

Zpracoval: houzvjir@fel.cvut.cz

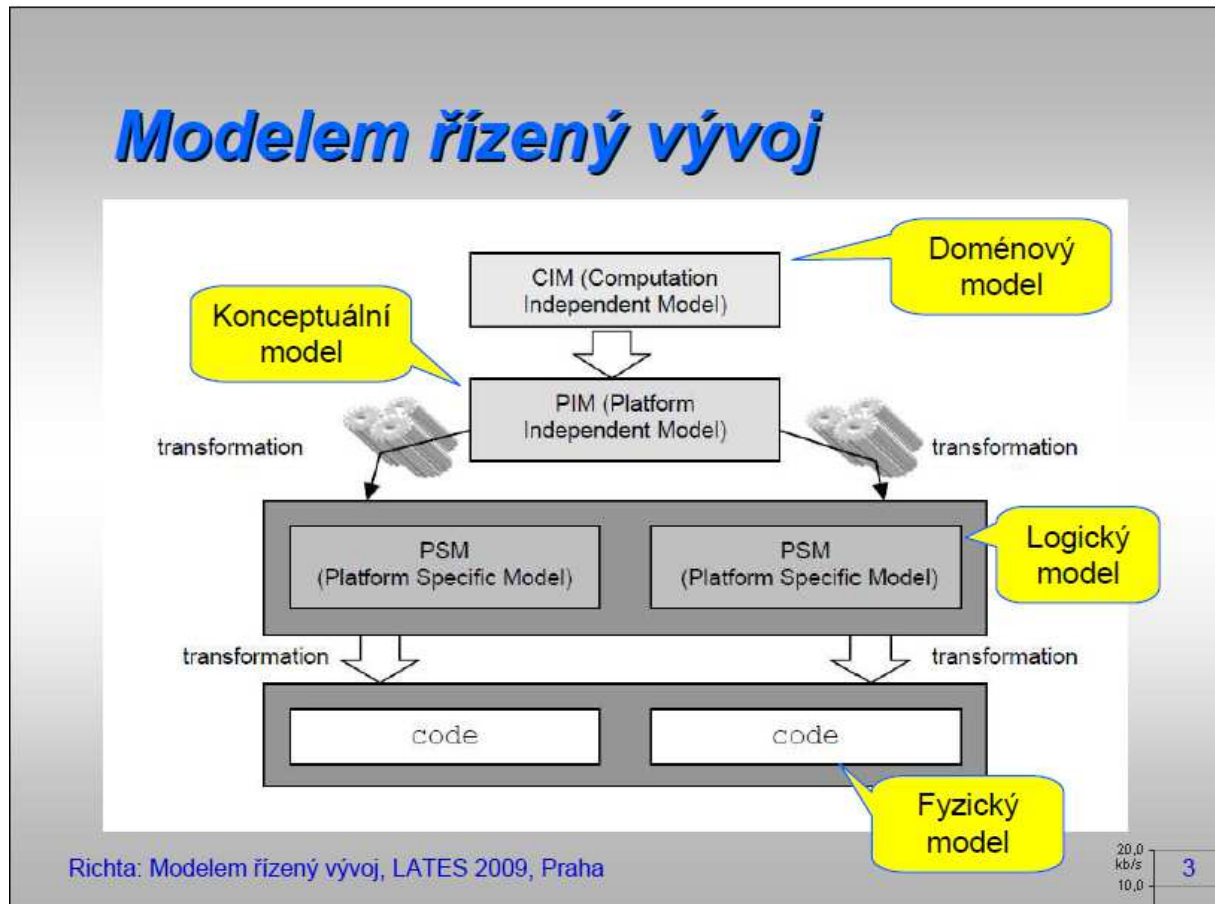
03. Modelem řízený vývoj. Doménový (business), konceptuální (analytický) a logický (návrhový) model. Vize projektu. (A7B36SIN)

Obsah

Modelem řízený vývoj	2
Cíl MDD, proč vlastně existuje.....	2
Průběh vývoje využívající MDD	2
Doménový model (CIM)	3
Konceptuální analytický model (PIM).....	7
Logický návrhový model (PSM)	9
Vize projektu	9

