

# Cvičenie z PTS

23.2.2010

# Požiadavky na systém

- Požiadavka môže byť opísanie funkcionality systému na vysokej úrovni, ale aj podrobne (matematicky) popísaná špecifikácia tejto požiadavky
- Dva typy:
  - Používateľské požiadavky
  - Systémové požiadavky

# Požiadavky na systém

## User requirement definition

1. The software must provide a means of representing and accessing external files created by other tools.

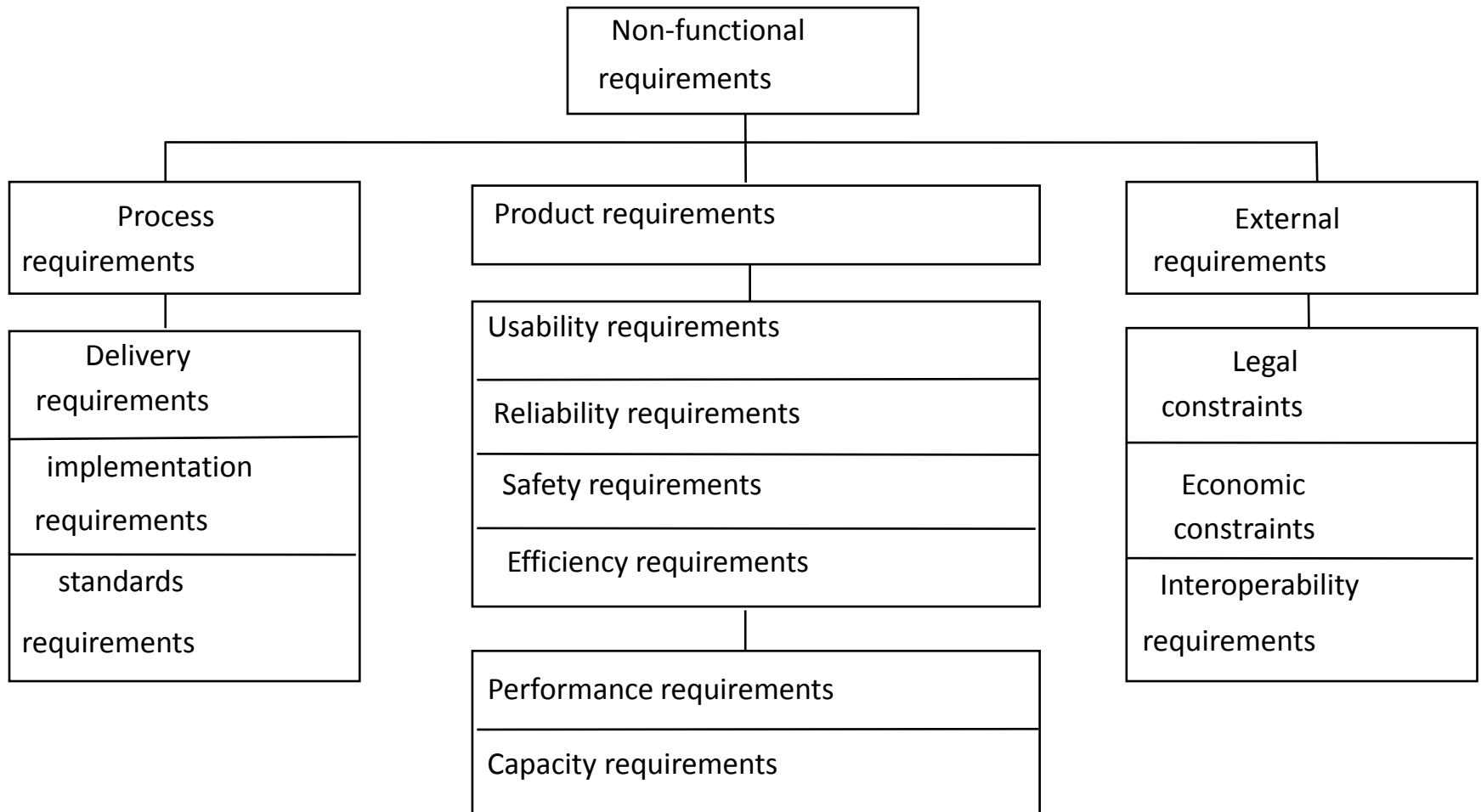
## System requirements specification

- 1.1 The user should be provided with facilities to define the type of external files.
- 1.2 Each external file type may have an associated tool which may be applied to the file.
- 1.3 Each external file type may be represented as a specific icon on the user's display.
- 1.4 Facilities should be provided for the icon representing an external file type to be defined by the user.
- 1.5 When a user selects an icon representing an external file, the effect of that selection is to apply the tool associated with the type of the external file to the file represented by the selected icon.

# Požiadavky na systém

- Funkčné požiadavky (functional)
  - Týkajú sa funkcionality systému, t.j. ako sa má systém správať, ako má reagovať na vstupy
- Netýkajúce sa funkčnosti (non-functional)
  - Rôzne obmedzenia na systém, ako rýchlosť odozvy systému, použitie štandardov, procesu vývoja systému a pod.
- Doménové (domain)
  - Vychádzajú z oblasti, do ktorej je systém navrhnutý.

# Požiadavky na systém



# Problémy s požiadavkami

- Požiadavky musia byť napísané zrozumiteľne a jednoznačne
  - Systém by mal poskytovať vhodný prehliadač na čítanie dokumentov v databáze.
  - Problém: Čo znamená „vhodný prehliadač“
    - Používateľ si predstaví pre každý typ dokumentu iný prehliadač, ktorý najvhodnejšie zobrazí jeho obsah
