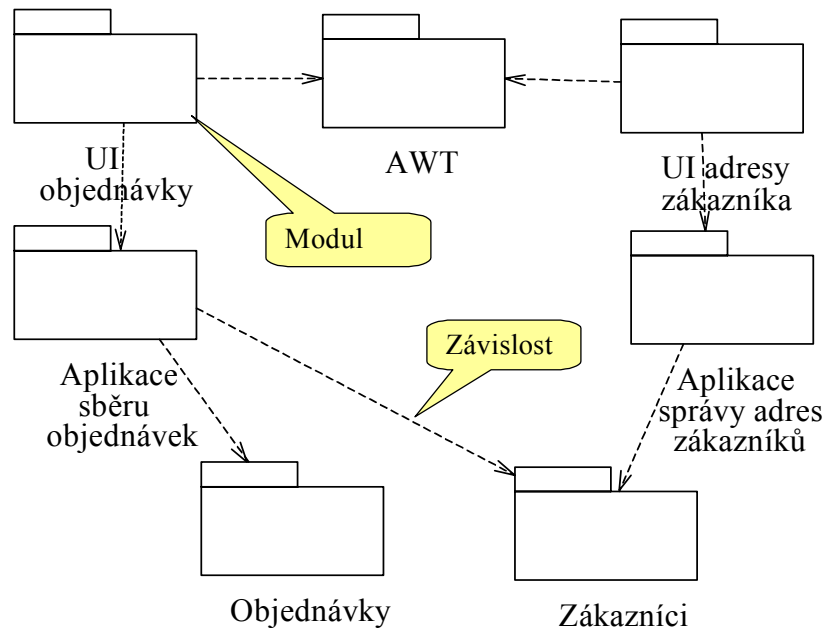


## 7.6 Další diagramy UML

### 7.6.1 Moduly (balíčky - package) a kolaborace (collaboration)

Jak rozložit rozsáhlý systém na menší?

- seskupování tříd (prvků modelu) do jednotek vyšší úrovně (package v UML).

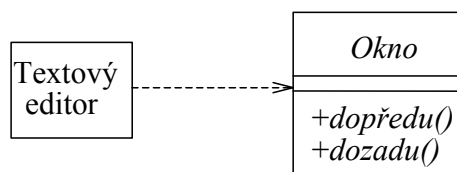


- obsah: - textově, graficky, další diagram

#### • Modelování závislostí v UML

**Závislost** – vztah mezi prvky modelu, kdy specifikace jednoho prvku může ovlivnit jiný prvek, který jej používá.

Př)



