



Chain of Responsibility

Reťaz zodpovednosti

Zabraňuje previazaniu odosielateľa požiadavky s jej príjemcom tak, že umožňuje túto požiadavku spracovať viacerým objektom. Reťazí objekty prijímateľov a posúva požiadavku po tejto reťazi pokiaľ ju nejaký objekt nespracuje.

Avoid coupling the sender of a request to its receiver by giving more than one object a chance to handle the request. Chain the receiving objects and pass the requests along the chain until an object handles it.

Návrhové vzory



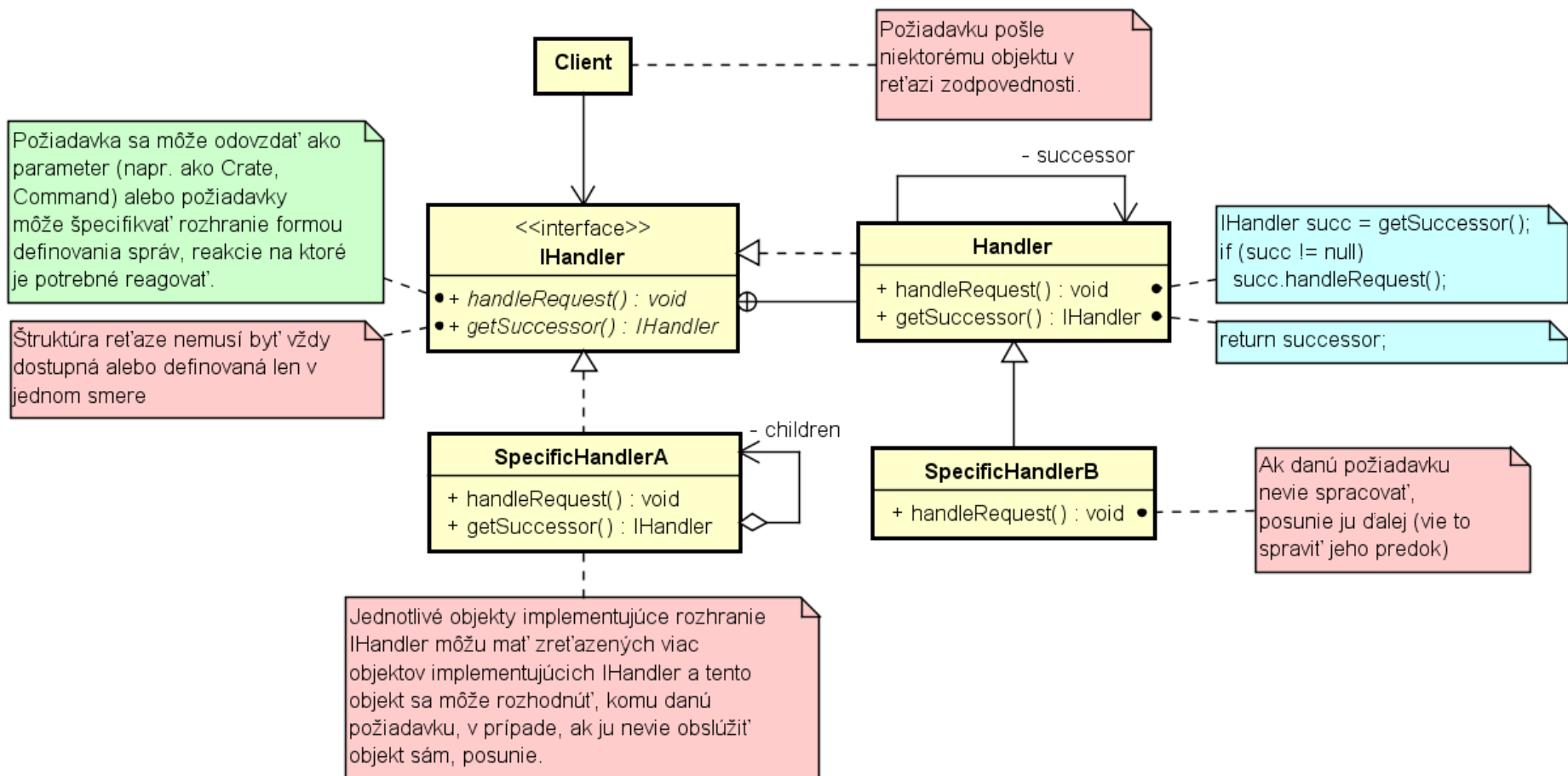
Motivácia

- Študent si chce nechať uznať známku z predmetu (požiadavka) - za kým pôjde najskôr a ako bude postupovať?