    - Vývojár to naprogramuje ako jednoduchý textový editor na prezeranie obsahu dokumentov
- Je dôležité vyhýbať sa vágnym, neoveriteľným požiadavkám:
  - „systém by mal byť spoľahlivý“, „systém by mal byť user friendly“

# Problémy s požiadavkami

- Požiadavky by mali byť
  - Kompletné, t.j. popisovať všetko čo má systém robiť
  - Konzistentné, t.j. navzájom si neodporovať
- V praxi je ťažké vytvoriť kompletný a konzistentný dokument s požiadavkami
  - Veľa požiadaviek si odporuje, napr. požiadavka na odozvu systému môže byť v konflikte s požiadavkou na vytvorenie prehľadu s veľkým množstvom dát.
- Je **nákladné** robiť zmeny v požiadavkách po tom, ako sa na nich dohodlo

# Dokument s požiadavkami

- Popisuje:
  - Služby a funkcie, ktoré by mal systém poskytovať
  - Obmedzenia, pod ktorými by mal systém pracovať
  - Ostatné systémy, s ktorými by mal systém komunikovať
  - Informácie o aplikačnej doméne systému, t.j. ako bude systém spracovávať a vykonávať požadované úlohy
  - Obmedzenia na proces vývoja systému
  - Popis hardvéru, na ktorom bude systém bežať
- Okrem iného by mal každý dokument s požiadavkami obsahovať úvodnú kapitolu, ktorá dáva náhľad na systém, popisuje “business need” za systémom a taktiež obsahuje slovník pojmov popisujúci používanú terminológiu



# Čitatelia dokumentu s požiadavkami

- Zákazníci
  - Špecifikujú požiadavky a čítajú dokument, aby overili, či bude systém robiť to, čo chceli
- Projektový manažéry
  - Používajú dokument na naplánovanie zdrojov potrebných na vývoj systému
- Systémový inžiniery
  - Používajú dokument na to, aby pochopili, čo má systém robiť
- Testovači
  - Používajú dokument na to, aby overili, či systém spĺňa požiadavky
- Správcovia systému
  - Používajú dokument na pochopenie systému