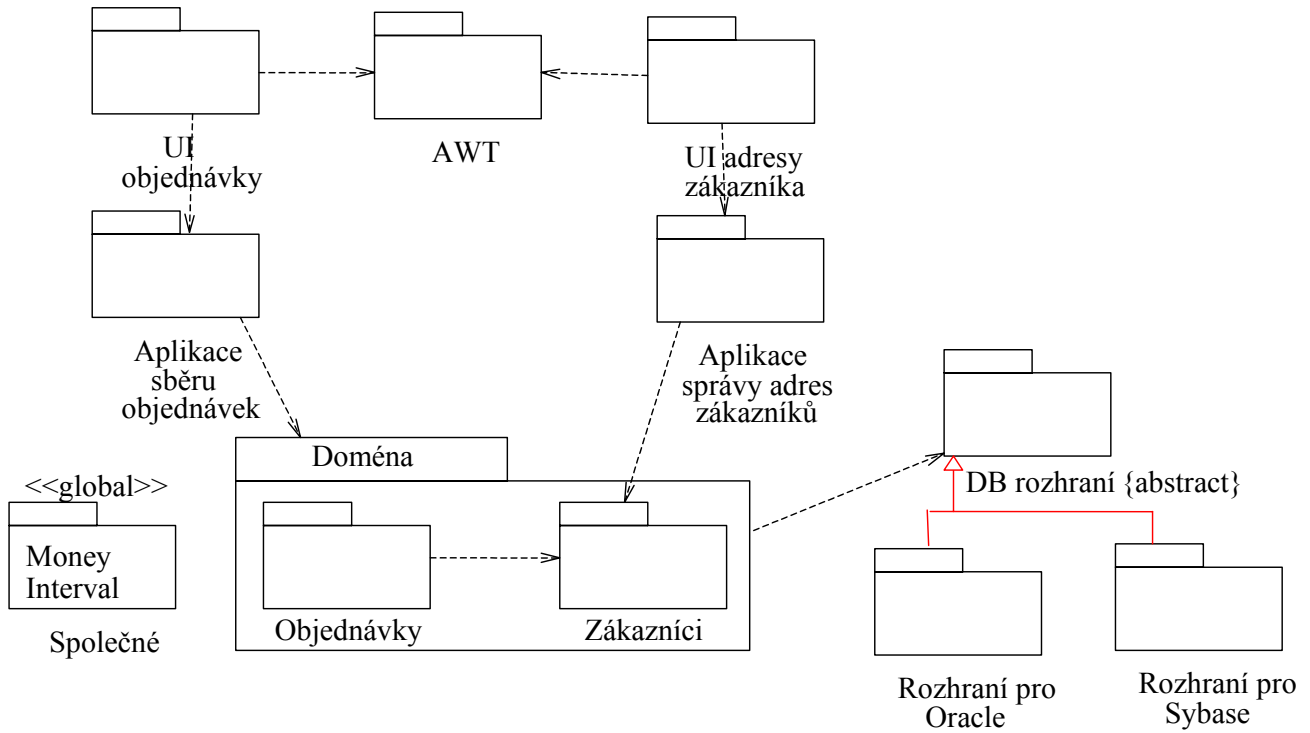
- řada různých závislostí s předdefinovanými stereotypy (include, extend, instantiate, instanceof, use, friend, ...)

- dvě seskupení jsou závislá, existuje-li závislost mezi prvky v nich

- obdoba kompilačních závislostí (nejsou ale tranzitivní)

- redukce závislostí v systému – cíl návrhu

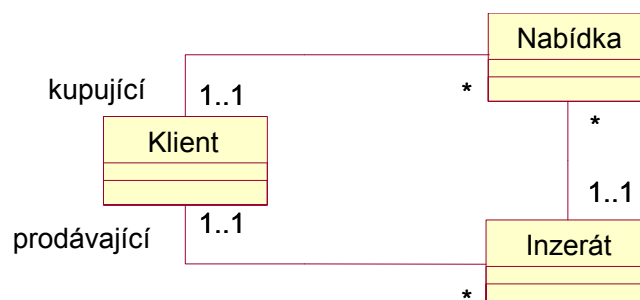
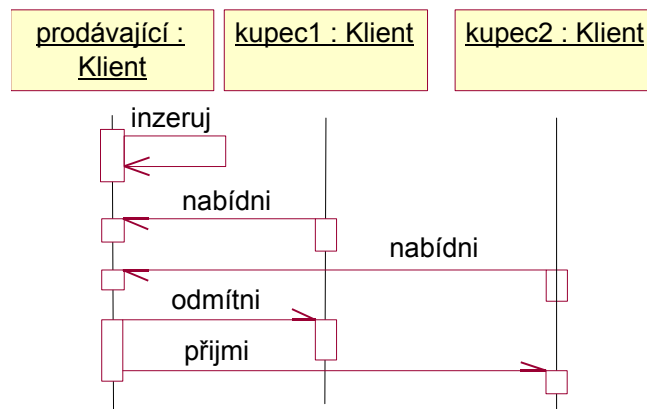
Př)