Modelem řízený vývoj

(Model Driven Development – MDD)



Cíl MDD, proč vlastně existuje

V současnosti jsou velkým problémem vysoké náklady spojené s údržbou softwarových aplikací a jejich malou flexibilitou vzhledem k požadovaným změnám. Tyto problémy pomáhá MDD řešit. MDD je koncept standardizující modely, které jsou v průběhu vývoje aplikace vytvářeny a definující způsoby mapování mezi nimi. MDD není převratnou novinkou, nicméně podstatné je, že toto členění standardizuje a dále definuje způsob transformace jednoho modelu v druhý.

Průběh vývoje využívající MDD

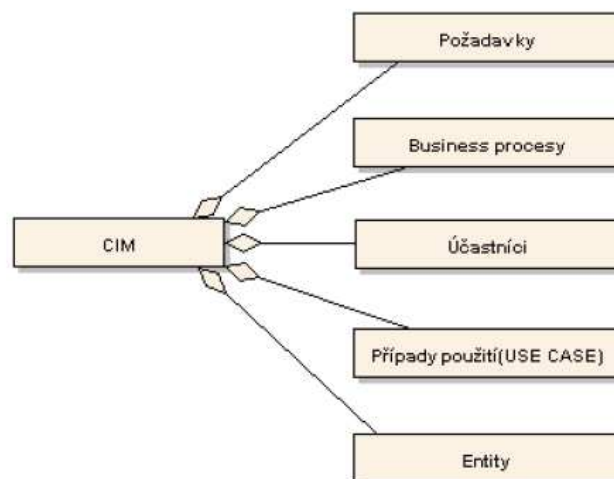
Základním prvkem celého projektu založeného na MDD jsou modely. MDD vidí modely ve čtyřech úrovních:

- CIM (Computer Independent Model) – model nezávislý na počítačovém zpracování
- PIM (Platform Independent Model) – platformově nezávislý model řešení
- PSM (Platform Specific Model) – platformově specifický model řešení
- Code – kód aplikace, tedy výsledná realizace řešení

Podstatné je, že samotný kód vyvíjené aplikace je jen jednou částí celého projektu. Hlavní práce je na modelech, ze kterých se nakonec vygeneruje velká část kódu (kostra celé aplikace) a teprve v poslední řadě se implementuje konkrétní funkčnost aplikace (kód). Modely přitom slouží i jako podstatná část dokumentace celého projektu a při změnách je nutné udržovat aktuální všechny modely i kód, nestačí jen upravit kód – také proto, že modely jsou i dokumentací, ale především proto, že právě modely definují funkčnost aplikace a definují všechny vztahy mezi jejími částmi.

Doménový model (CIM)

- Zaměřuje se výhradně na prostředí a obecné požadavky systému. Detailní struktura a konkrétní zpracování jsou v této fázi skryté nebo dosud neurčené.
- Reflektuje „business“ požadavky zákazníka a pomáhá přesně popsat to, co se od systému očekává.
- Musí být nezávislý na technickém zpracování a popisovat systém čistě věcně a logicky. Důležitý především proto, aby vývojáři pochopili činnosti uživatele a chápali příslušné souvislosti.
- Uživatel těchto modelů je člověk, který není obeznámen s modely nebo konstrukcemi užívaných k vyjádření funkčnosti těch požadavků, které jsou právě specifikované v CIM. Tedy člověk, který plně nerozumí modelu PIM a nižším.
- Nejčastějšími uživateli tohoto typu modelů jsou „business“ analytici případně sami uživatelé systému.
- Slouží jako prostředek k pochopení problémů a vymezení problémové oblasti, také jako sdílený zdroj pojmů, které se používají v jiných modelech (úrovních MDD) nebo pro další modelování na stejné úrovni (procesní modely, Use case diagramy případně diagramy činností).
- Slouží jako strategický materiál pro odhad ceny projektu.
- Slouží jako strategický materiál pro vypracování nabídky.
- Podklad pro uzavření smlouvy.