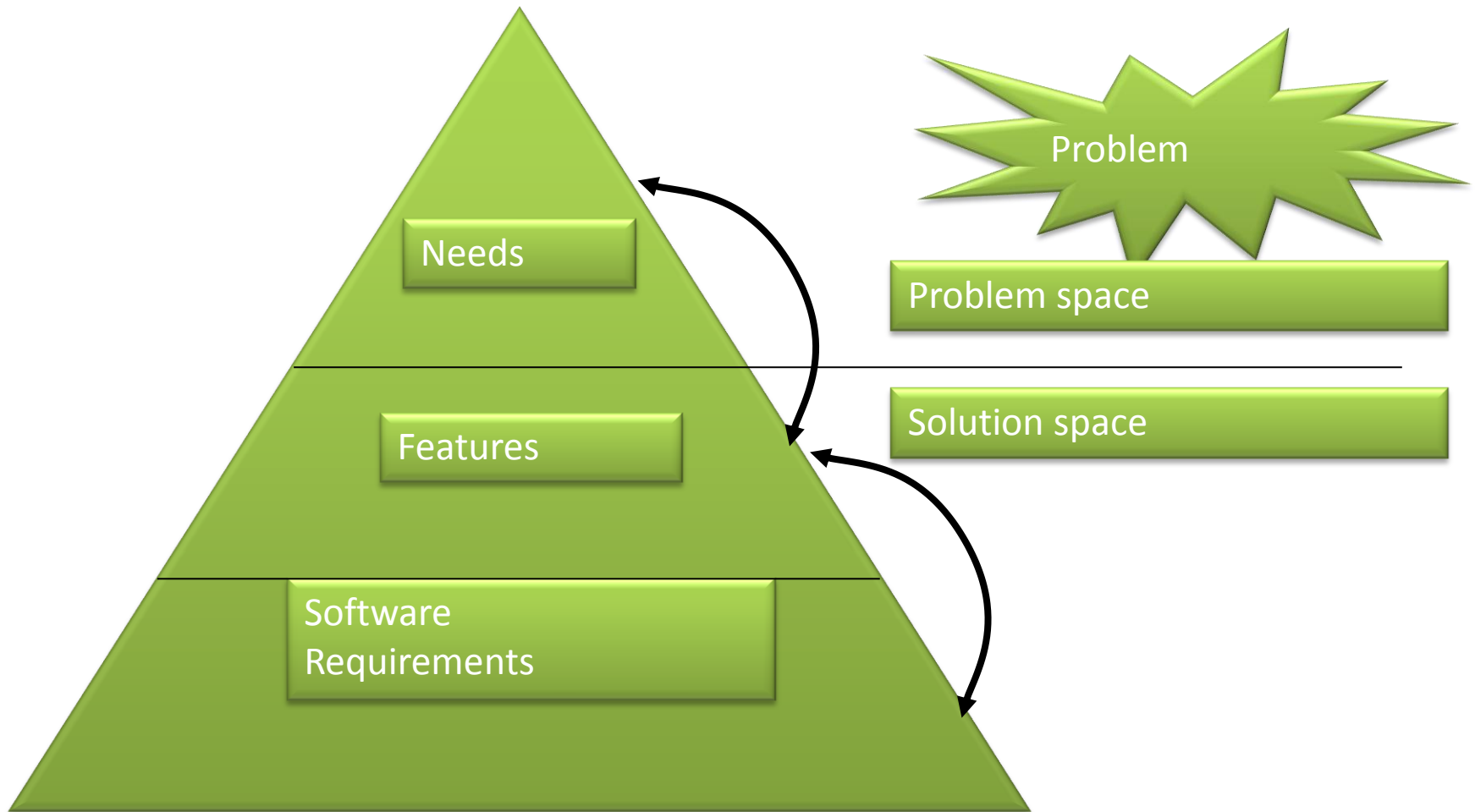
# Štandard dokumentu s požiadavkami

- IEEE/ANSI 830-1993 – štandard popisujúci štruktúru dokumentu s požiadavkami na systém
- Introduction
  - 1.1 Purpose of requirements document
  - 1.2 Scope of the product
  - 1.3 Definitions, acronyms and abbreviations
  - 1.4 References
  - 1.5 Overview of the remainder of the document