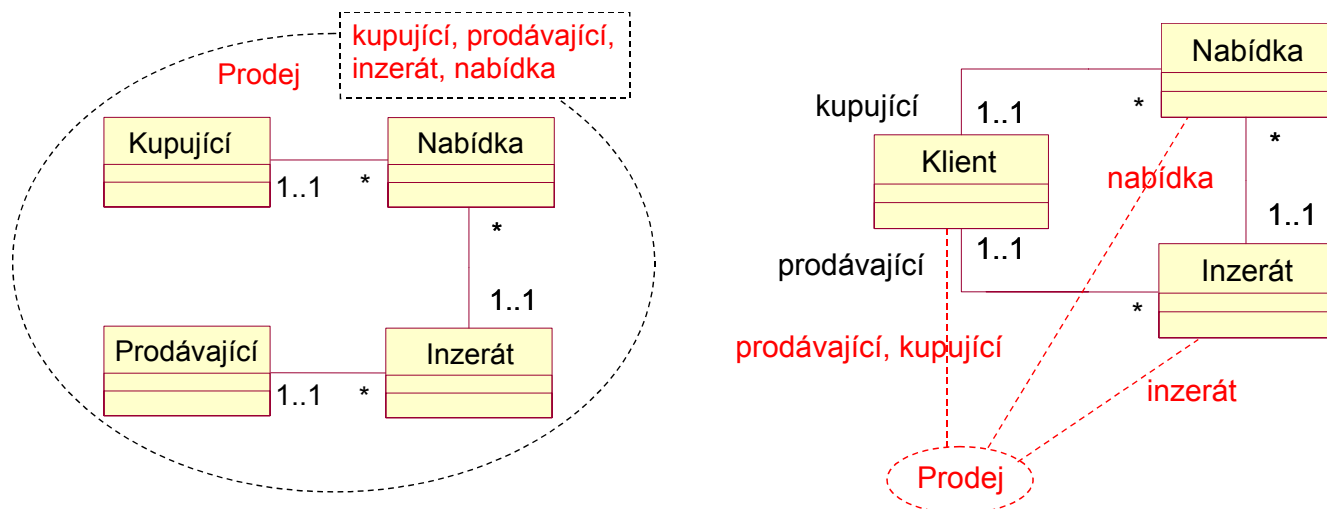
## • Kolaborace

- seskupení spolupracujících tříd

**Kolaborace** – jméno přiřazené interakci mezi dvěma a více třídami.



- kolaborace může ukazovat implementaci operace nebo realizaci případu použití
- parametrizovaná kolaborace (vzor – pattern)

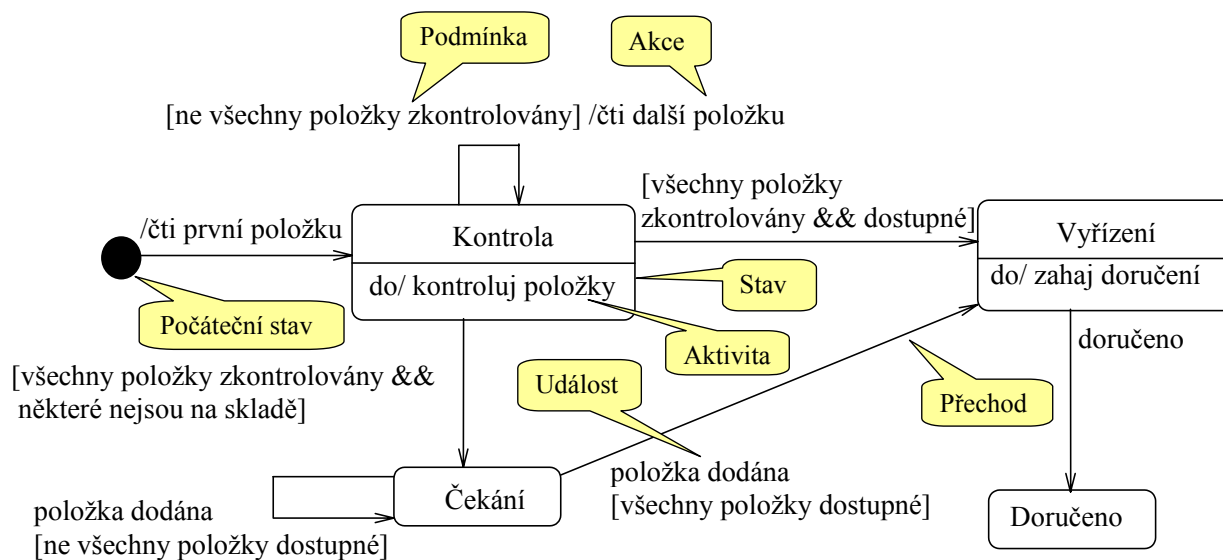


- Kdy použít seskupení a kolaborace
  - rozsáhlé projekty, vhodná jednotka testování

## 7.6.2 Stavové diagramy

**Stavový diagram** popisuje **stavy**, v nichž se může objekt nacházet, a jak se stav objektu **změní** jako následek **události**, která na objekt působí. Kreslí se pro jednu třídu a ukazuje chování objektu třídy v průběhu jeho životního cyklu.

