

6t – 21.3.2016 – pondelok, 30.3.2016 - streda

- ✓ opakovanie a otázky k domácemu štúdiu
 - body prípadov použitia
 - modelovanie údajov – vzťahy, príklady
- ✓ diagram tokov údajov
- ✓ modelovanie správania
- ✓ architektonický návrh softvéru
- **úlohy:**
 - študovať...

IMAGINE CUP

**po prednáške –
dohodneme termín stretnutia**

30.3.2016 - Michal Barla: Manažment verzií

6.4.2016 - Jakub Šimko: Architektúry, návrhové vzory a webové rámce v tvorbe webových aplikácií

27.4.2016 - Ján Suchal: Vybrané témy z implementácie webových aplikácií

Experiment s ALEF TNG

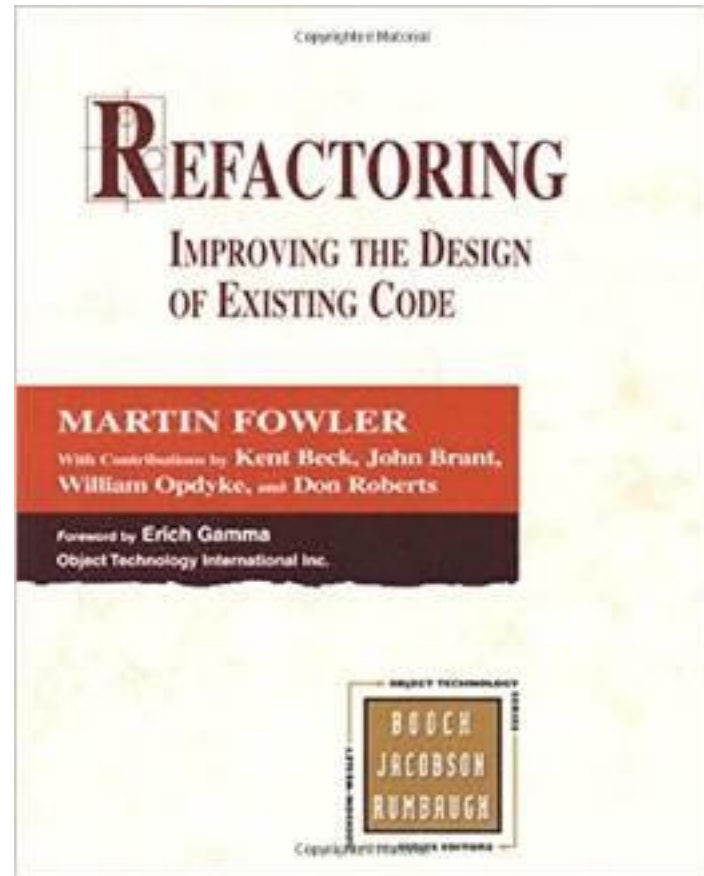
**kto sa nechce zúčastniť,
dajte mi vedieť**



Catalog of Refactorings



Martin Fowler
10 December 2013



EDÍCIA UČEBNÝCH TEXTOV
INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

ARCHITEKTÚRA SOFTVÉROVÝCH SYSTÉMOV

ARCHITEKTÚRA INTERNETOVÝCH SYSTÉMOV A ARCHITEKTÚRA ORIENTOVANÁ NA SLUŽBY

LUBOR ŠEŠERA, PETER GREC, PAVOL NÁVRAT



Architektúra softvérových systémov

Autorom sa podarilo v predložennom texte uviesť skutočne obdivuhodné množstvo poznatkov z oblasti architektúry softvérových systémov a to na viacerých úrovniach abstrakcie. Je vidieť nielen nadsled autorov, ale aj hlbokú znalosť súčasných technológií, ktoré v učebnom texte prezentujú.

prof. Ing. Mária Bielíková, PhD., z recenzného posudku

Predložený učebný text považujem za výborne spracovanú učebnicu, ktorá sa venuje veľmi aktuálnej oblasti a spája v sebe charakteristiku problému, abstrakciu jeho dôležitých aspektov, uvedenie možných riešení prehľadným a adekvátne formalizovaným spôsobom a v neposlednom rade sú tieto riešenia primerane ilustrované na konkrétnych a aktuálne používaných technológiách.

Ing. Marek Paralič, PhD., z recenzného posudku

- Význam architektúry softvérového systému
- Architektonické pohľady a štýly
- Príklad modelovania architektúry systému na elektronické bankovníctvo v UML 2
- Princípy a vzory na tvorbu vrstiev internetového a intranetového systému
- Príklady v technológiách EJB 3, JPA, JSP, JSF
- Spôsoby integrácie systémov a MOM
- Architektúra orientovaná na služby s príkladmi v štandardoch SOAP, WSDL a BPEL

RNDr. Lubor Šešera, PhD. je konzultant firmy SOFTEC. Externe prednáša na FIIT STU v Bratislave, kde vybudoval predmet Aplikácie architektúry softvérových systémov. Je autorom viacerých medzinárodných publikácií, najmä na konferenciách PLoP a EuroPloP.

Ing. Peter Grec je hlavný architekt firmy SOFTEC. Zameriava sa na architektúru orientovanú na služby (SOA). Je architekt viacerých systémov v oblasti poisťovníctva, bankovníctva a telekomunikácií.

Prof. Ing. Pavol Návrat, PhD. je riaditeľ Ústavu informatiky a softvérového inžinierstva FIIT STU. Garantoval zavedenie študijného programu softvérové inžinierstvo ako prvého na Slovensku a vybudoval preň predmet Architektúra softvérových systémov.

**Design is there to enable you to
keep changing the software easily
in the long term.**

"code and fix" nightmare

Základné koncepty softvérového návrhu

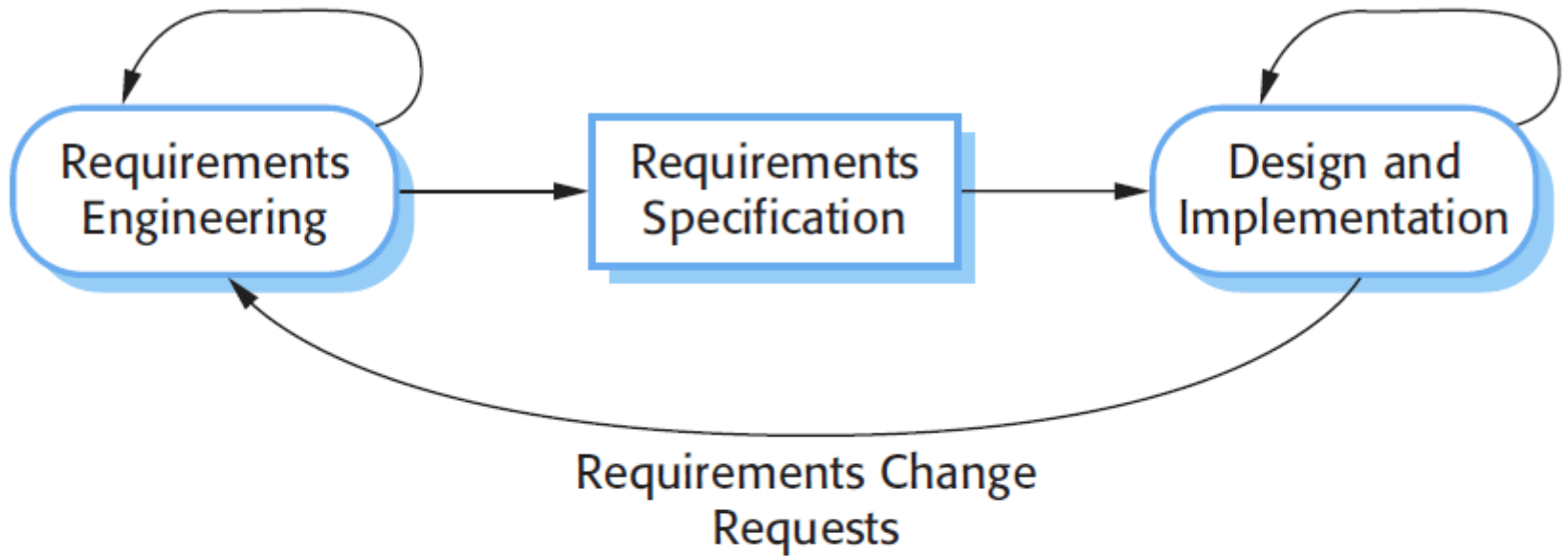
- Abstrakcia
- Vzory – architektonické, návrhové, implementačné (coding patterns, idioms)
- Dekompozícia, modularita
- Ukrytie informácií (efektívna modularita)
- Funkcionálna nezávislosť (dôsledok modularity) -> súdržnosť a zviazanosť
- Stratégie návrhu – postupné zjemňovanie (stepwise refinement)

Dobrý návrh

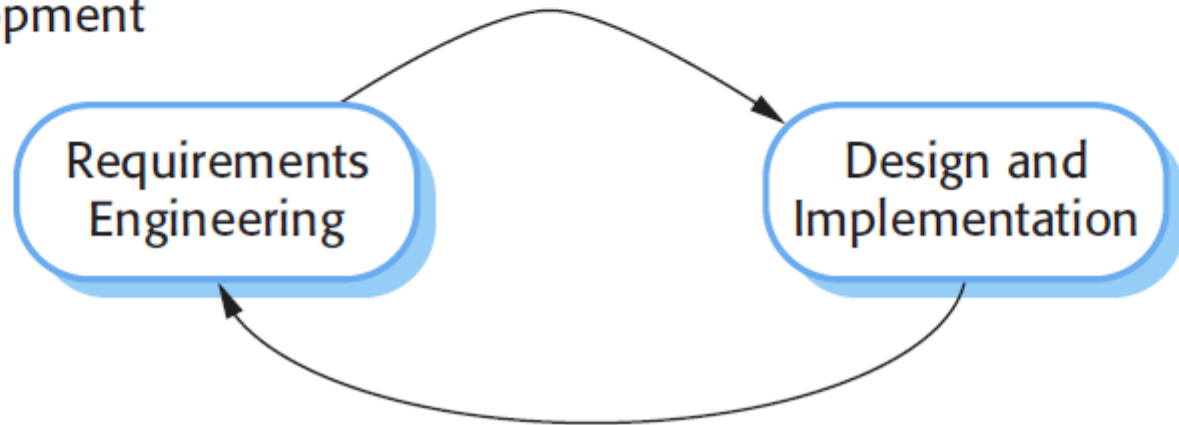
- voľne zviazané softvérové súčiastky
- veľká súdržnosť jednotlivých súčiastok

„Klasický“ návrh vs. evolučný návrh

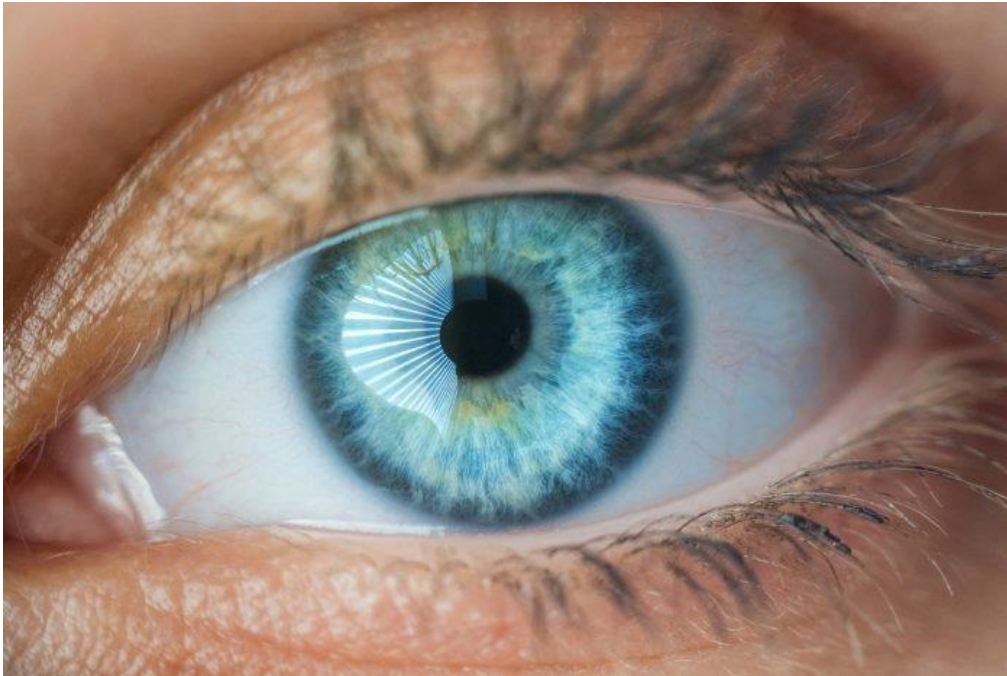
Plan-Based Development



Agile Development



Prifotťe sa na FIIT...



Zalogujte svoj pohľad na život okolo nás vo FIITAPIXEL.