Požadavky:

- Zachycují očekávání zákazníka
- Mají jasné identifikátory (čísly se)
- Hierarchicky se dělí na funkční a nefunkční. Funkční požadavky definují co bude systém umožňovat, nefunkční požadavky jsou obecnější – většinou se realizují vhodnou volbou jádra systému.
- Jeden požadavek = jedno měřitelné očekávání
- Obsahují pokud možno minimální množství informací o implementaci

Business procesy slouží pro potřeby tvorby nabídky. Model business procesů slouží pro odsouhlasení rozsahu aplikace klientem. Také slouží pro testery a pro školitele.

Účastníci jsou pro systém externí entitou, která systém využívá nebo ho ovlivňuje. Většinou je účastníkem osoba, ale může jím být např. i čas nebo dokonce blesk, nebo koncový spínač apod.

Případ užití by měl popisovat jednu rutinní akci jednoho účastníka v jednu chvíli. Je vždy iniciován účastníkem. USE CASE diagram znázorňuje funkce systému z pohledu účastníků.

Entity jsou většinou reálně existující „věci“, se kterými systém bude manipulovat. Seznam entit a jejich popis je slovníkem pojmů. Ze seznamu entit se vychází při tvorbě analytických tříd v PIM.

Příklad:

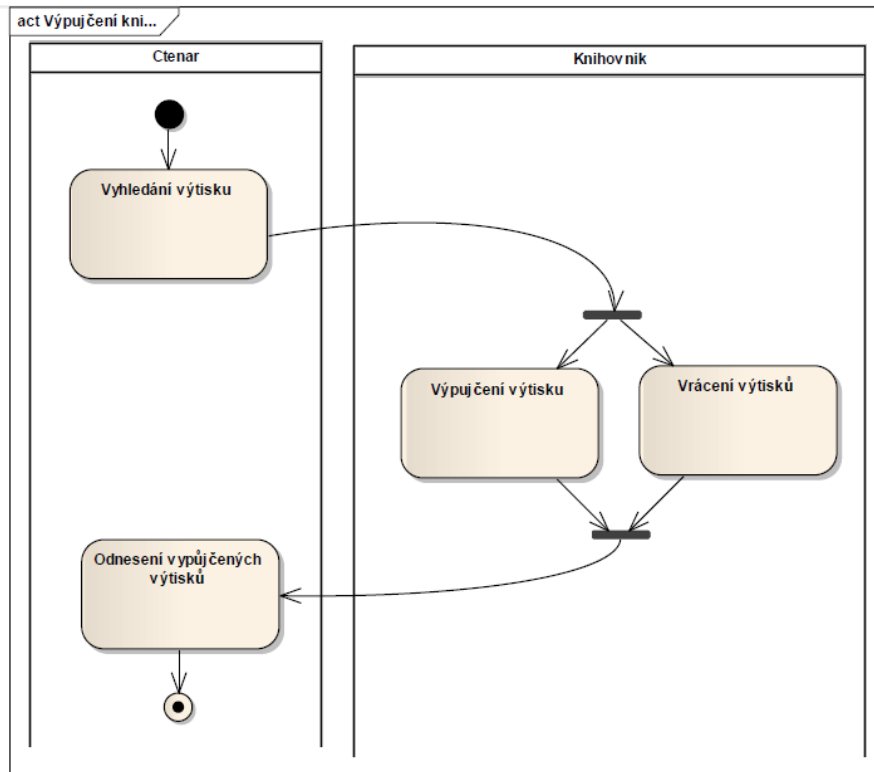
Business process model

Procesy v knihovně:

Kapitola obsahuje popis procesů, které musejí být v knihovně vykonávány při jejím běžném provozu. Jedná se o půjčování a vrácení knih, správu výtisků, které knihovna vlastní a správu čtenářů, kteří si mohou knihy půjčovat.