Př)



- **Stav objektu**

- situace během života objektu, během níž objekt splňuje nějakou podmínku, provádí nějakou operaci nebo čeká na událost

Př) Student: přihlášený, přijatý, zapsaný, student 1.stupně, ...

- stavu může být přiřazena *aktivita* a *vstupní, výstupní a interní akce* (do/..., entry/..., exit/..., událost/...)

- počáteční a koncový stav (pseudostavy)

**Aktivita** – proces, který trvá „nějakou dobu“.

**Akce** – proces, který proběhne „rychle“ a je nepřerušitelný.

- **Událost**

**Událost** - něco, co se stane v určitém časovém okamžiku, nemá trvání.

**Syntaxe: jméno\_události (seznam parametrů)**

Př) vysokáTeplota (požadovaná teplota)

- událost po uplynutí času: **after** (např. after (30 minut))

- událost po splnění podmínky: **when** (např. when (tlak >50) kP))

- událost ve stavu: **entry, exit**.

- **Přechod mezi stavy**

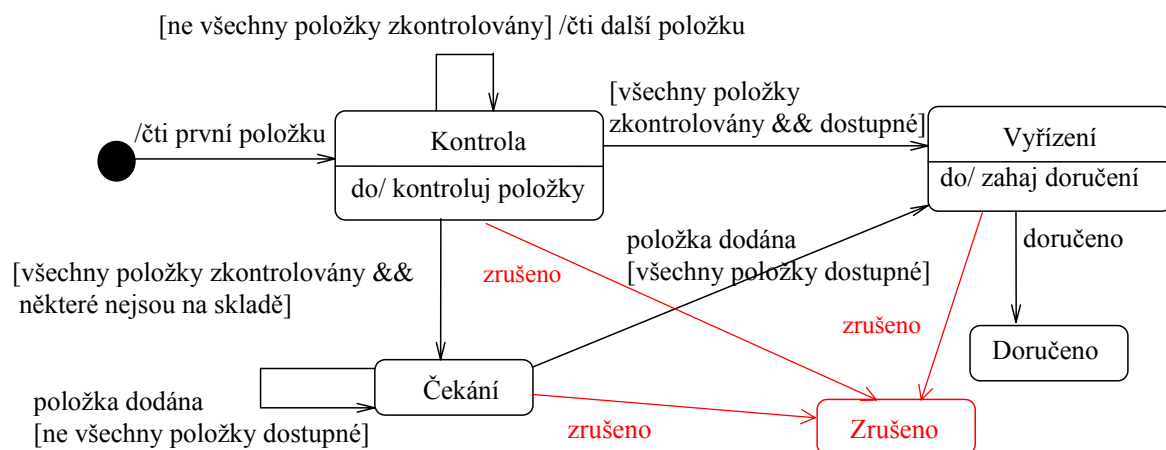
**Syntaxe ohodnocení přechodu: Událost [Podmínka]/ Akce**