<http://foto.fiit.stuba.sk/>

27. marca 2016

... trinásť kôl, vyše 6800 fotografií, 1030 účastí fotografov, cca 36 000 lajkov a troch odborných porotcov z Filmovej a televíznej fakulty, Ateliéru kameramanskej tvorby a fotografie Vysokej školy múzických umení v Bratislave.

Softvérová architektúra

- disciplína informatiky zaoberajúca sa navrhovaním štruktúry softvérového systému a jeho zložiek, ktoré majú stanovené funkcie a vzájomné vzťahy
- schéma, ktorá vyjadruje základnú štrukturálnu organizáciu jednotlivého softvérového systému alebo špeciálnej triedy softvérových systémov
- poskytuje spôsoby abstraktného opisu štruktúry softvérových systémov a princípy a pravidlá, ktorými sa riadi ich návrh a vývoj v čase

Softvérová architektúra

- základná organizácia systému stelesnená v **komponentoch**, ich **vzájomných vzťahoch** a prostredí a **princípoch** [*a rozhodnutí*], ktoré viedli k jej návrhu a vývoju **(IEEE)**
- štruktúra alebo štruktúry systému, ktoré zahŕňa **softvérové elementy**, zvonku viditeľné **vlastnosti** týchto elementov a **vzťahy** medzi nimi **(SEI)**

Architektúra

- redukuje celkovú zložitosť a umožňuje lepšie porozumieť vytváranému systému
- dokumentuje výber dôležitých rozhodnutí; zabraňuje tak opätovnému „vymýšľaniu kolesa“ v budúcnosti
- slúži ako báza pre konštrukciu systému: stanovuje pre programátorov funkčné elementy, ktoré majú implementovať, resp. s ktorými sa majú integrovať
- poskytuje primárny nástroj na komunikáciu medzi účastníkmi softvérového procesu: v rámci vývojového tímu aj so zákazníkom
- uľahčuje vzdelávanie účastníkov: zjednodušuje začlenenie nových ľudí do tímu tak v etape implementácie ako aj v následnej údržbe
- pomáha pri riadení projektu; jednotlivé elementy predstavujú objekty pre plánovanie, riadenie, správu verzií a dodávku zákazníkovi

Architektonické štýly

podľa Mary Shaw a Davida Garlana

- vrstvy
- dátovody a filtre
- modulárna dekompozícia
- klient-server
- úložisko údajov, tabuľa
- interpreter, produkčný systém (pravidlový systém)

Štúdium

- Šešera et al. Architektúra softvérových systémov, 1. kapitola
- Sommerville: Software Engineering
 - 3 Agile software development (s. 57-77)
 - 4 Requirements engineering (s. 83-90, 100-110)
 - 6 Architectural design (s. 148-172)

Modelovanie softvéru

Martin Fowler's advice for using diagrams

First keep in mind **what you're drawing the diagrams for**. The primary value is communication. **Effective communication means selecting important things and neglecting the less important.** This selectivity is the key to using the UML well. Don't draw every class - only the important ones. For each class, don't show every attribute and operation - only the important ones. Don't draw sequence diagrams for all use cases and scenarios - only... you get the picture.

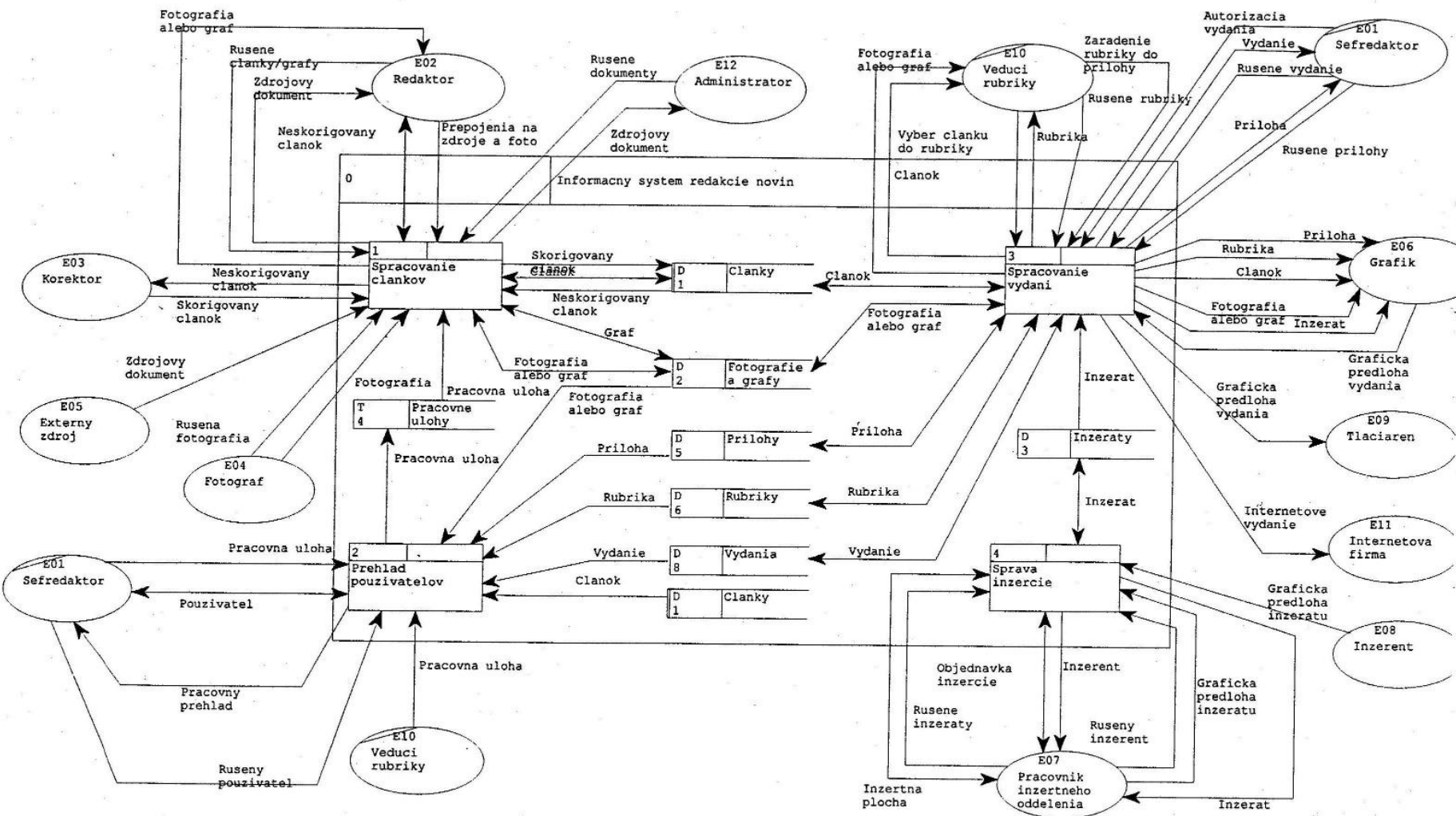
A common problem with the common use of diagrams is that people try to make them comprehensive. The code is the best source of comprehensive information, as the code is the easiest thing to keep in sync with the code.

For diagrams comprehensiveness is the enemy of comprehensibility.

Martin Fowler's advice for using diagrams

... Effective communication means selecting important things and neglecting the less important...

... For diagrams comprehensiveness is the enemy of comprehensibility...



Obr 4.1 Informačný systém redakcie novín

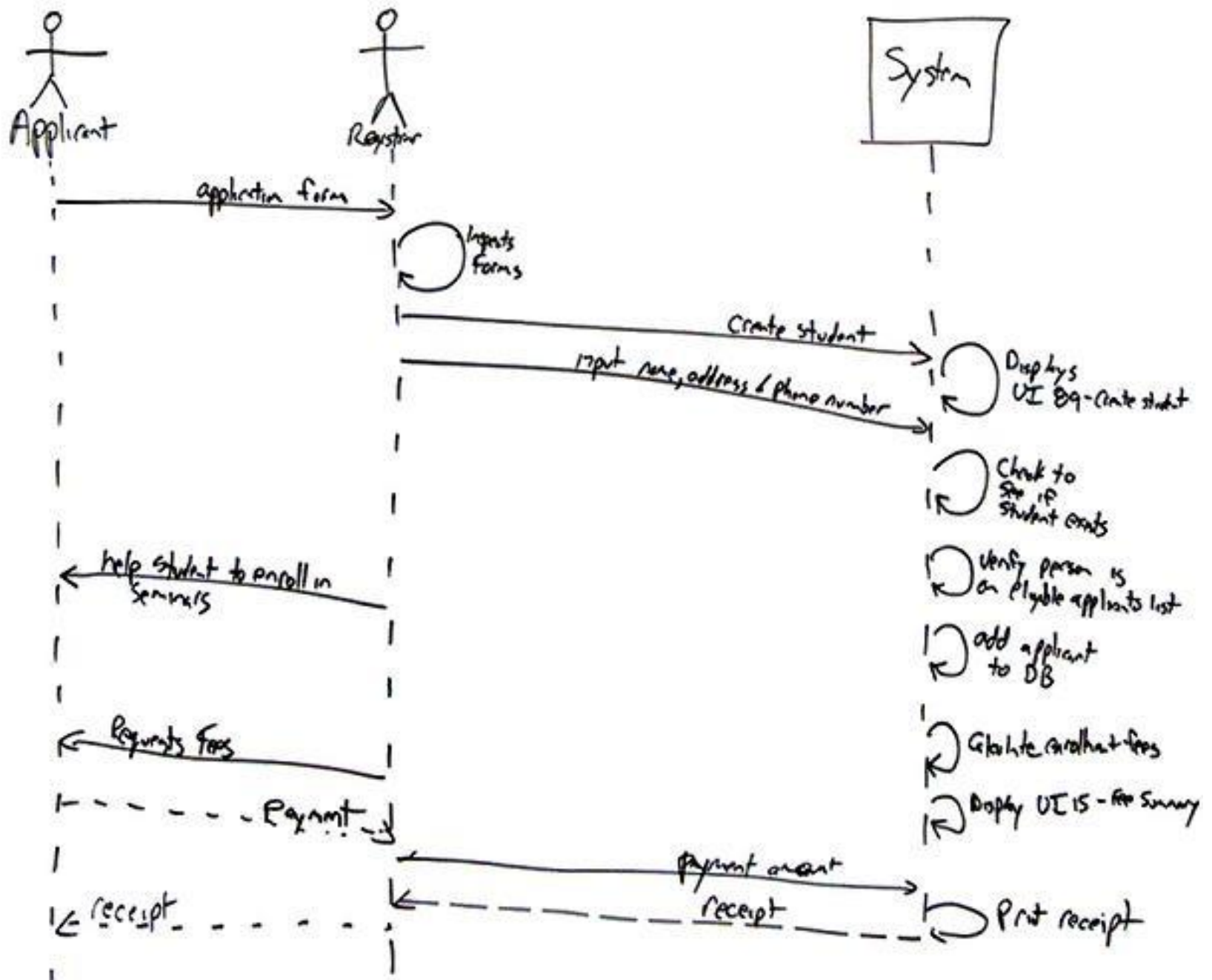
Modelovanie údajov

- Postup
- Techniky
- UML vs. Non UML

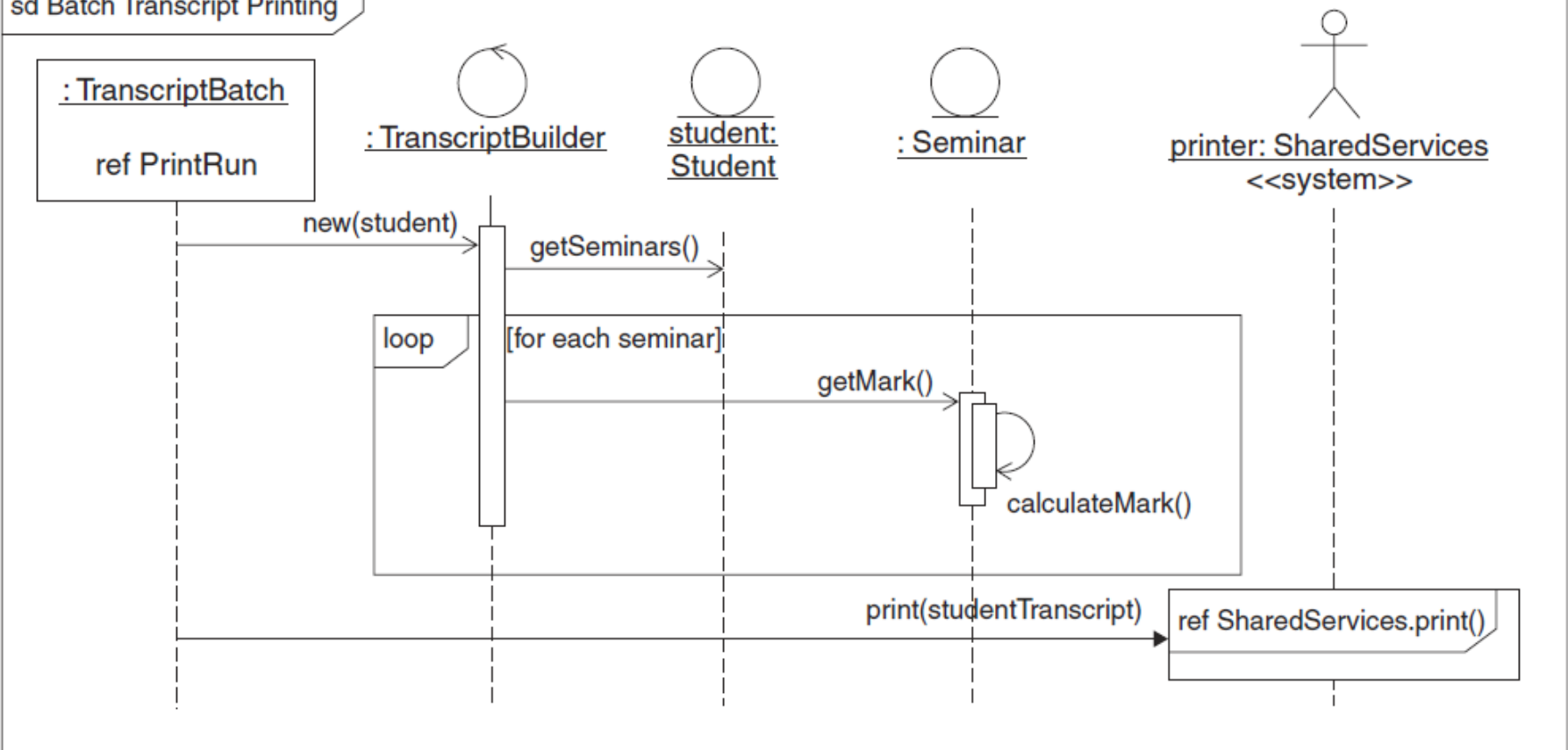
persistentné údaje vs. programové štruktúry