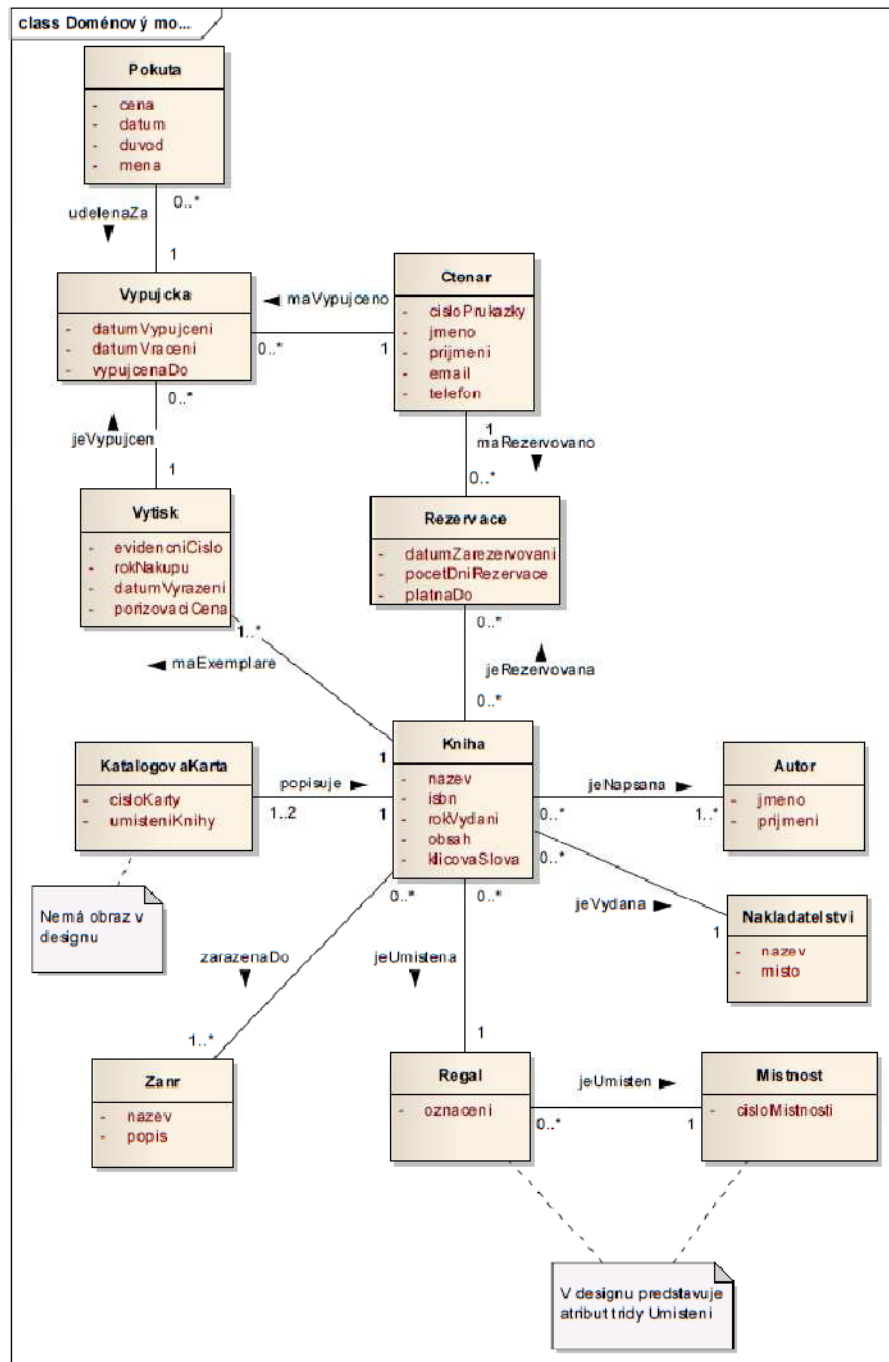
1. Vypůjčení knihy

Vypůjčení knihy je jeden z hlavních procesů, které jsou v knihovně vykonávány a který umožňuje čtenáři vyhledání požadovaného výtisku, jeho vypůjčení a následně také vrácení.