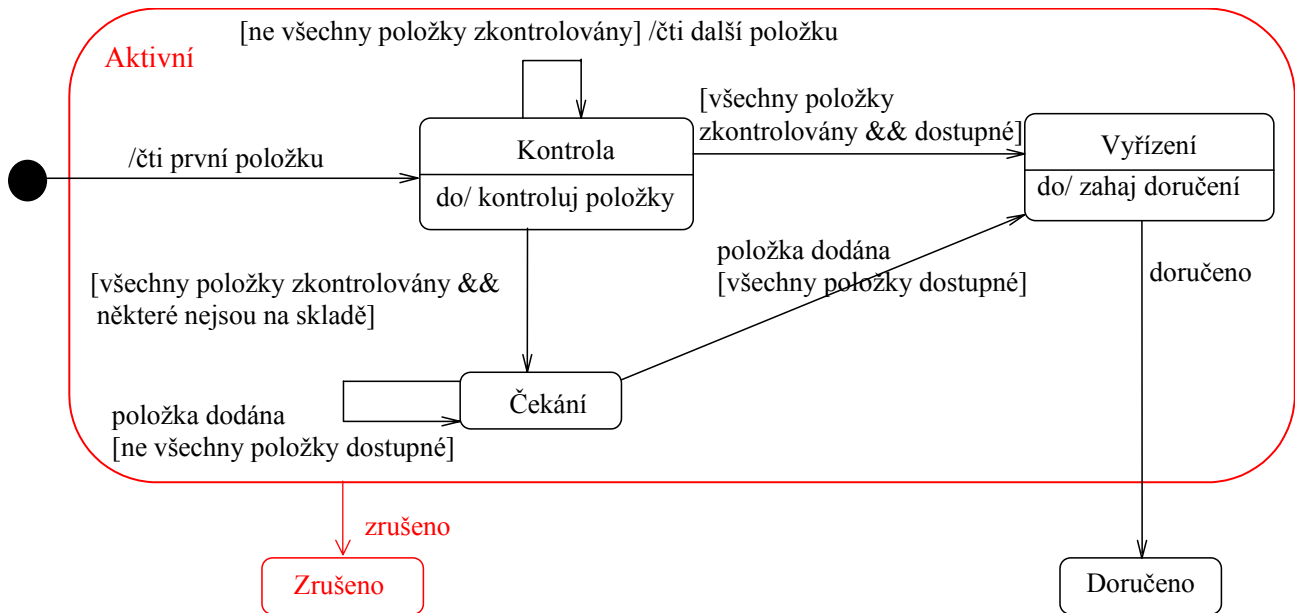
**Podmínka (guard)** – booleovský výraz platný podmiňující přechod.

- pravidla pro aktivaci přechodu

- jen jeden přechod ze stavu může být aktivován → není-li událost musí být podmínky výstupních přechodů vzájemně vylučné

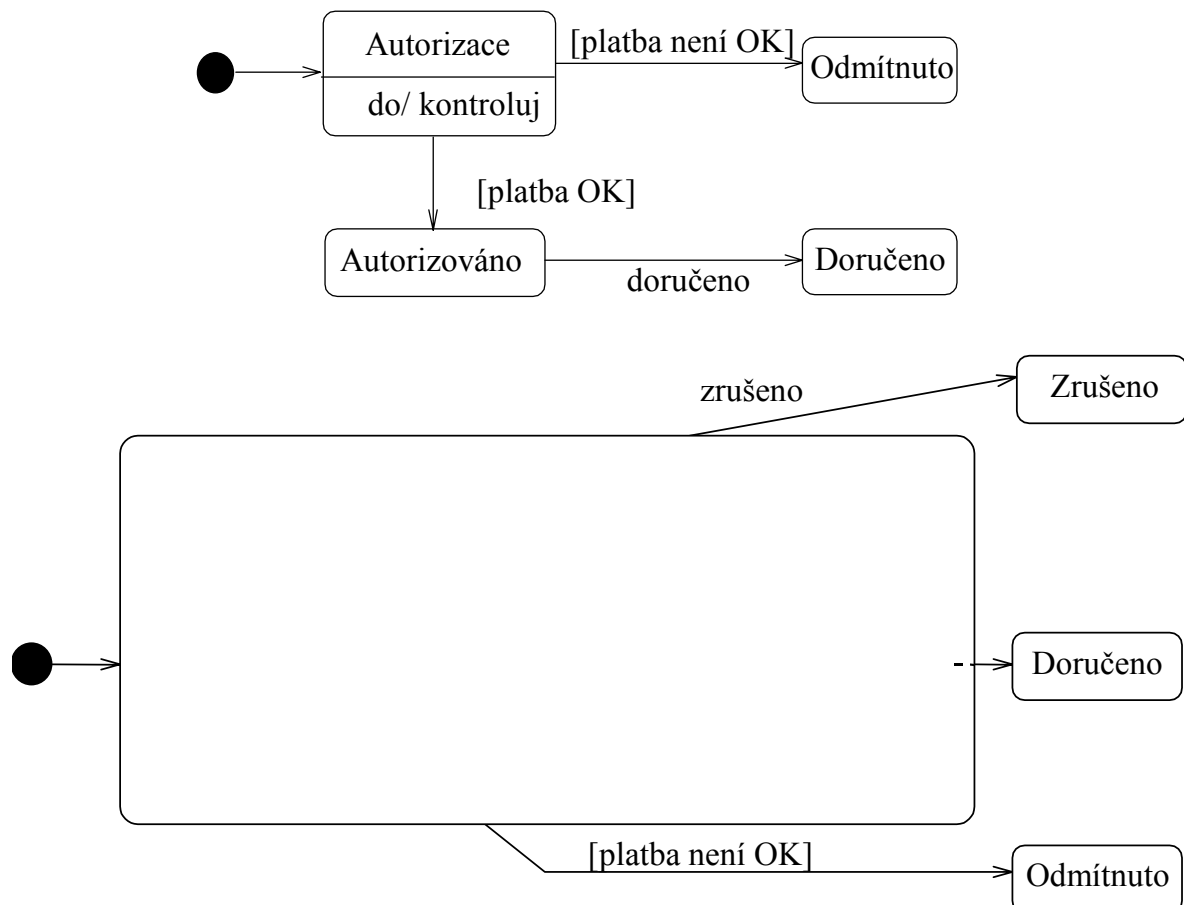
Př) Přechod pro zrušení objednávky událostí „Zrušena“