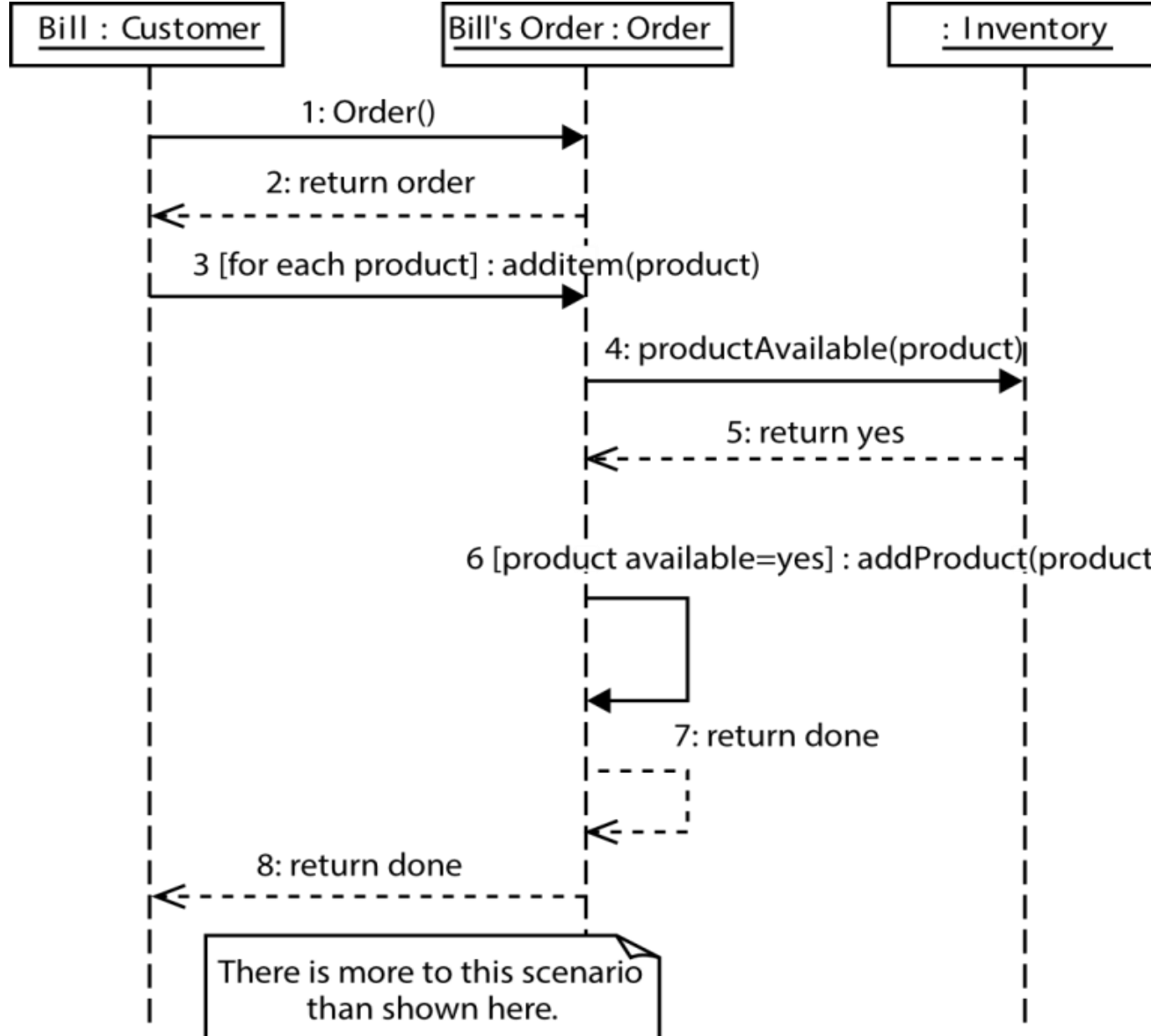
Sekvenčný diagram

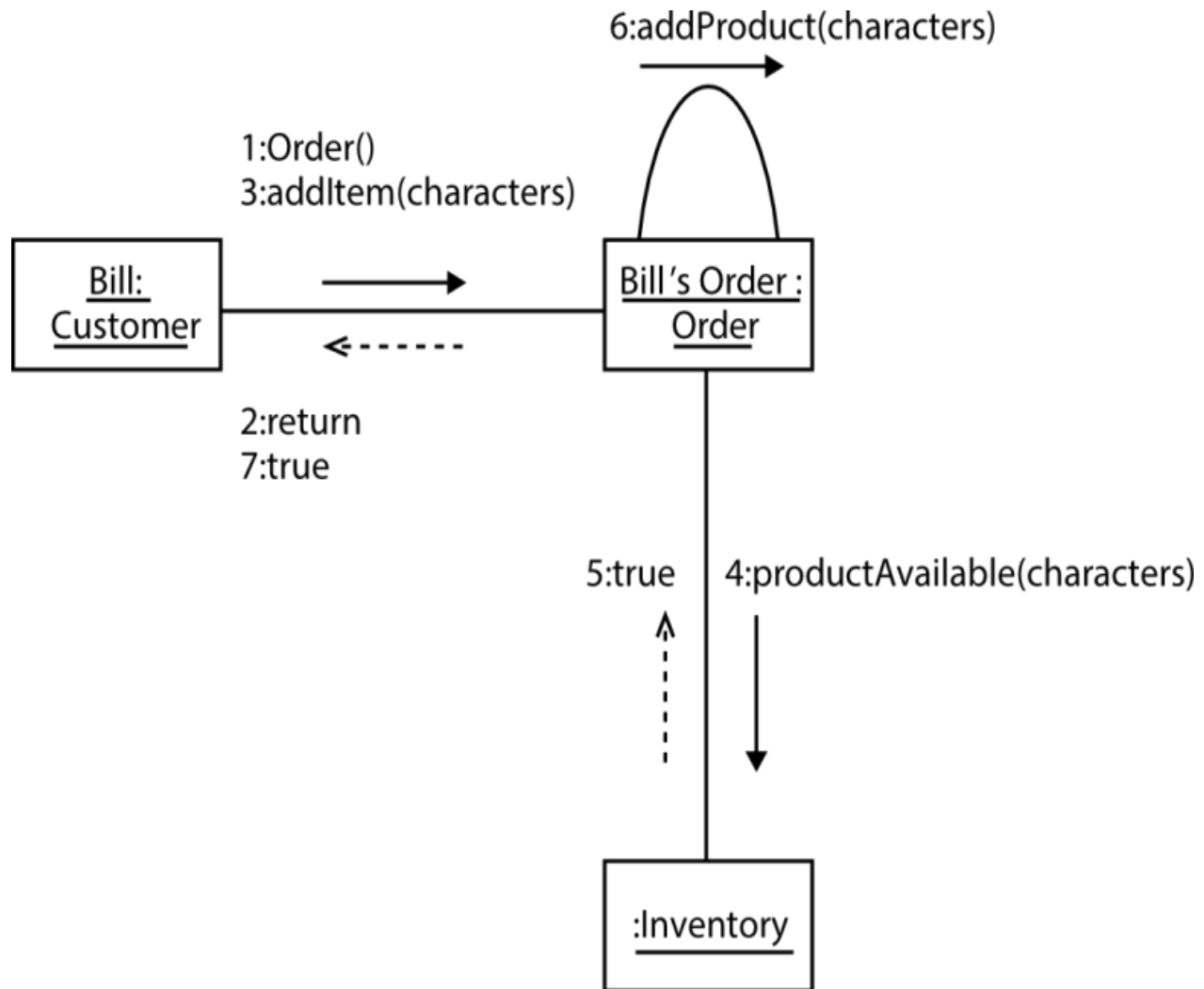
- modelovanie dynamického správania systému, podsystemu, triedy, operácie
- znázorňuje interakcie medzi objektami v časovej postupnosti
- je sémanticky ekvivalentný s **komunikačným diagramom**, ale neobsahuje vzťahy medzi objektami
- základné entity:
 - objekt (inštancia triedy)
 - správa
- ďalej môže obsahovať:
 - inštancia rozhrania, komponentu, balíka, uzla a pod., poznámka, ohraničenie

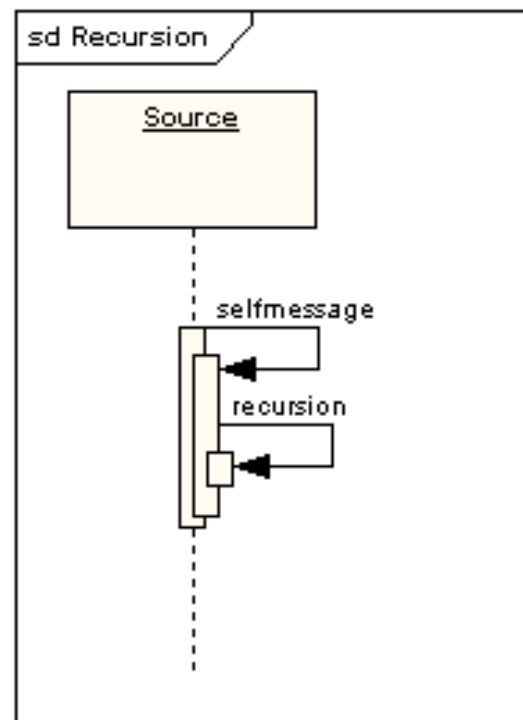
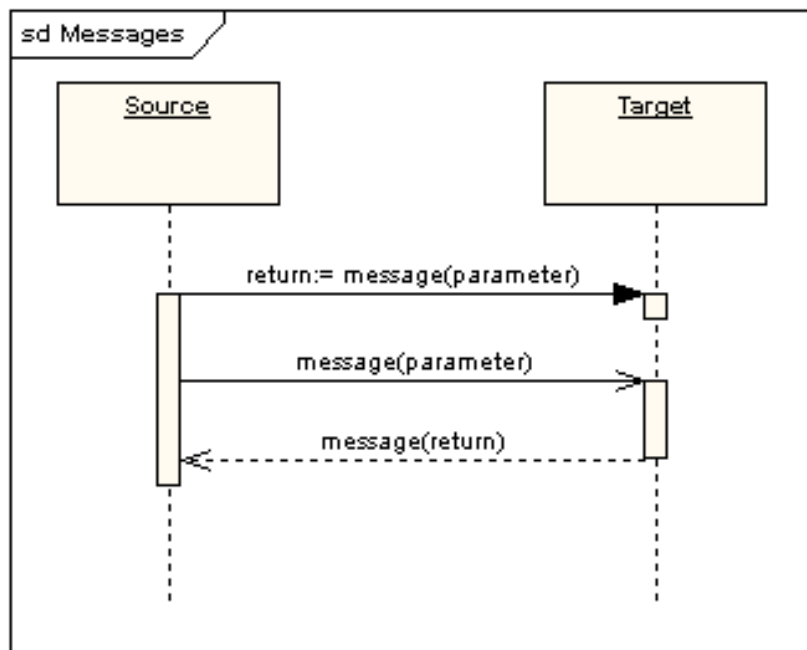
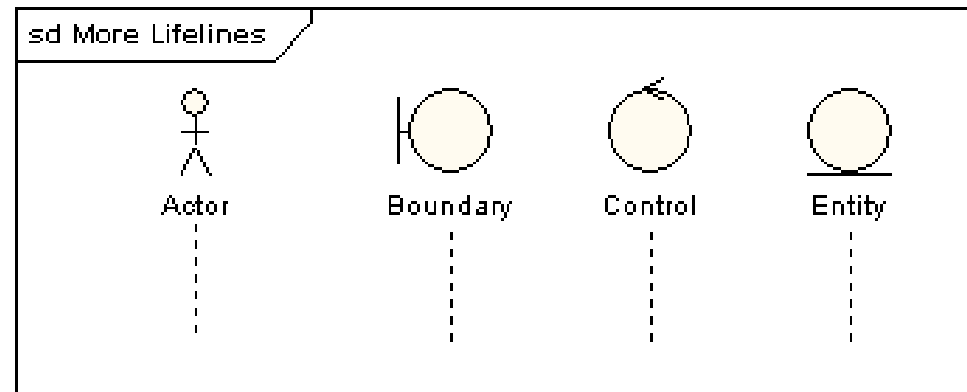
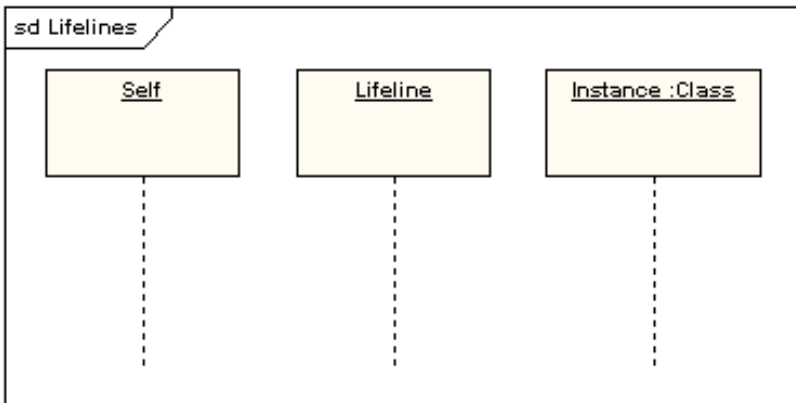