# Štandard dokumentu s požiadavkami

- 2. General description
  - 2.1 Product perspective
  - 2.2 Product functions
  - 2.3 User characteristics
  - 2.4 General constraints
  - 2.5 Assumptions and dependencies
- 3. Specific requirements
  - Covering functional, non-functional and interface requirements.
- 4. Appendices
- Index

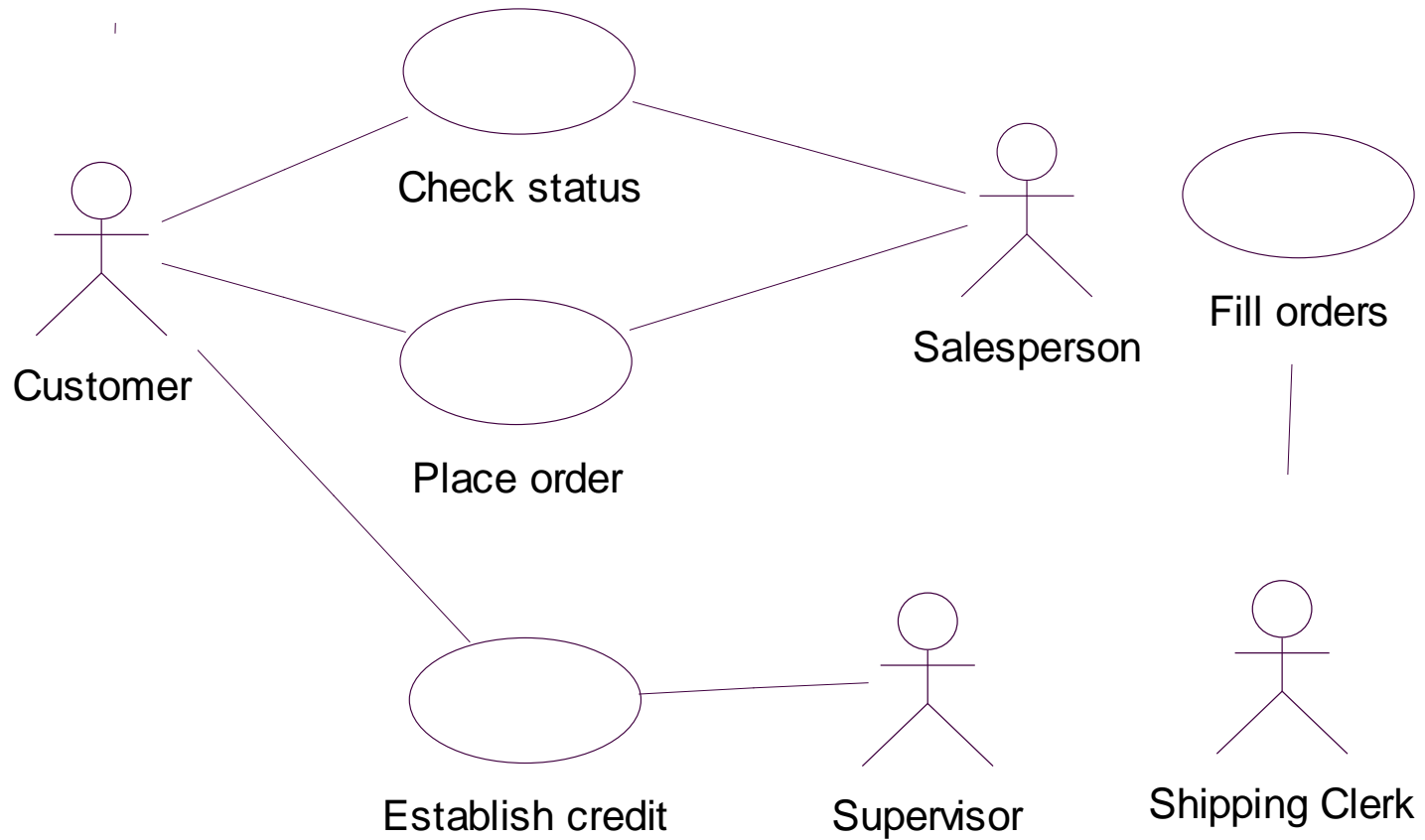
# Požiadavky na systém



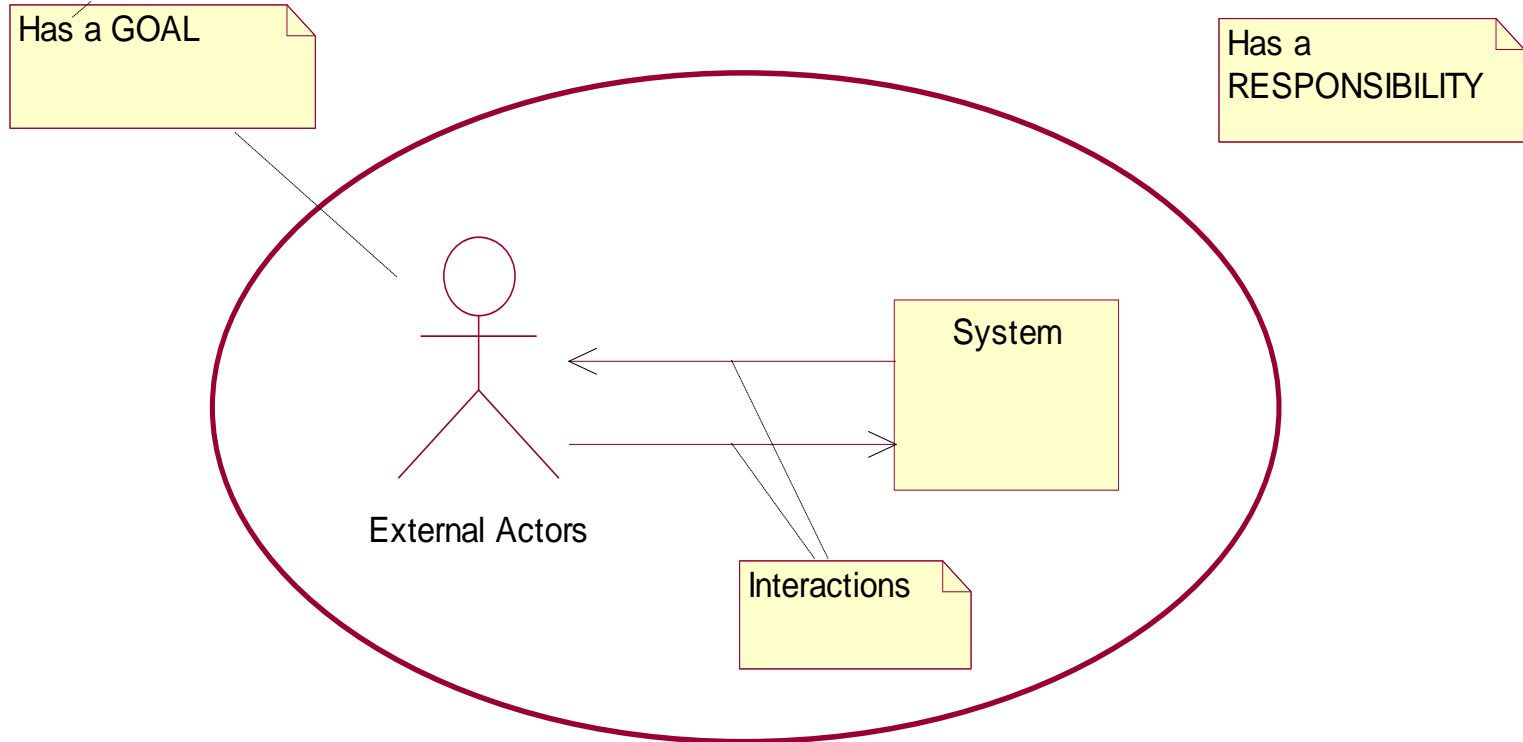
# Use Case

- Zobrazuje vzťahy medzi „používateľmi“ (actors) a rôznymi „službami“ ktoré systém poskytuje (use cases)