• **Souběžné stavové diagramy**

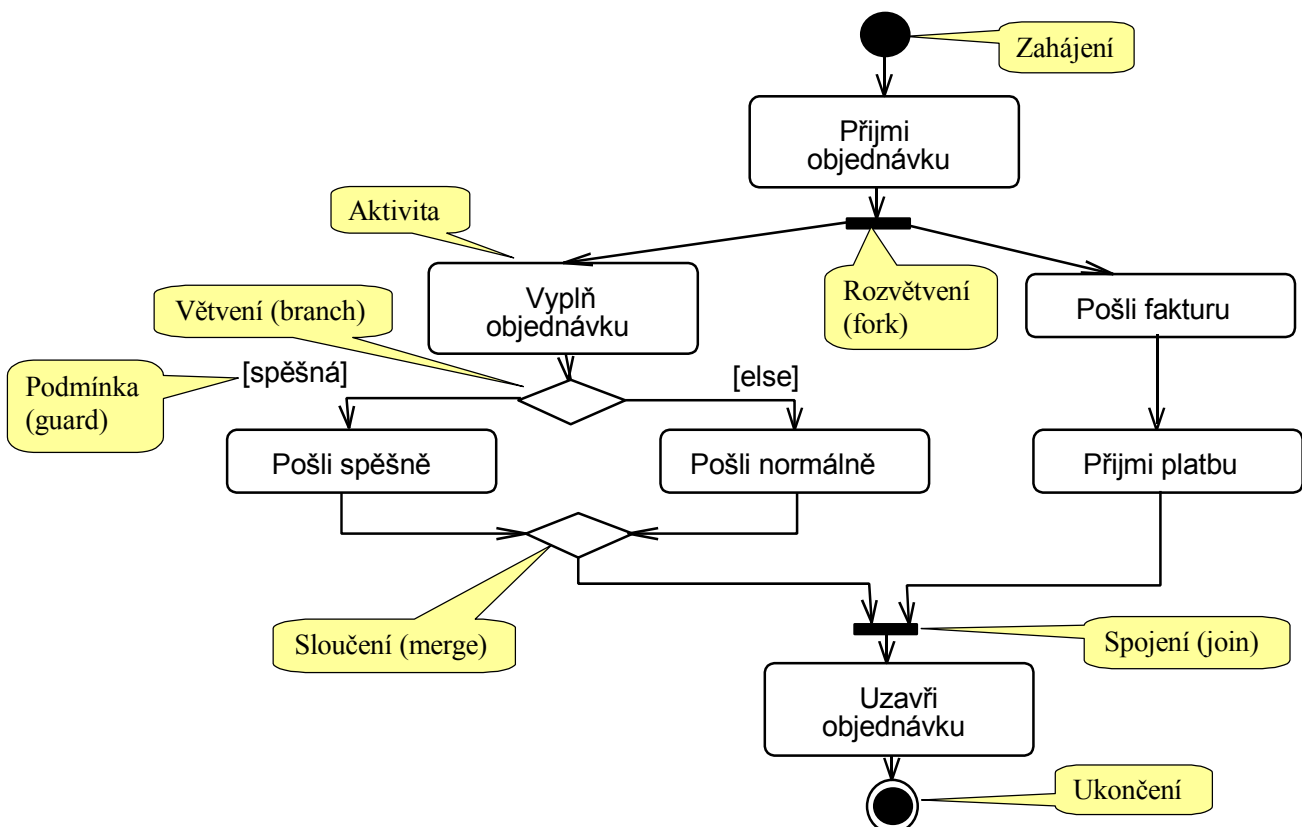
- složený stav může obsahovat několik souběžných stavových diagramů
- stav je dán kombinací stavů souběžných diagramů