sd Batch Transcript Printing

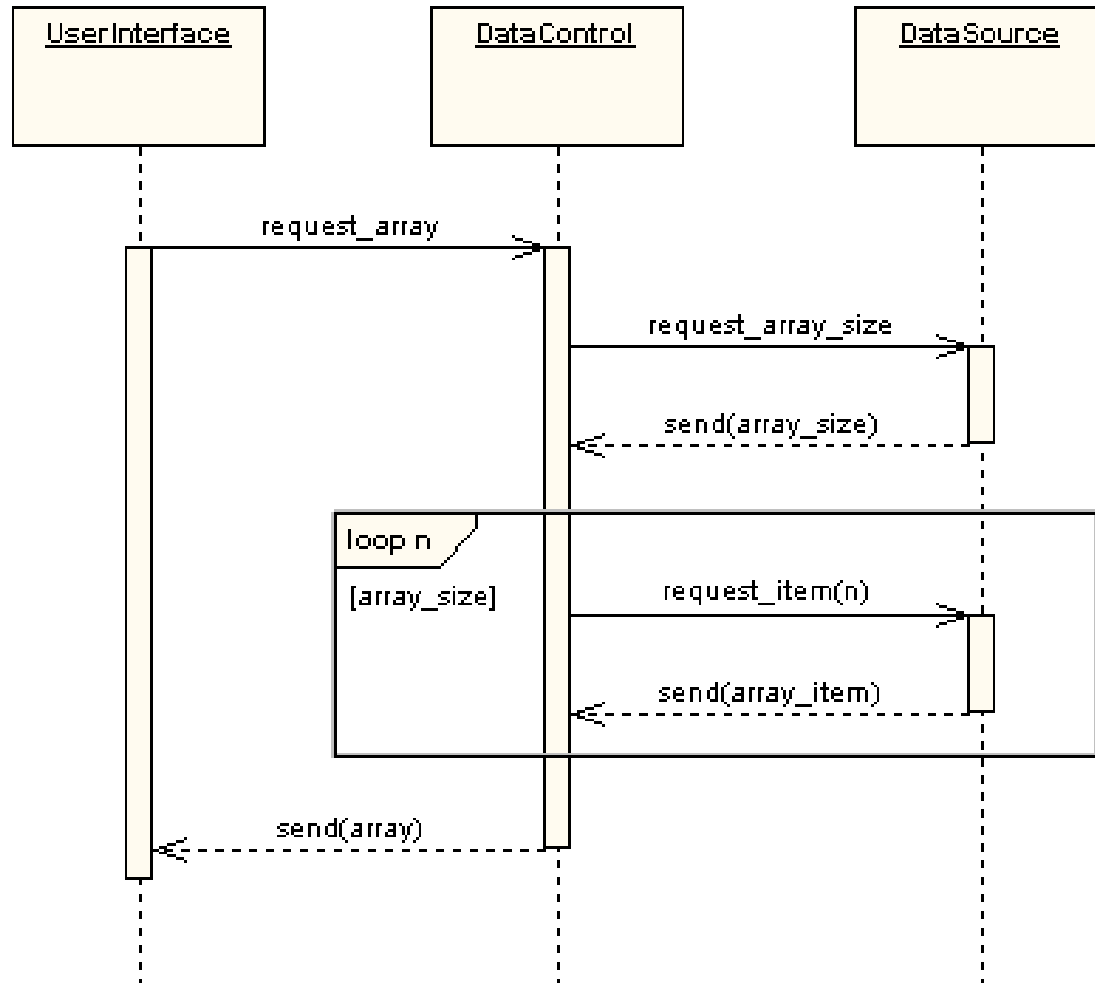


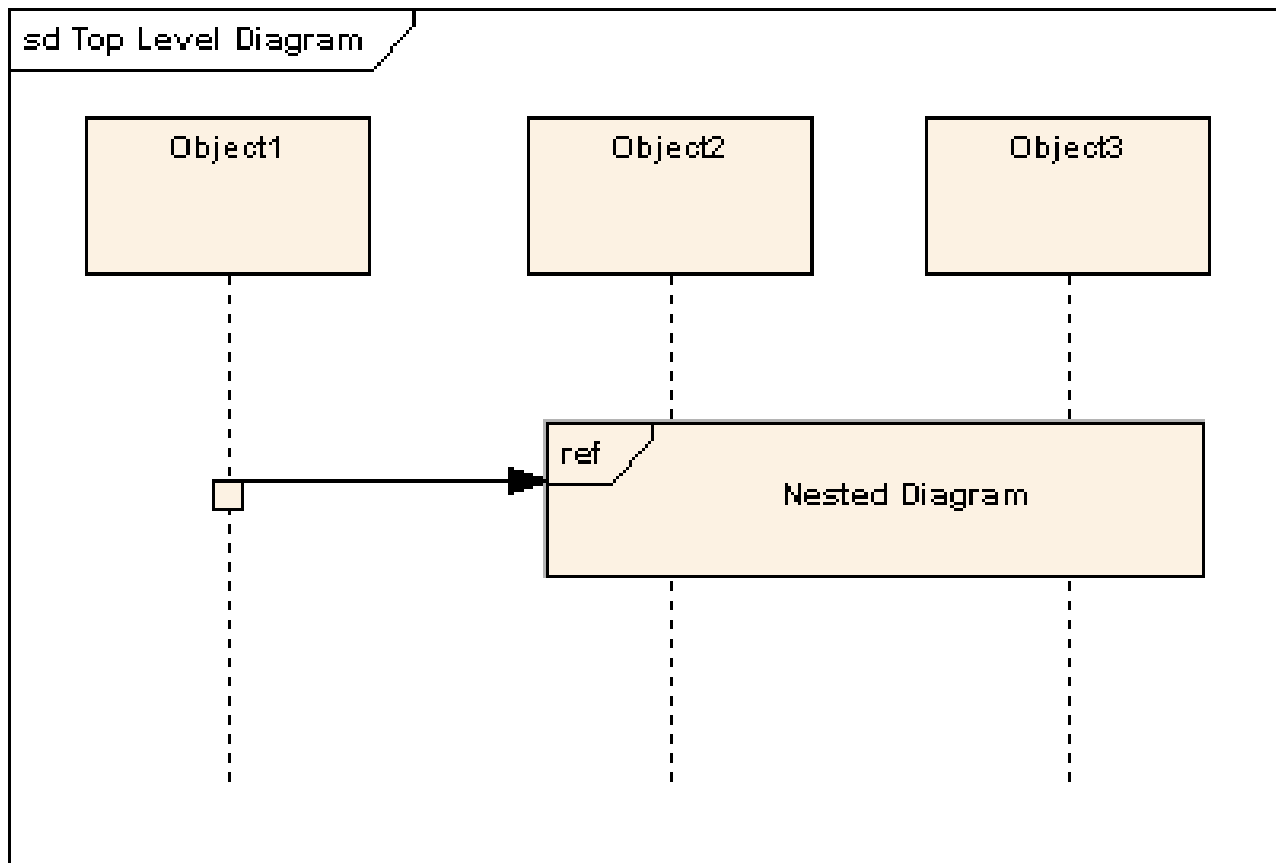




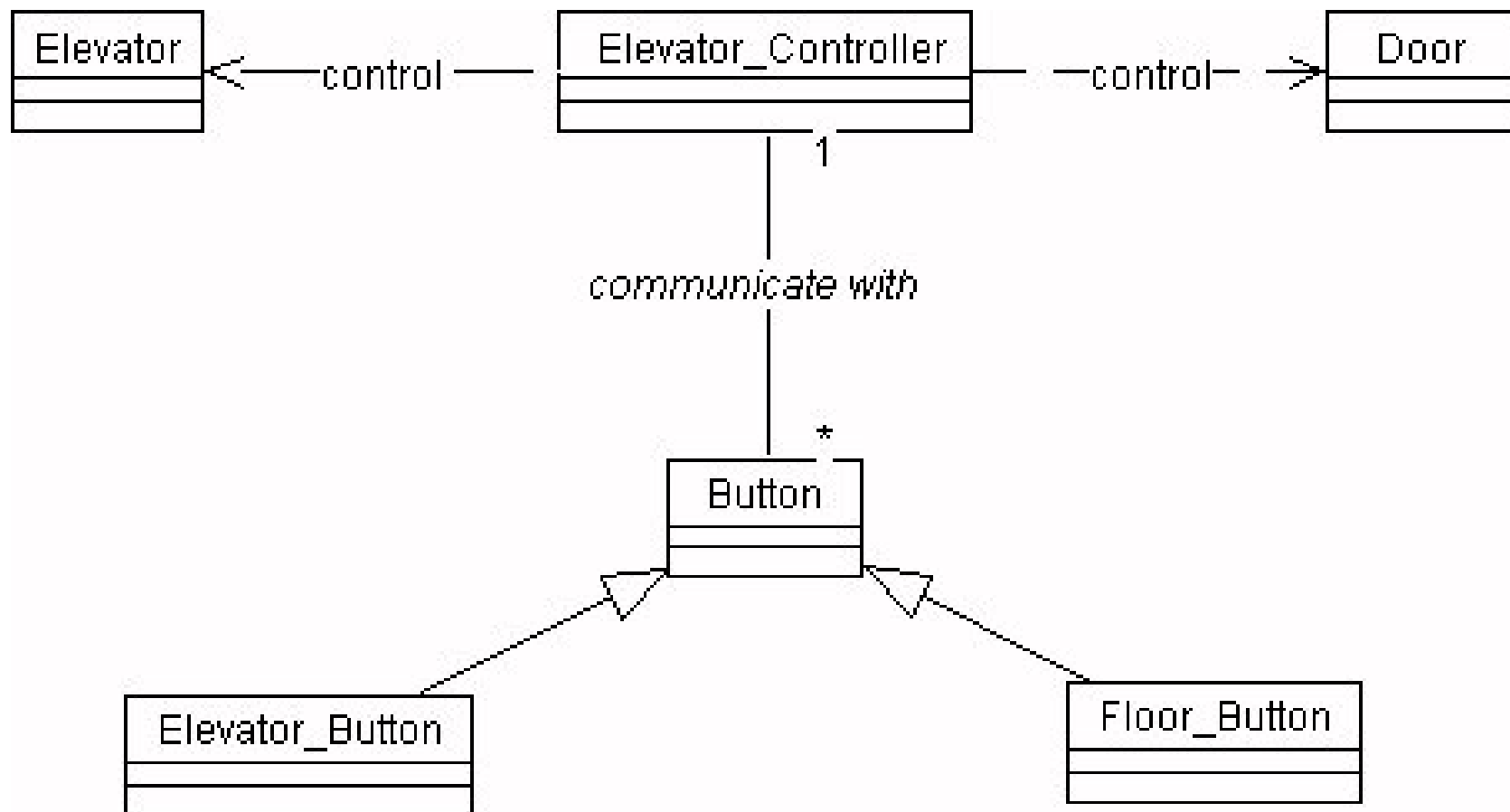


sd Fragment

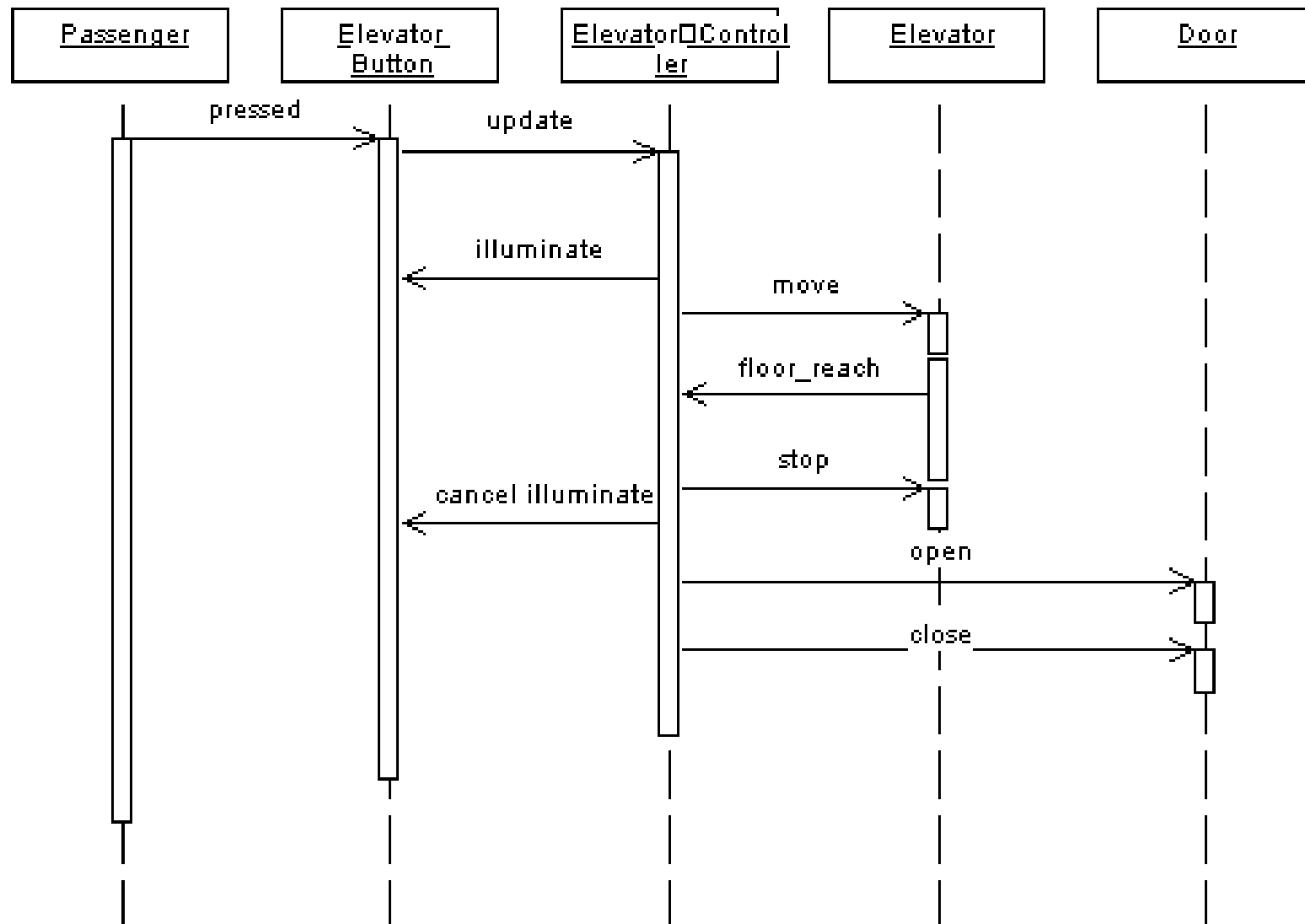




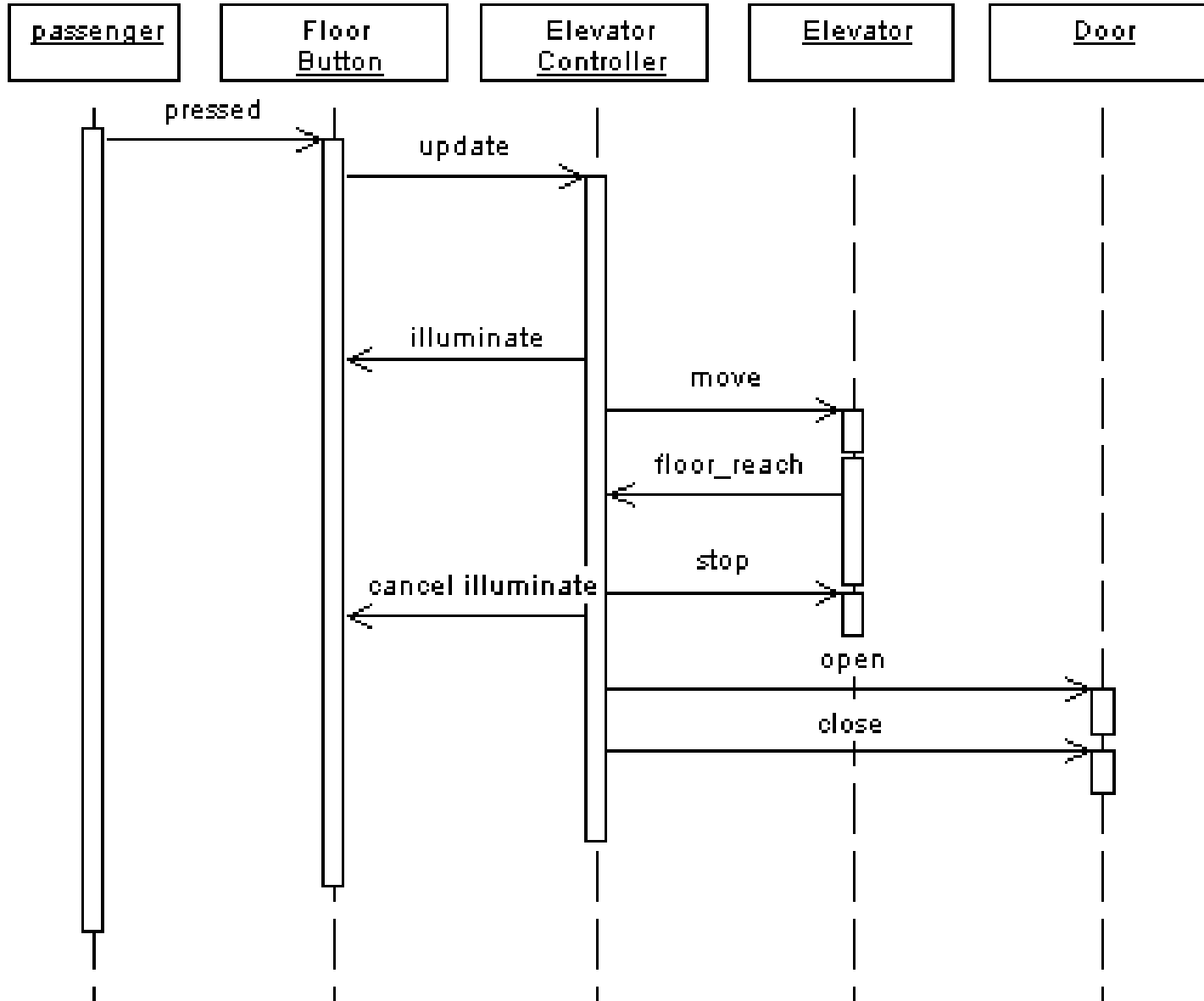
Príklad výťah



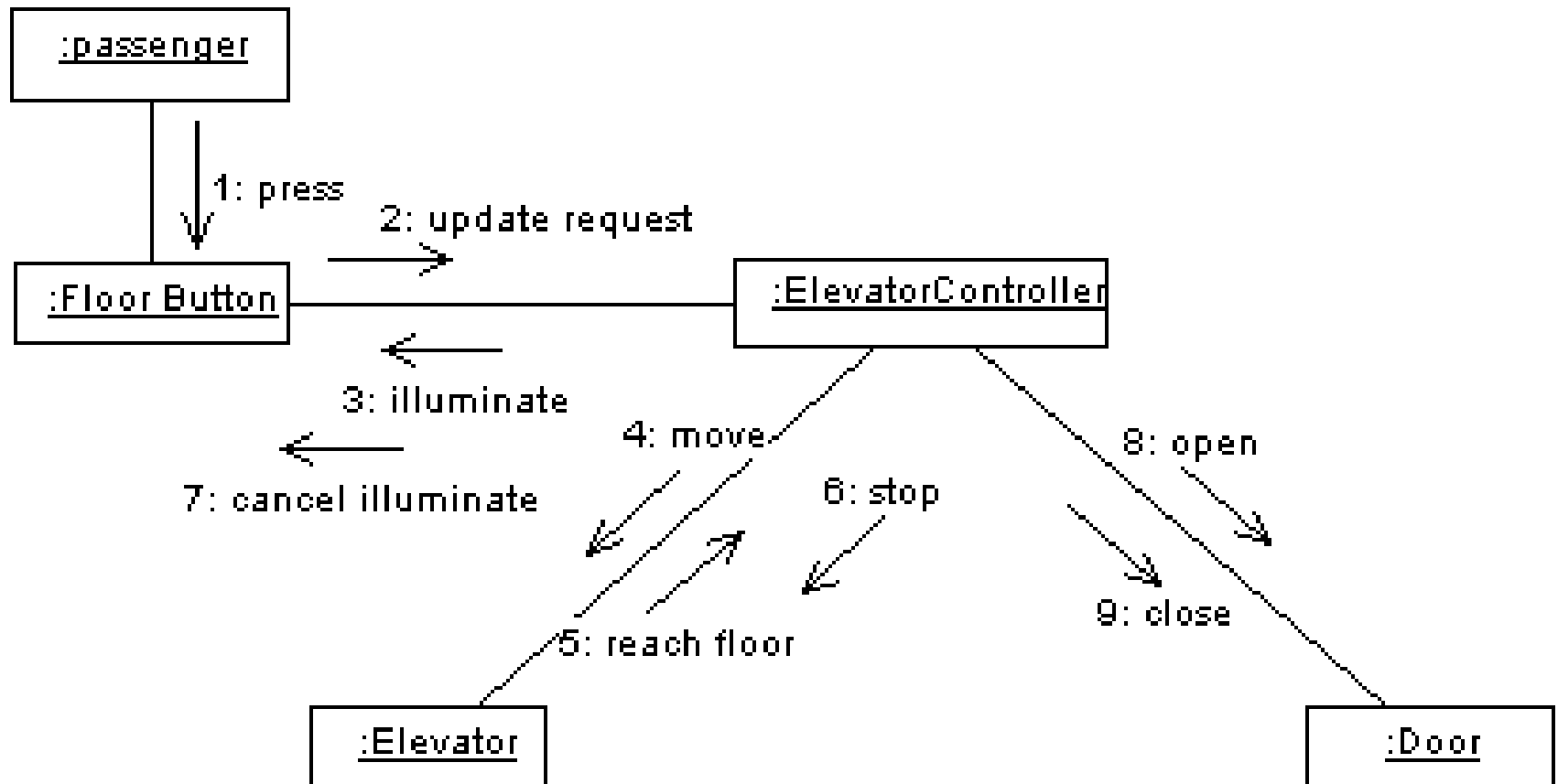
Obsluha tlačítka vo výtahu



Obsluha tlačítka na dverách výtahu