- Majme relatívne zložitú aplikáciu s užívateľským rozhraním. Užívateľ stlačí F1 (typicky nápoveda) – ako zareaguje aplikácia?
 - Čo ak práve aktívny prvok žiadnu nápovedu neposkytuje?

Aplikovateľnosť

- Reťaz zodpovednosti je vhodné použiť, keď:
 - viac ako jeden objekt môže spracovať požiadavku a tento objekt nie je vopred známy.
 - chcete požiadať o spracovanie požiadavky objekt bez toho, aby ste tento objekt explicitne uviedli.
 - by mala byť množina objektov, ktoré môžu spracovať požiadavku, definovaná dynamicky.

Implementácia



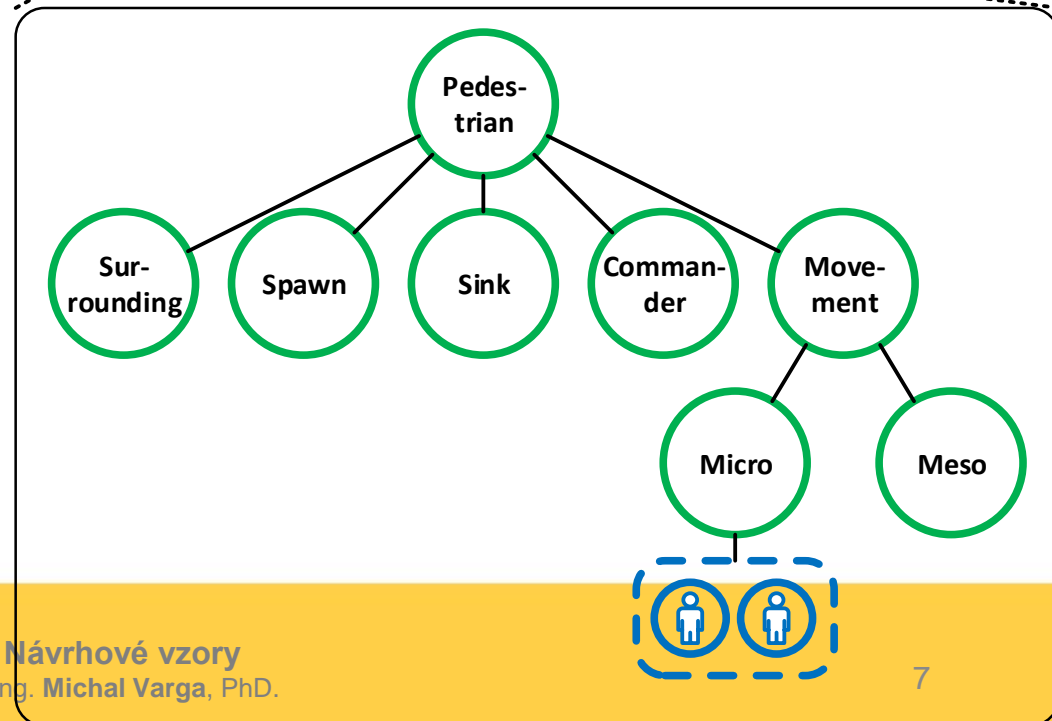
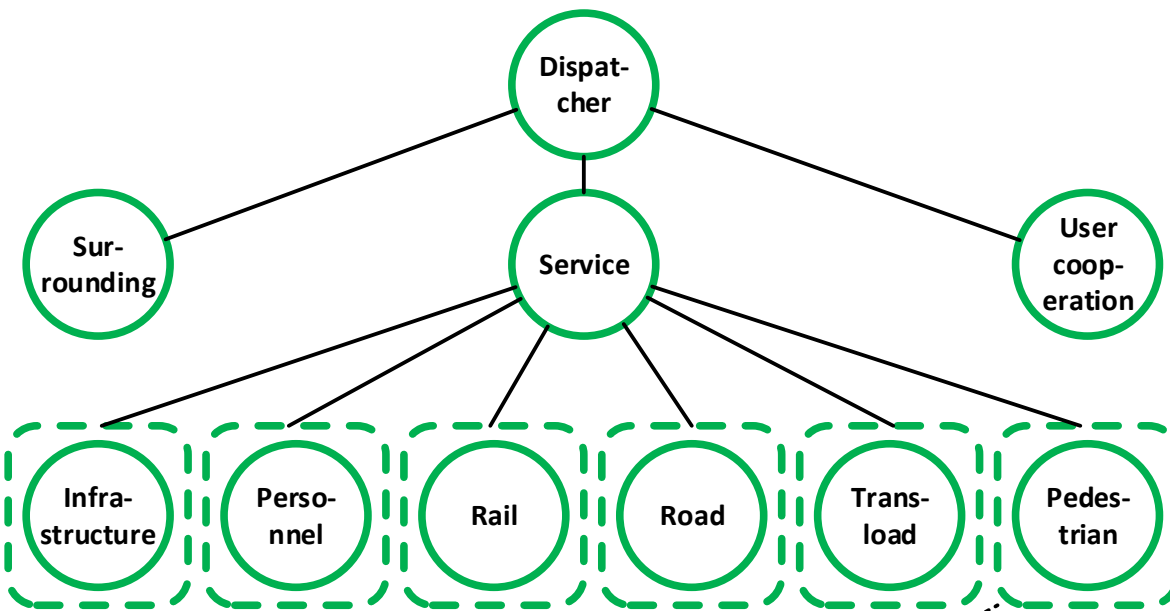
Dôsledky