- Kdy použít stavové diagramy

- vhodné pro popis chování jednoho objektu přes několik případů použití, resp. v průběhu celého životního cyklu
- pouze pro třídy se zajímavým chováním (objekty uživatelského rozhraní, řídicí objekty, ...)

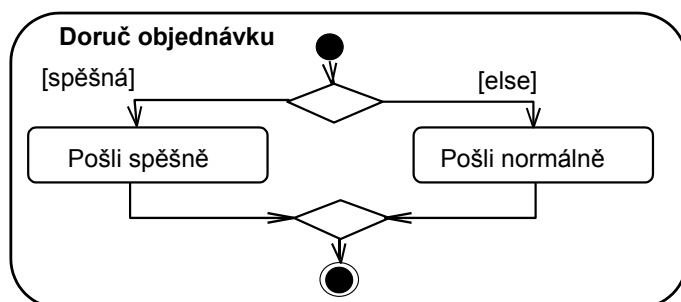
### 7.6.3 Diagramy aktivit



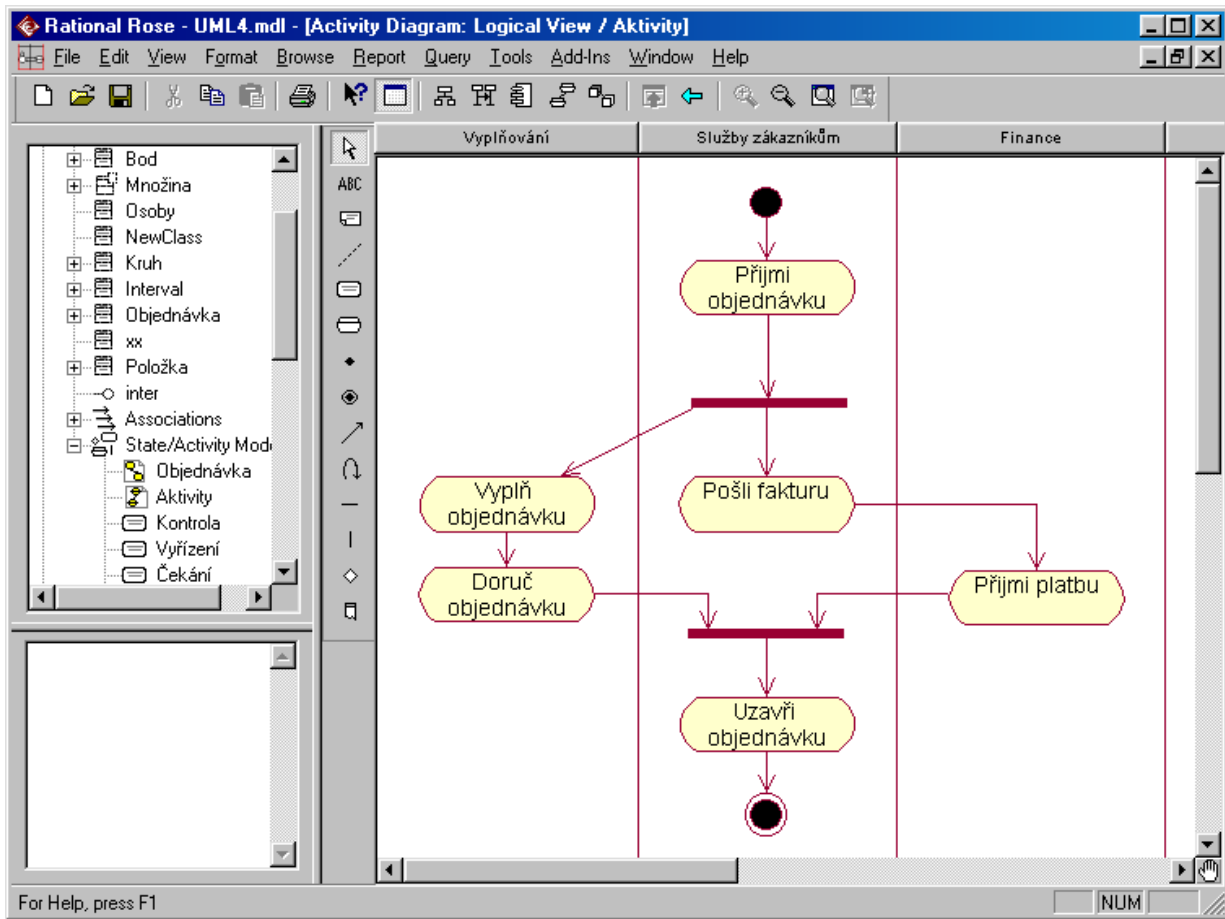
- nemají původ v technikách autorů UML
  - vývojový diagram ukazující tok řízení z jedné aktivity do druhé
  - obdoba stavového diagramu, v němž stavy reprezentují vykonávání aktivit a přechody jsou vyvolány ukončením aktivity
  - popisuje řazení aktivit s podporou jak sekvenčního, tak paralelního chování
- Prvky diagramu:
    - Stav aktivity (aktivita)** – stav nějaké činnosti, běžně krok nějakého algoritmu (procedury) nebo toku činnosti (workflow).
    - Větvení** – jeden vstupní a několik ohodnocených výstupních přechodů, jen jedna podmínka splněna.
    - Sloučení** – označuje konec větvení.
    - Rozvětvení** - jeden vstupní a několik výstupních přechodů, následkem vstupního přechodu jsou provedeny všechny výstupní paralelně (pořadí není relevantní).