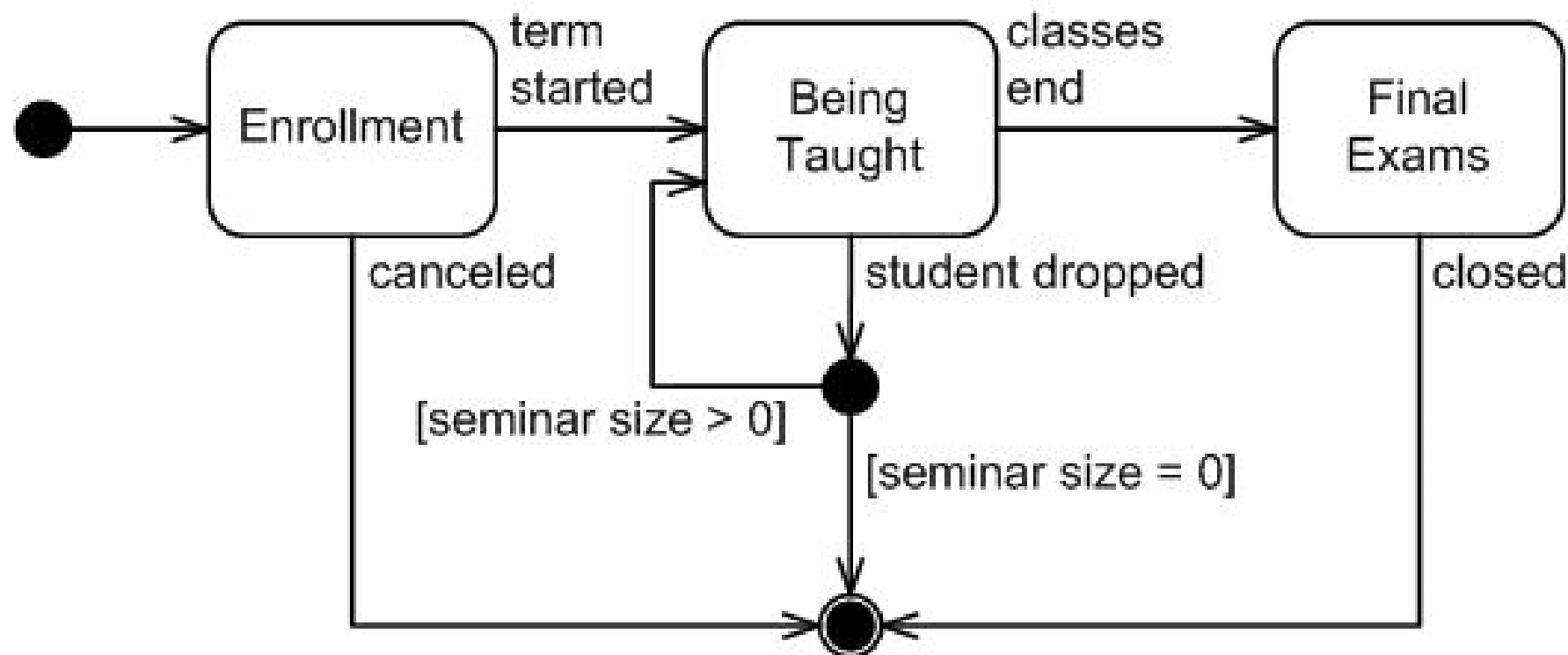


Obsluha tlačítka na dverách výtahu

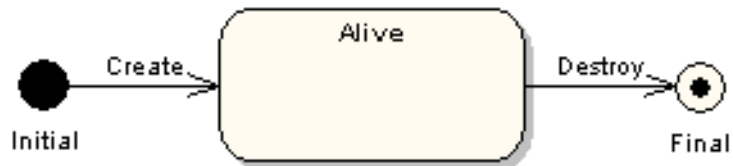


Stavový diagram

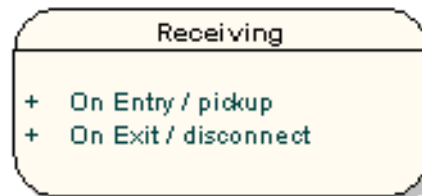
- prostriedok na vyjadrenie kauzálnych a časových súvislosti (následnosti) akcií a stavov v systéme (časti systému, triedy, údajovej entity,...)
- na systém alebo časť systému sa pozeráme ako na množinu stavov, pričom systém reaguje na určité možné udalosti