- Use case model zobrazuje funkcionality systému alebo triedy ako interaguje s externými zdrojmi (ľuďmi, systémami)
- Zobrazuje „obrysy“ systému

# Use Case



# Use Case



Use Case

# Use Case

- Zobrazuje úlohu jednotlivých „účastníkov“ v systéme
- UML definícia: a coherent unit of functionality provided by a system as sequences of messages among the system and one or more outside interactors (called *actors*) together with actions performed by the system
- Množina všetkých „use caseov“ popisuje celú funkcionálnosť systému
- Popisuje **ČO** má systém vykonávať, **nie AKO**.

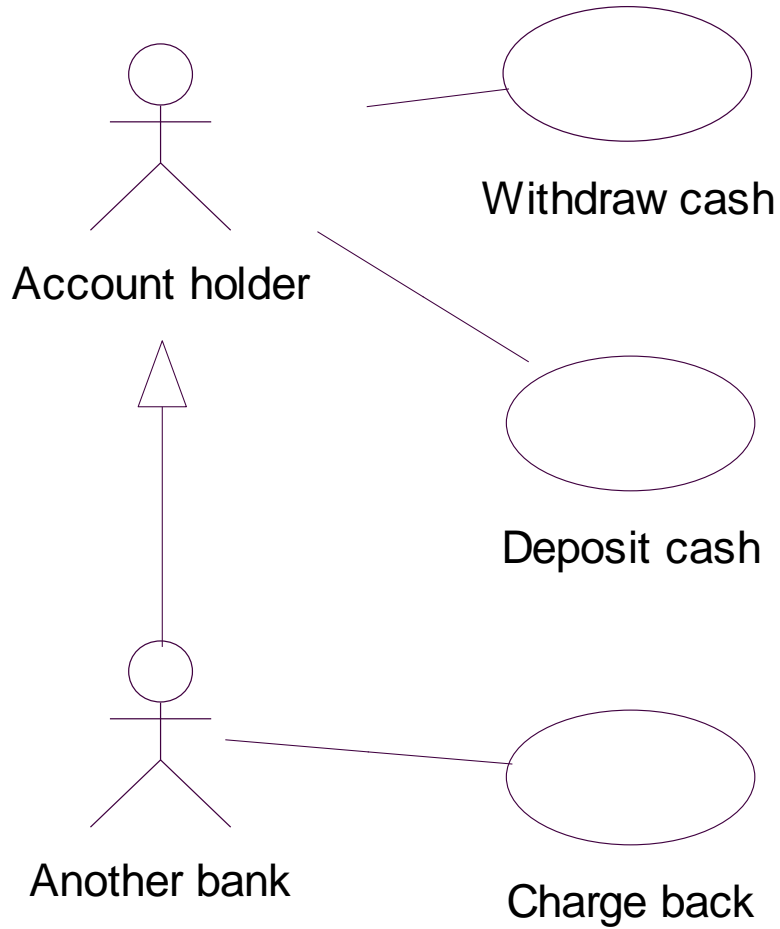


# Use Case

## Účastník:

- Rola objektu (objektov) MIMO systému, ktorá interaguje priamo so systémom
- Jeden (fyzický) objekt môže mať viacero rolí, čo modelujeme viacerými účastníkmi.
- Dva typy účastníkov:
  - Primárni – sú iniciatormi (spúšťajú) use case
  - Sekundárni (väčšinou stroje) – sú volaní systémov v rámci use case
- Dedenie účastníkov:
  - Špecializovaný účastník dedí vlastnosti otca

