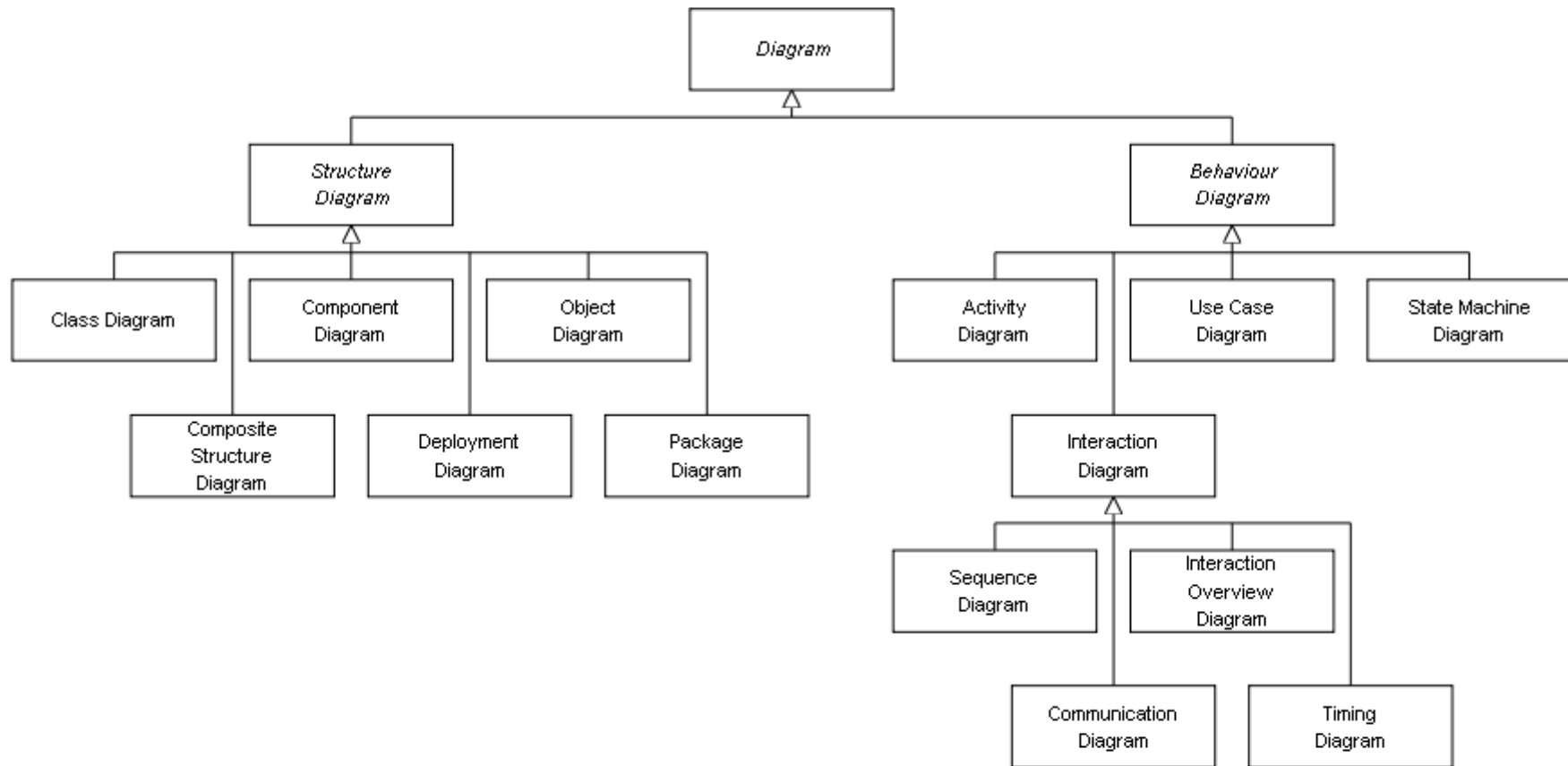




- Diagram nasazení ukazuje konfiguraci procesních prvků použitých při běhu a softwarové procesy, které jsou na ně umístěny
- Diagram nasazení ukazuje rozložení komponent v podniku.



Modely zahrnuté v UML





- do UML doplnit další prvky notace pomocí stereotypů
- stereotypy mohou být použity pro klasifikaci a rozšíření asociací, vztahů dědičnosti, tříd a komponent
- Příklady:
 - stereotypy tříd: hranice, řízení, entita, utilita, výjimka
 - stereotypy dědičnosti: užívá a rozšiřuje
 - stereotypy komponent: subsystém