- znázorňujú sa povolené prechody medzi stavmi, v ktorých sa môže nachádzať skúmaný objekt (údajová entita, systém, časť systému)
- entity
 - stavy
 - prechody medzi stavmi



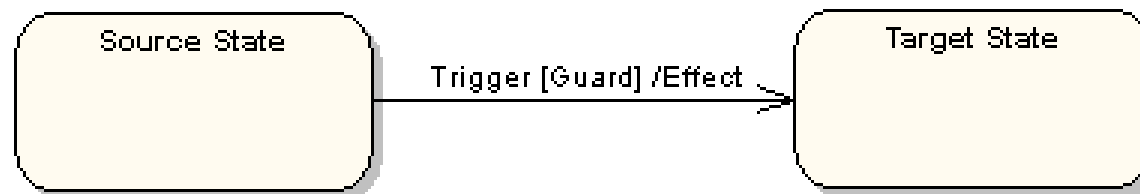
sm Initial and Final



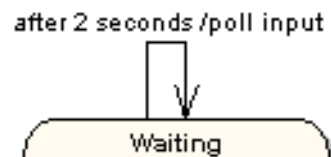
sm Entry and Exit

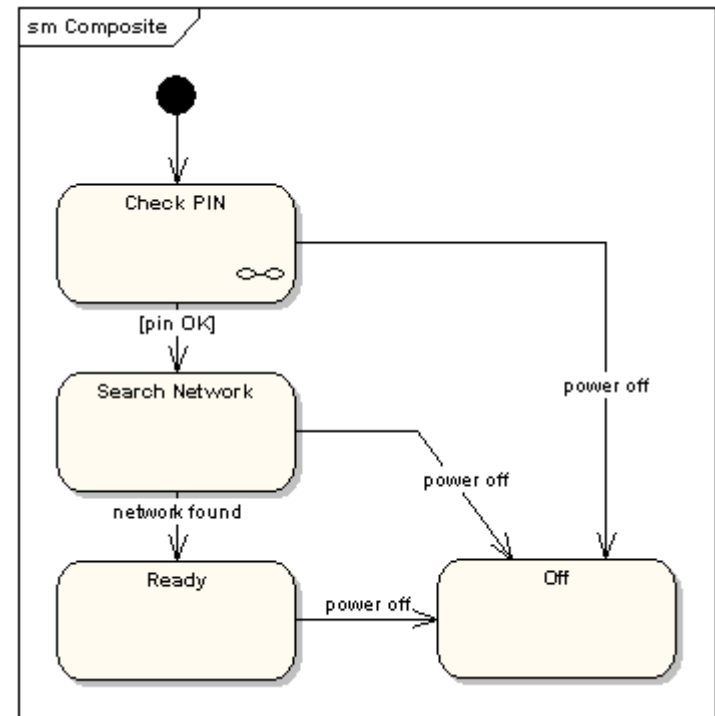
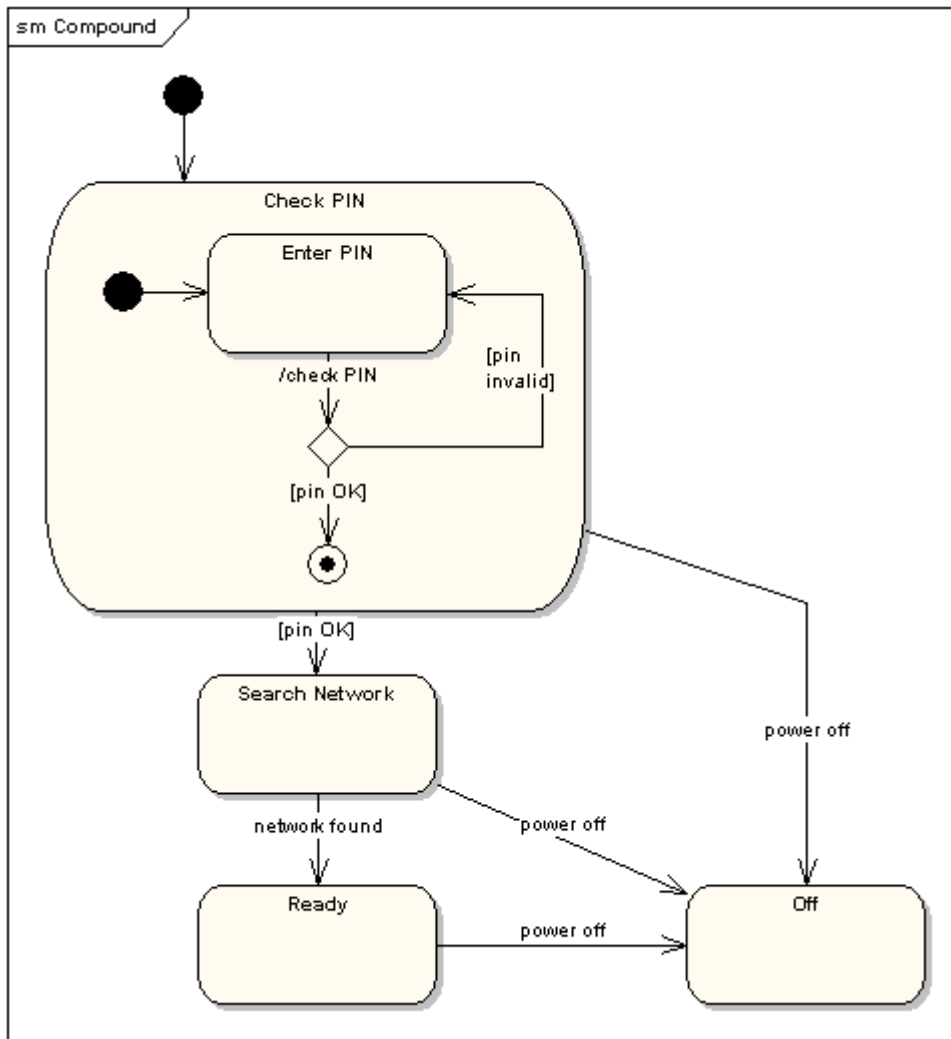