- Redukuje previazanosť objektov.
 - Odosielateľ požiadavky nemusí poznať príjemcu, len vie, že sa požiadavka dostane na správne miesto, kde sa spracuje.
 - Príjemca nemusí poznať odosielateľa a dokonca nemusí mať znalosť o štruktúre reťaze.
- Pridáva dodatočnú funkčnosť tým, že distribuuje kompetencie medzi jednotlivé objekty a umožňuje potenciálne zmeny za behu.
- Nemôžete sa vždy spoliehať, že požiadavka
 - bude obslúžená konkrétnym objektom,
 - a ani, že bude vôbec obslúžená (reťaz môže byť zle nakonfigurovaná)!

Príklad

- Komunikácia v rámci agentovej štruktúry:
 - Vyplnením adresáta (získať sa dá rôzne, napr. pomocou zlatých stránok).
 - Čiastočné adresovanie (doručenie správy na submodel).
 - **Neadresne**, zverejnením poskytovaných služieb.

Príklad



Príbuzné vzory

- *Composite* – často sa spája s týmto vzorom, nakoľko rodič Komponentu (iný Komponent) slúži ako nasledovní v reťazi.

Upozornenie

- Tieto študijné materiály sú určené výhradne pre študentov predmetu 5I132 Návrhové vzory (Design Patterns) na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline.
- Reprodukovanie, šírenie (i častí) materiálov bez písomného súhlasu autora nie je dovolené.

Ing. Michal Varga, PhD.
Katedra informatiky
Fakulta riadenia a informatiky
Žilinská univerzita v Žiline
Michal.Varga@fri.uniza.sk