# Use Case



# Use Case

- **Cieľ**
  - by mal byť centrálnym pojmom počas use case analýzy
  - väčšina ľudí (aj nie technickí) myslí, aké ciele chce dosiahnuť počas práce
  - hlavnou úlohou softvéru je poskytovať podporu pre ich prácu používateľov
  - hovorte jazykom používateľov:
    - v jazyku CIEĽOV a ZODPOVEDNOSTI

# Use Case

- Name
- Goal
- Preconditions
- Postconditions (success and failed)
- Normal flow of events (main scenario)
- Alternative flows of events (alternative scenarios)
- Additional properties
  - priority, schedule, time, frequency
  - other non-functional requirements
- Open issues

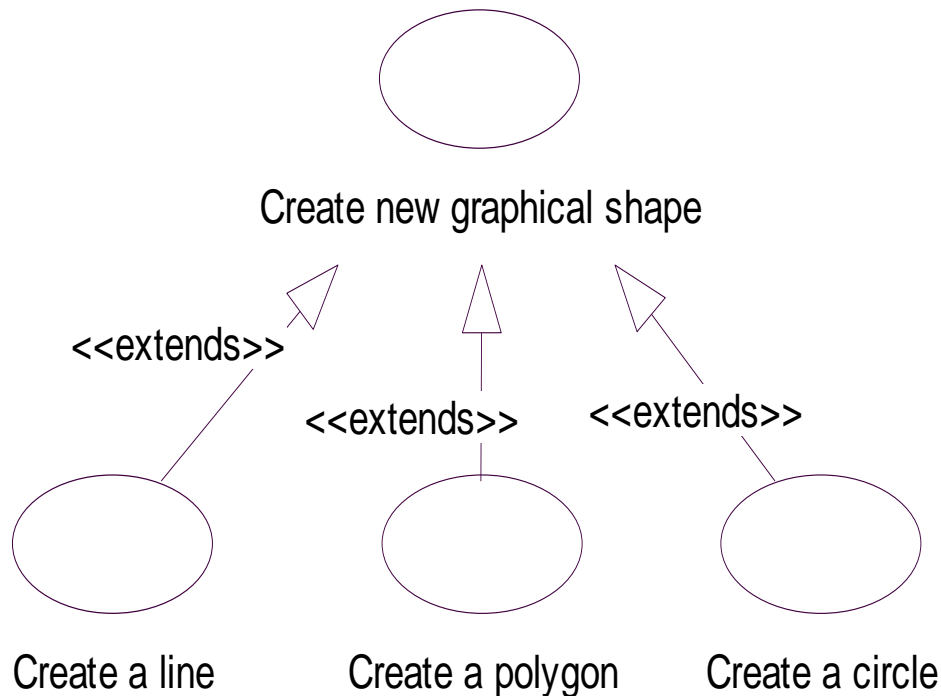
# Use Case

## Vzťahy:

- Komunikácia
  - Podiel účastníka v use case
  - Jediný vzťah medzi účastníkmi a use casemi
- Rozšírenie a využitie (Extend & Uses)
  - Používajú sa na štrukturizovanie a zjednodušenie funkčných požiadaviek

# Use Case

## Extends:



- indicates that the instance of extended (base) use case MAY include the behavior specified by the extending use case
- a single use case may have several extenders
- extension often represents unusual behavior (exception, error)
- an *extension point* is a location within a use case at which action sequences from other use cases may be inserted.
- the base use case may be unaware of the extension

# Use Case

**Uses** - indicates that an instance of the use case will also include the behavior as specified by the included use case(s)