Obrázek 1 - Vypůjčení knihy

Doménový model



Konceptuální analytický model (PIM)

- Reprezentuje koncepci řešení dané problémové oblasti na základě konkrétních požadavků.
- Oproti předcházejícímu modelu je doplněn o ty informace (algoritmy, principy, pravidla, omezení...), které jsou nezbytně důležité k řešení dané problémové oblasti prostřednictvím informačních technologií.
- Přesně nevychází z CIM, nýbrž si z něho obvykle IT analytik vybere pouze to podstatné, co se považuje za smysluplné pro potřeby počítačového zpracování konkrétní problémové oblasti.
- Umožňuje vyřešit určitou oblast na obecné úrovni, a teprve pak zvažovat konkrétní technologii pro vlastní realizaci.
- Je mnohem jednodušší než model, který již specifické technologické informace obsahuje (PSM), a tudíž se v tomto modelu snadno zorientuje analytik, který obvykle nemá (ani to není žádoucí) detailní technologické znalosti.

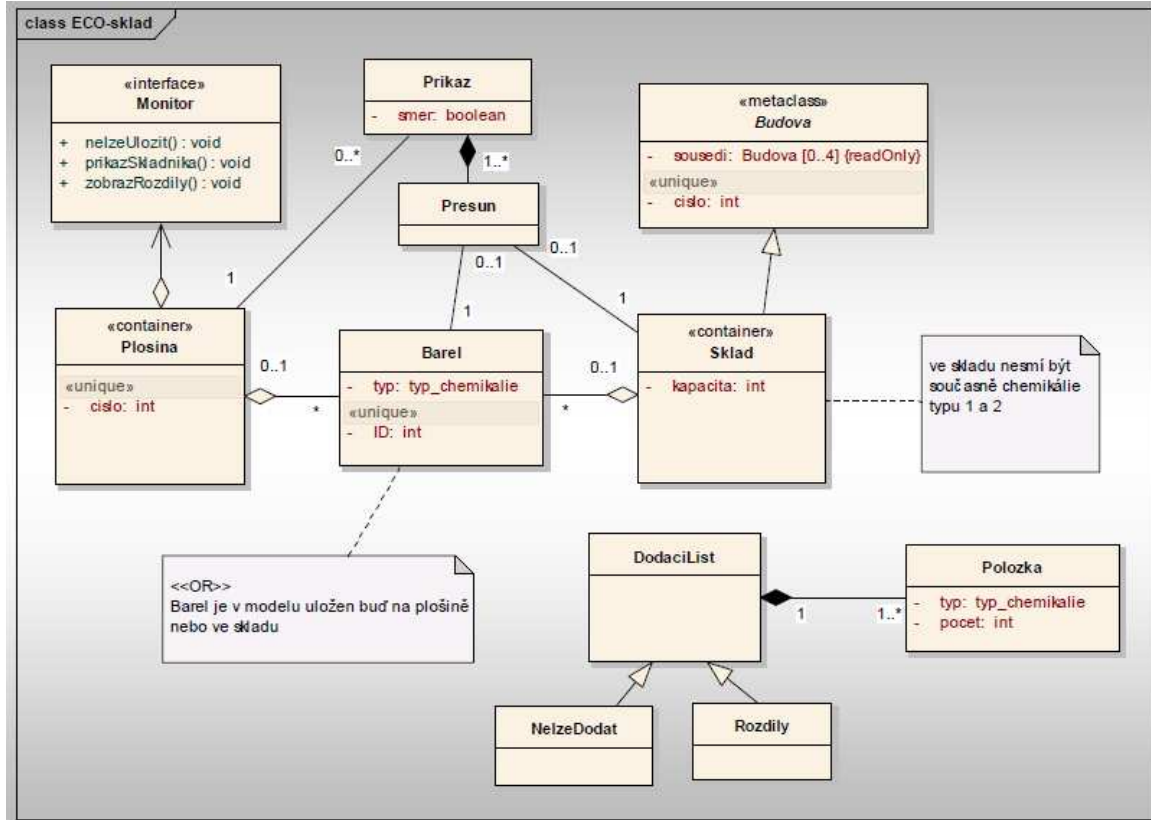
Lze rozdělit na 3 části:

- Konceptuální funkční model
- Konceptuální datový model
- Konceptuální dynamický model

Funkční model zachycuje služby, které má systém poskytovat. Tyto služby již byly popsány v úvodní studii, zde je třeba je definovat přesně. Vyjadřovacím prostředkem je model jednání obsahující seznam událostí, příp. scénáře. U složitějších aktivit je vhodné uvést diagramy aktivit.