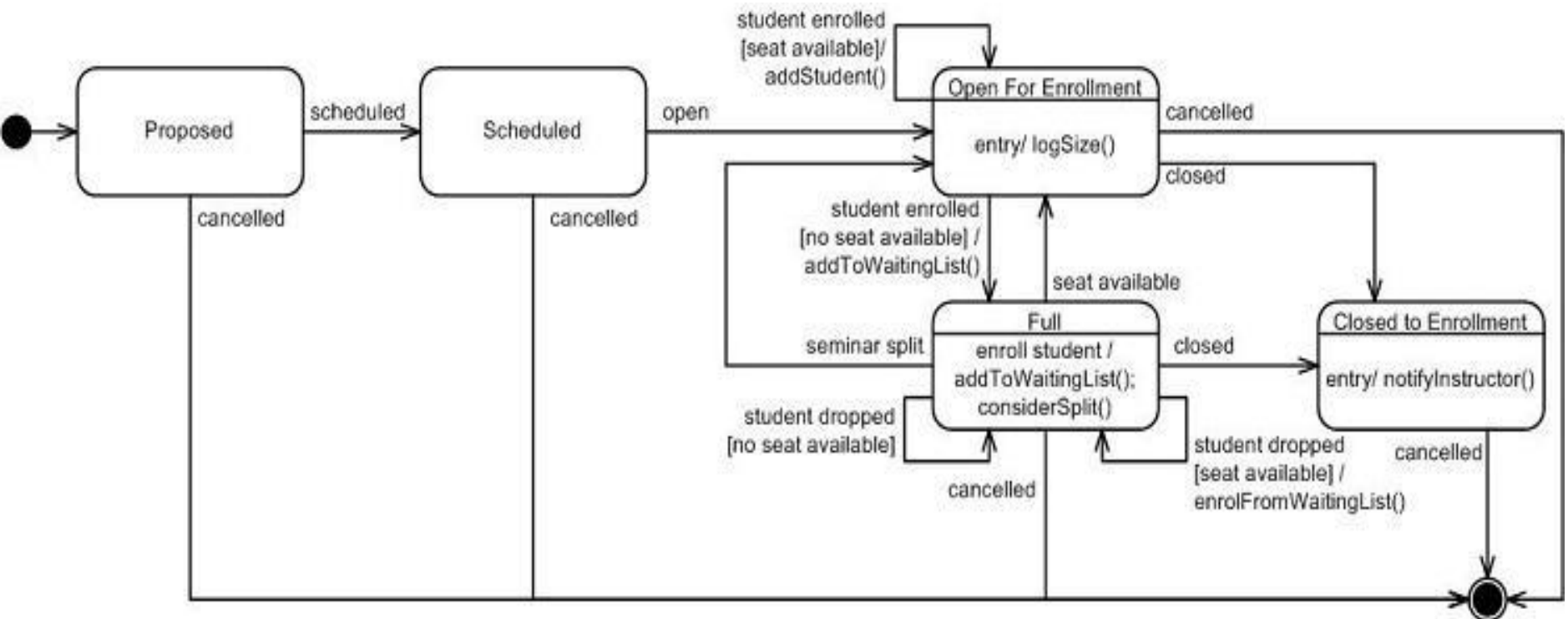
sm Transition

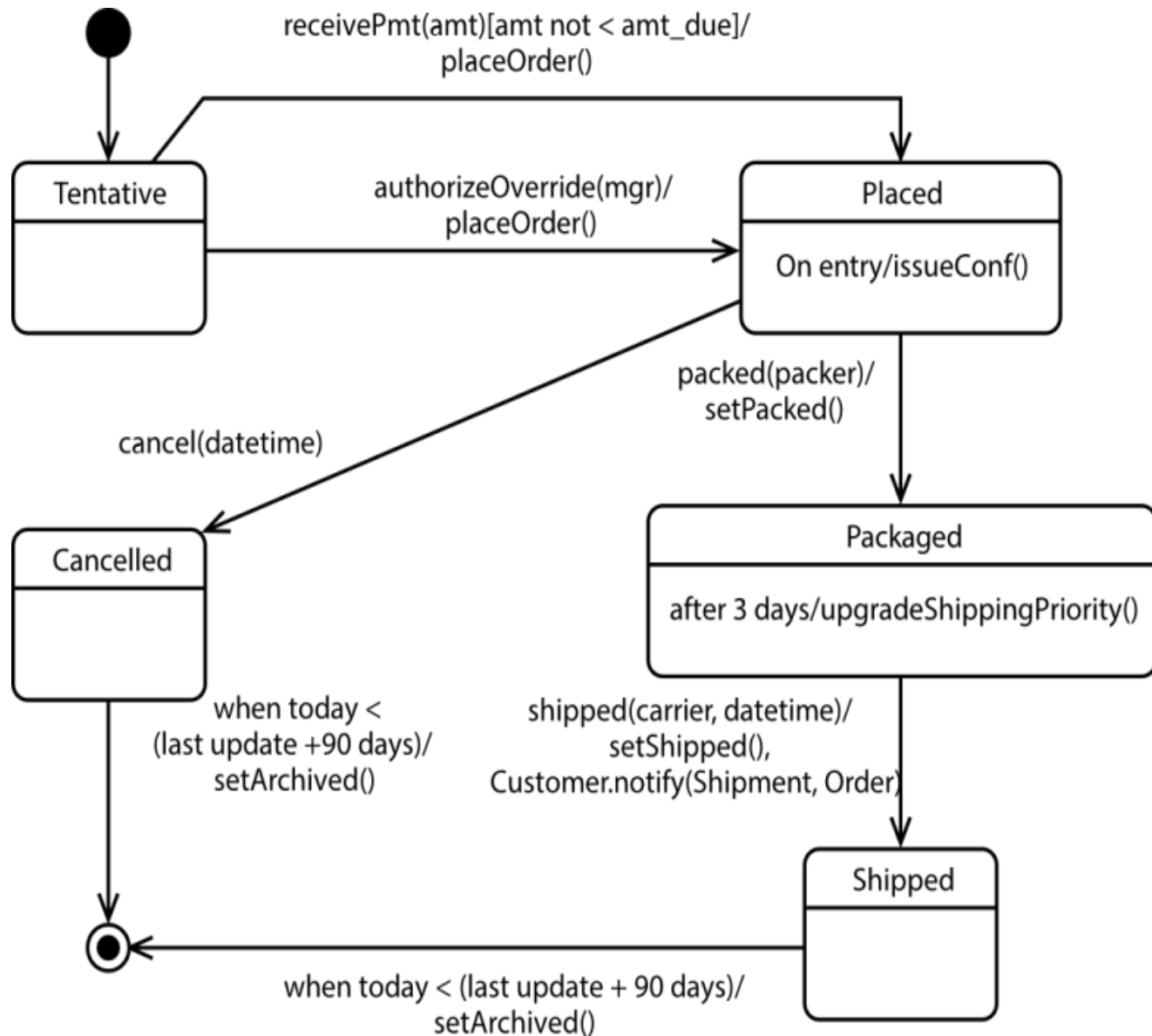


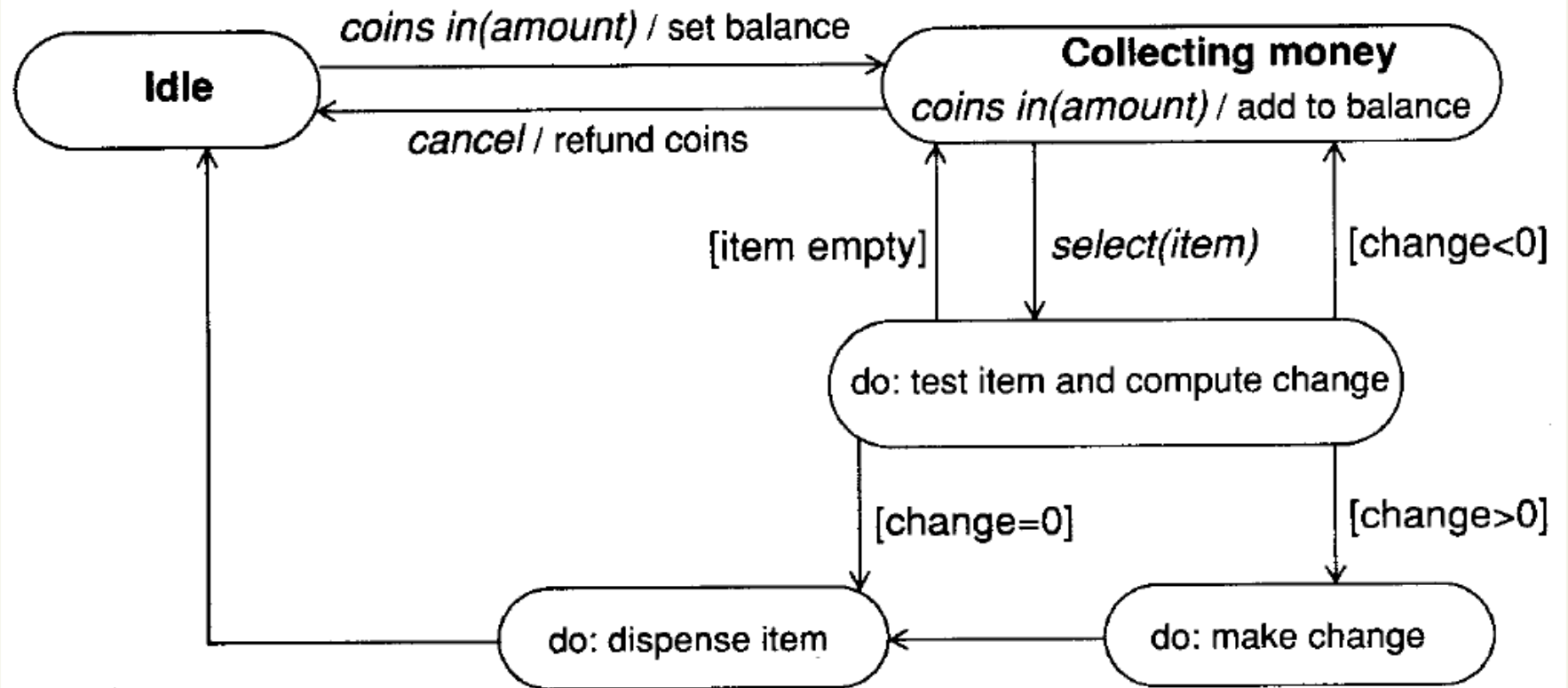
sm Self Transition

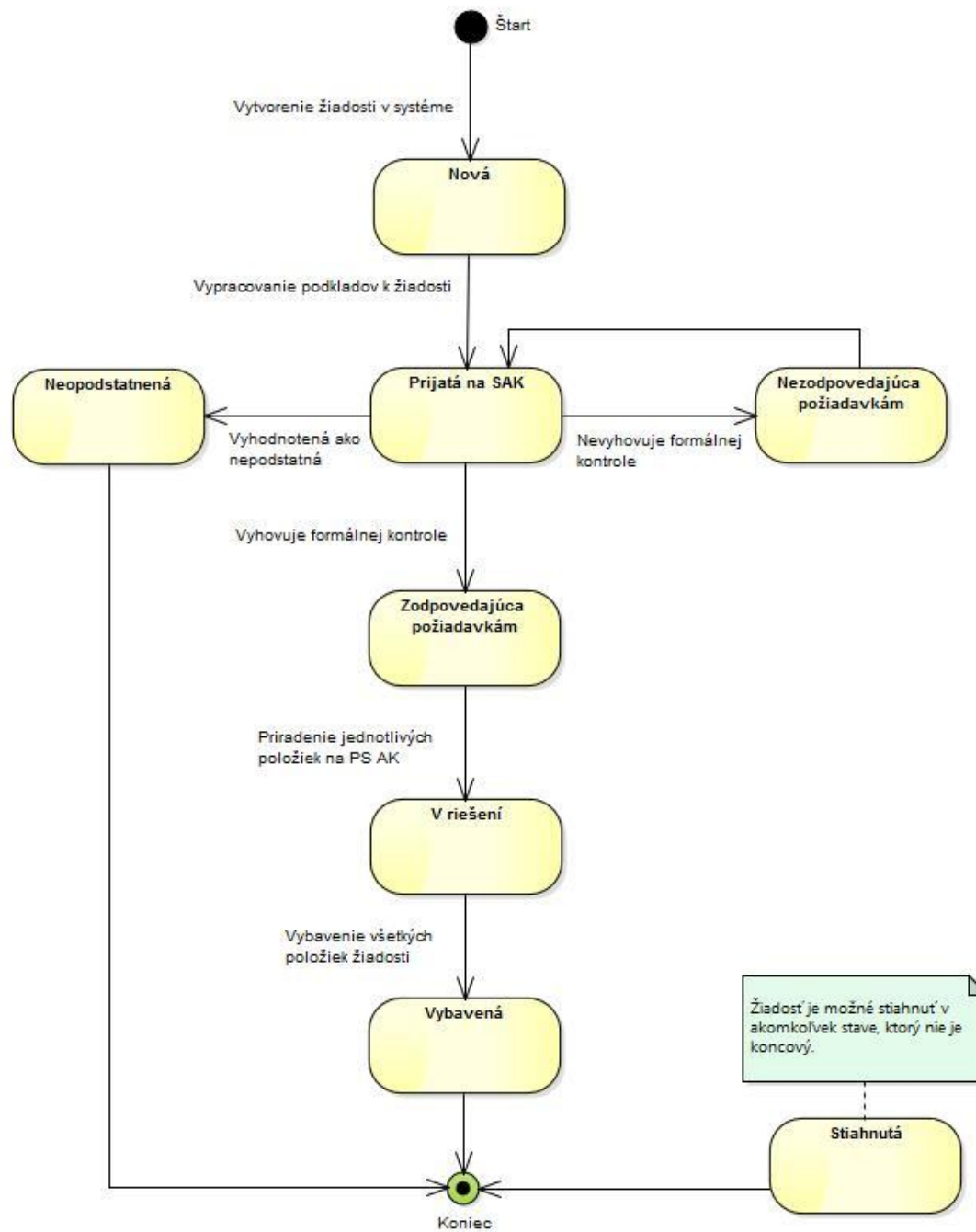


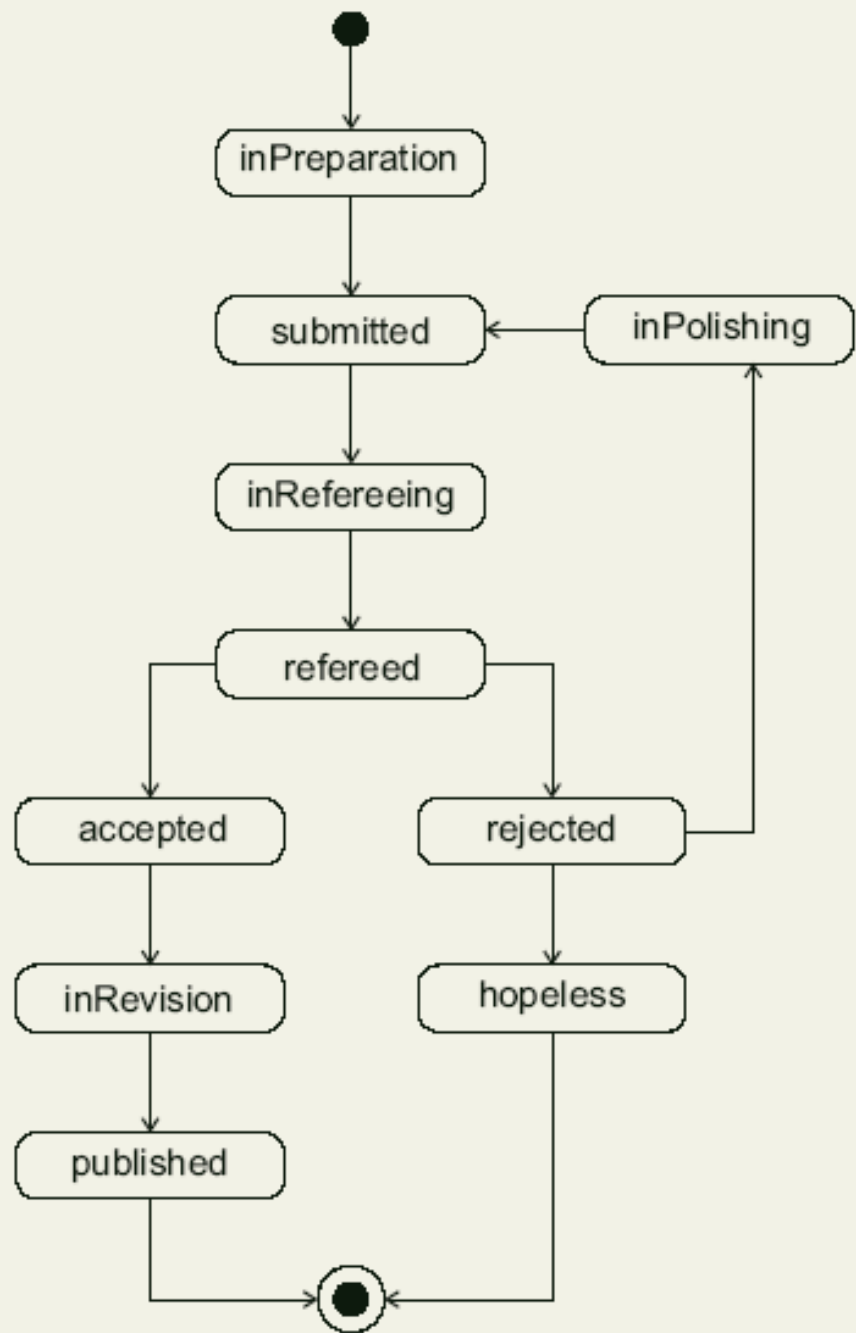