    - Spojení** – výstupní přechod je proveden, až byly ukončeny všechny vstupní aktivity (synchronizace).

Diagram aktivit ukazuje podstatné řazení aktivit (sekvenční části), u ostatních umožňuje zvolit pořadí (paralelní části).

- důležité pro modelování podnikatelských (business) procesů
- užitečné pro paralelní programy
- aktivitu lze rozložit (podobně jako stav)



- Zóny (swimlanes)
  - diagram aktivit neříká nic o přiřazení aktivit objektům tříd → lze rozdělit aktivity do zón, které reprezentují zodpovědnosti tříd (resp. osob, oddělení při modelování aplikační domény)



- **Kdy použít diagramy aktivit**

**Výhoda:** podpora paralelního chování → modelování toku činností (workflow), vícevláknového (multithreaded) programování.

**Nevýhoda:** nejasná vazba mezi aktivitami a objekty.

**Vhodné situace pro použití:**

- analýza případu použití
- pochopení toku činností
- popis komplikovaného sekvenčního algoritmu (vývojový d.)
- vícevláknové aplikace

**Nepoužívat:**

- popis spolupráce objektů
- popis chování jednoho objekt v průběhu jeho života
- reprezentace složitého větvení

## 7.6.4 Diagramy fyzické organizace

- diagramy nasazení (deployment) a diagramy komponent (viz dále)