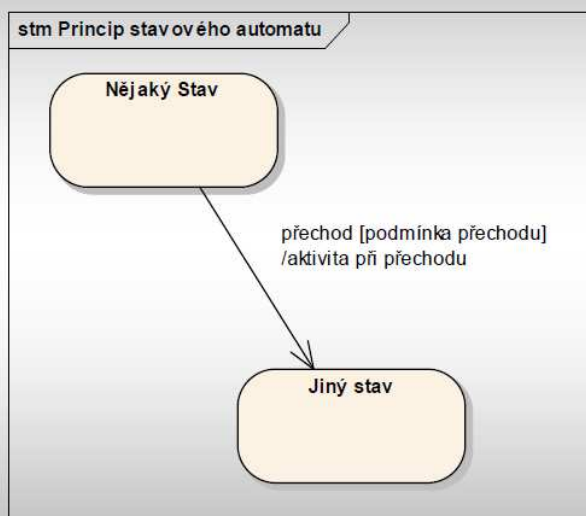
Datový model musí popsat práci s daty, aby bylo možné služby poskytovat. V úvodní studii bylo zhruba popsáno, s jakými daty budeme pracovat, zde je třeba to definovat přesně. Jako vyjadřovací prostředek je vhodné použít konceptuální model tříd nebo ER model (diagramy + textový popis).

Příklad datového modelu



V **dynamickém modelu** musí být přesně popsáno, jak se mění chování prvků systému a na základě jakých okolností tyto změny nastávají. Jako vyjadřovací prostředek je vhodné použít scénáře životních cyklů, stavové diagramy.

Princip stavového automatu



Životní cyklus skutečného „barelu“



Logický návrhový model (PSM)

- Model řešení na dané platformě (např. J2EE nebo C# a ASP.NET) je podkladem pro vlastní implementaci.
- Věrně odráží strukturu kódu a je již dostatečným podkladem pro implementaci. V podstatě se jedná o vizualizaci kódu na stejné úrovni abstrakce.
- Vyskytují se v něm objekty, které úzce souvisí se zvoleným technologickým prostředím (např. operace pro přístup k atributům, konstruktory a destruktory tříd,...). Tento model vytvářejí návrháři. Databázový model (model relační databáze) lze považovat také za PSM.

Vize projektu

Účelem tohoto dokumentu je předložit obecnou vizi nejzákladnějších požadavků, účelu, klíčových vlastností a hlavních omezení týkajících se softwarového projektu. Po jeho přečtení by mělo být čtenáři jasné o jaký projekt se jedná a co se od něj očekává.

Vize projektu má vzniknout na začátku při zahájení projektu. Je vstupem do úvodní studie.

Vize projektu má obsahovat:

- Určení účelu produktu - stručný popis produktu, smysl zadání. Tato část by měla řešit následující otázky:
 - Jaký problém by software měl řešit - v čem problém spočívá, jakých oblastí či procesů se dotýká?
 - Kam lze produkt zařadit na trhu?

- Do jaké kategorie produkt spadá?
- Existují nějaké konkurenční produkty?
- Jaké výhody oproti konkurenčním produktům toto řešení nabízí, proč by měl mít zadavatel zájem na jeho tvorbě?
- Charakteristiky zadavatelů - jedná se o seznam osob zainteresovaných na projektu včetně popisu jejich pravomocí a odpovědnosti.
- Charakteristiky uživatelů
- Omezení projektu
- Podpora určitých standardů (např. požadavek na komunikaci přes protokol TCP*IP).
- Kvalitativní kritéria - dostupnost, řešení nestandardních stavů aj.
- Požadavky na prostředí - podpora určitých platforem, uživatelské rozhraní.