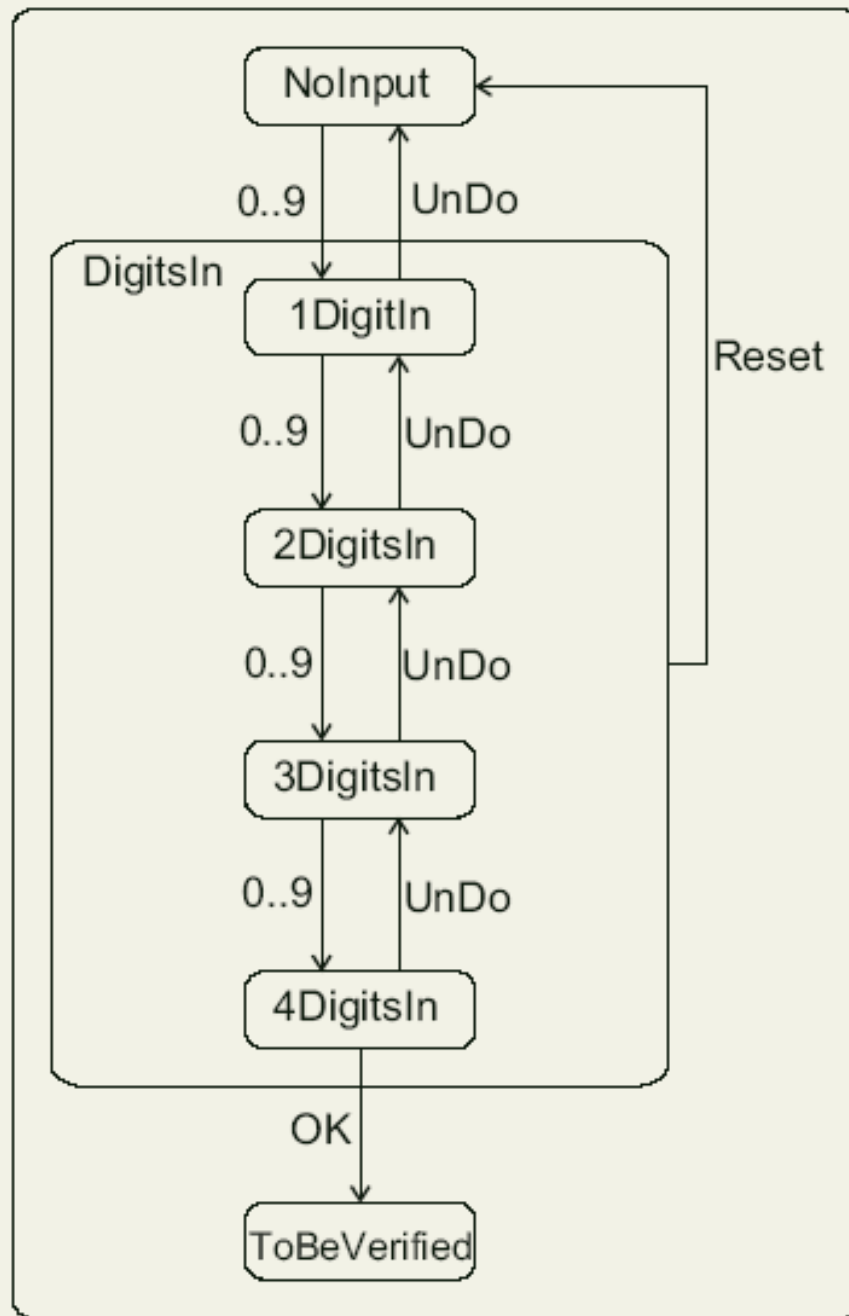
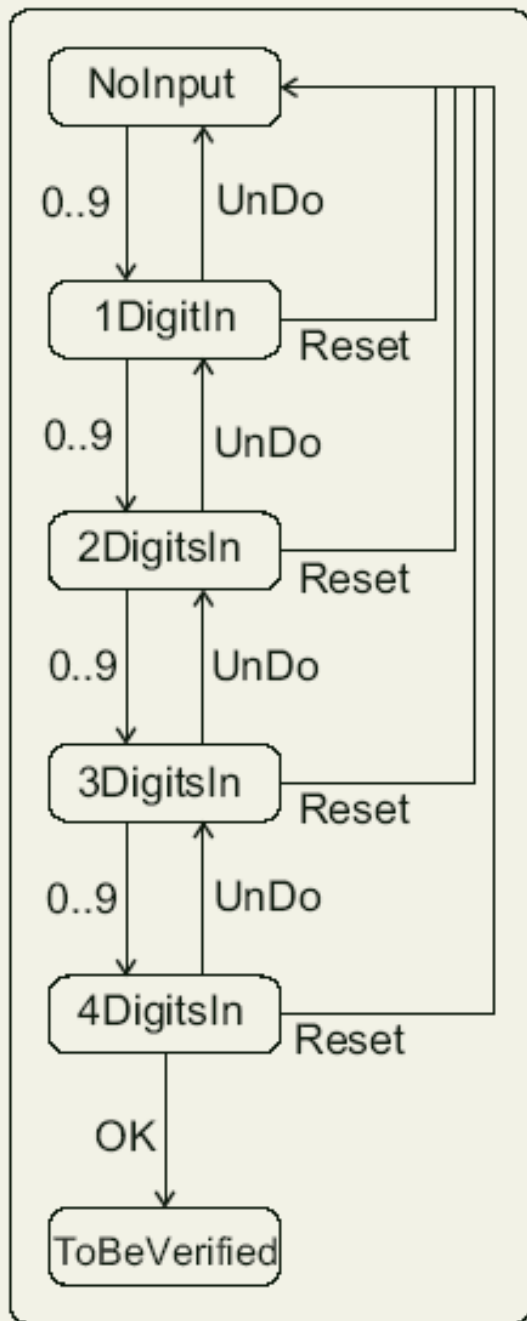


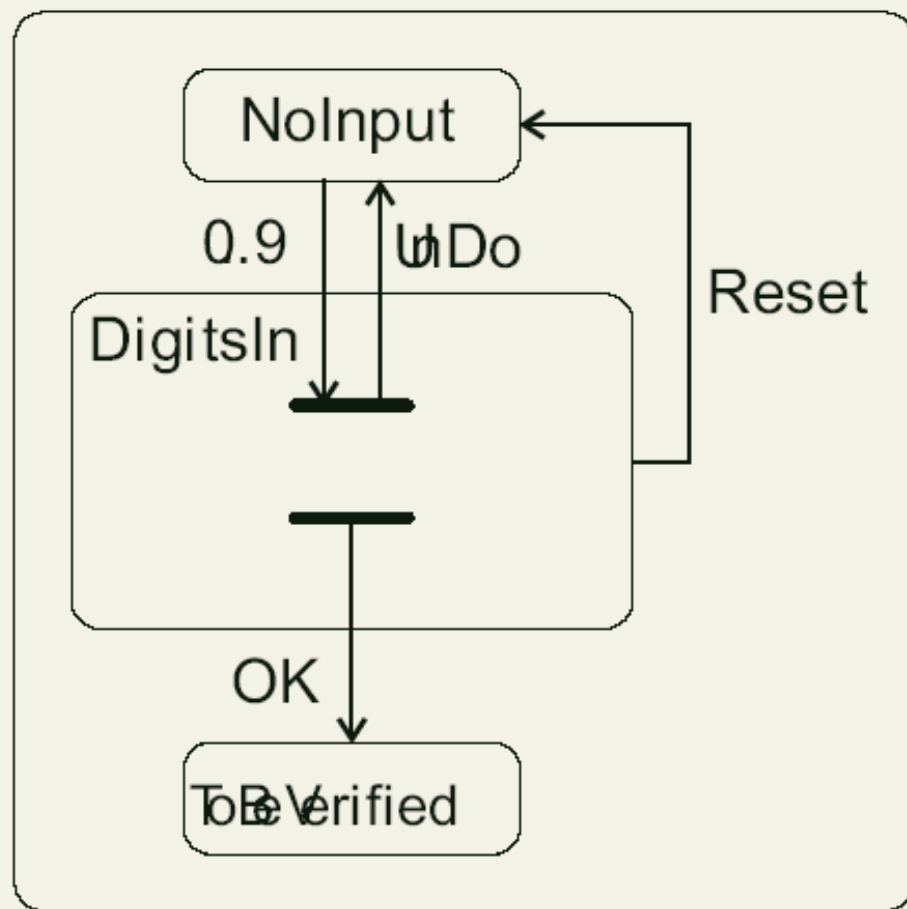


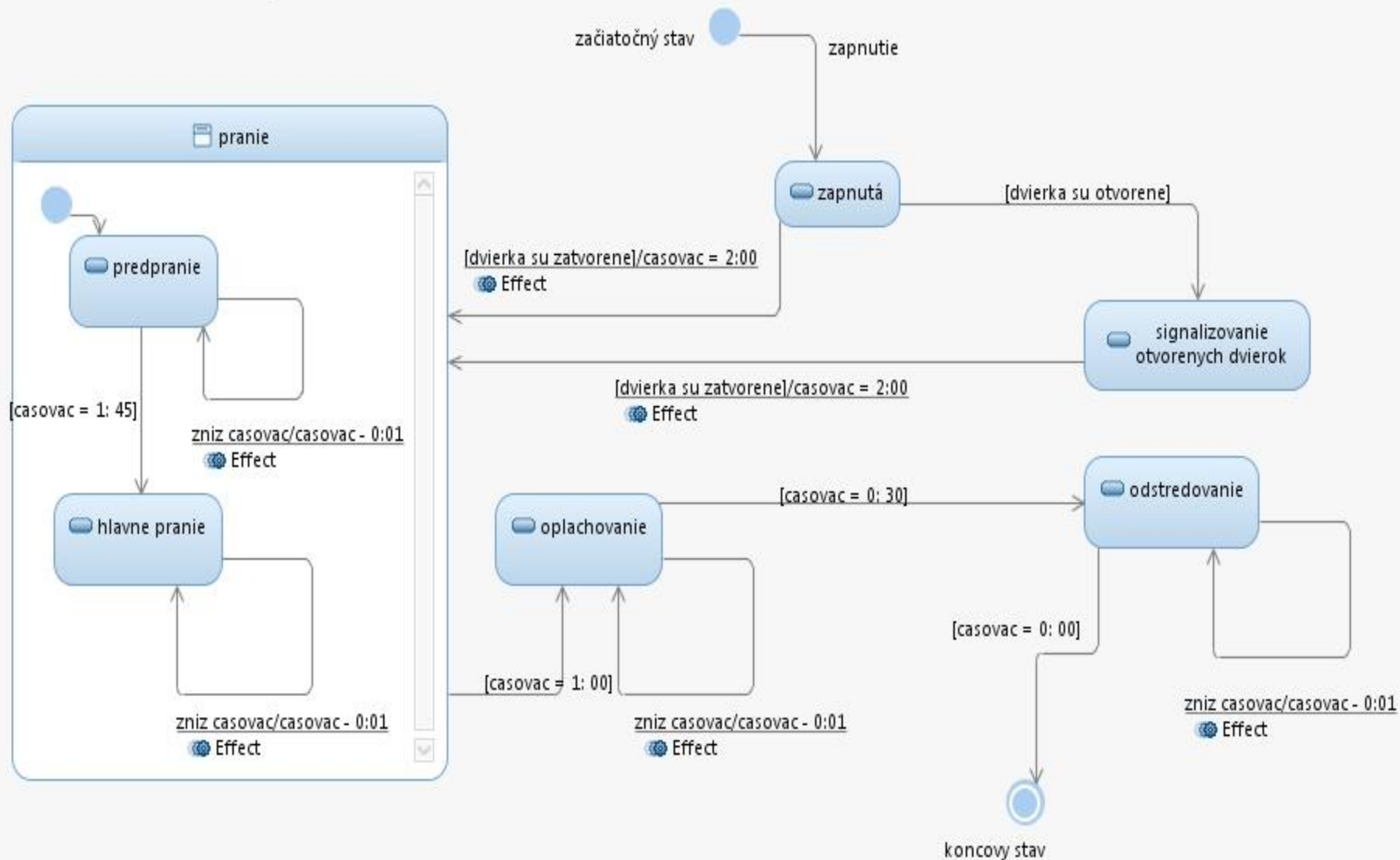












☑ Zapnutie mobilu a prihlasenie do siete - Autor: Frederik Autner

