

# PA 116 – Domácí úkol 3

---

## 1 Řešitelé

**David Gešvindr** (255653)

**Petr Kunčar** (389926)

**Radek Foltýn** (389962)

**Tomáš Kuba** (207778)

## 2 Zadání

Namodelujte konceptuální model komplexnější aplikace, nejlépe aplikace, na které se sami podílíte, nebo ji máte dobře odzkoušenou. Je třeba, aby měl model minimálně 15 entit.

Identifikujte v modelu slabá místa nebo omezení a vytvořte druhý konceptuální model, kde je opravíte. Aplikujte analytické vzory tam, kde je to vhodné a povede to ke zjednodušení modelu.

Pokud vaše aplikace skutečně slabiny na úrovni konceptuálního modelu nemá, namodelujte novou funkcionalitu, která vám v aplikaci schází a zakomponujte ji do celkového modelu (zkuste při tom použít analytické vzory).

## 3 První fáze

Naše práce modeluje Content Management systém (zkratkou CMS). CMS se v dnešní době velice intenzivně využívá pro správu informací souvisejících s provozem firmy nebo třeba pro správu obsahu vystaveného v rámci webové prezentace. Obecně lze CMS rozdělit na 2 hlavní skupiny – Enterprise CMS, což jsou systémy nasazené ve firemním prostředí, které jednotným způsobem zpravují informace potřebné pro chod společnosti, vyznačují se vysokou spolehlivostí a integrací s různými jinými firemními systémy. Další podstatnou skupinou jsou Web CMS, které se intenzivně využívají právě pro správu obsahu v rámci webových prezentací a bývají doplněny o uživatelské rozhraní, které zvládne používat téměř každý.

Námi navrhovaný CMS využívá zejména dědičnosti entit, což je přístup, který chápeme jako efektivní cestu ke snadné rozšiřitelnosti bez nutnosti stále dokola nad různými obsahovými entitami implementovat již implementovanou logiku.

Dalším podstatným konceptem je stromové uspořádání jednotlivých obsahových entit, jenž má výhody hlavně ve zvýšení přehlednosti uloženého obsahu, jeho přirozené kategorizaci a jednodušší implementaci oprávnění, která je možné v rámci stromu obsahových entit dědit.



## 3.2 Kernel entity

Objektem typu (**#CONTENT\_ITEM**) je každá položka uložená v systému, která je nositelem obsahu, mohou nad ní být definována oprávnění a může být dále propojena s dalšími položkami.

Objektem typu (**#ITEMS\_CONTAINER**) je každá položka uložená v systému za účelem seskupování položek do složek za účelem zvýšení přehlednosti.

*Pozn.: Specializace kernel entity (**#CONTENT\_ITEM**)*

Objektem typu (**#AUTHOR**) je každá obsahová položka uložená v systému, která reprezentuje autora publikovaných slohových útvarů uložených v systému.

*Pozn.: Specializace kernel entity (**#CONTENT\_ITEM**)*

Objektem typu (**#ARTICLE**) je každá obsahová položka reprezentující slohový útvar uložený v systému.

*Pozn.: Specializace kernel entity (**#CONTENT\_ITEM**)*

Objektem typu (**#COMMENT**) je každá obsahová položka reprezentující slovy vyjádřený názor uživatele služby.

*Pozn.: Specializace kernel entity (**#CONTENT\_ITEM**)*

Objektem typu (**#CATEGORY**) je každá obsahová položka, která může být využita k označení obsahu za účelem zvýšení přehlednosti nebo zvýraznění specifického rysu na základě definovaných kritérií.

*Pozn.: Specializace kernel entity (**#CONTENT\_ITEM**)*

Objektem typu (**#PRINCIPAL**) je každá entita schopná provádět aktivní operace se systémem a současně může vždy aktivně prokázat svou identitu.

Objektem typu (**#USER**) je každá entita schopná provádět aktivní operace se systémem, která je reprezentována autentizovanou osobou, která se registrovala do systému a má právo k němu přistupovat.

*Pozn.: Specializace kernel entity (**#PRINCIPAL**)*

Objektem typu (**#GROUP**) je každá entita schopná provádět aktivní operace se systémem, která je reprezentována vybranou podmnožinou uživatelů systému.

*Pozn.: Specializace kernel entity (**#PRINCIPAL**)*

Objektem typu (**#OPERATION**) je každá podstatná činnost systému, která svým provedením modifikuje data spravovaná systémem.

Objektem typu (**#WORKFLOW**) je každý pracovní postup, který je spravován a prováděn externím systémem a k jeho spuštění dojde automaticky v návaznosti na operace provedené uživatelem v systému.

Objektem typu (**#CONTENT\_TYPE**) je každá identifikace typu obsahu uchovávaného v systému.

Objektem typu (**#RELATIONSHIP\_TYPE**) je každý obecný popis vztahu, který může vzniknout mezi dvěma typy obsahových položek. Vztah s prvním typem obsahové položky nazýváme primárním vztahem a vztah s druhým typem nazýváme sekundárním vztahem.

*Pozn.: Definice významu primárního a sekundárního vztahu je definována jako popisný atribut v této entitě. V jedné implementaci může být primární vztah roven vztahu "jsem rodič" a v jiné může být roven vztahu "jsem dcera". :-)*

Objektem typu (**#EXTENSION\_ATTRIBUTE**) je každý rozšiřující parametr obsahové entity, který nebyl znám v době jejího návrhu nebo nebyl ustanoven jako dostatečně důležitý, aby byl realizován přímo jako její atribut.

### 3.3 Asociativní entity

Objektem typu (**#RELATIONSHIP**) je každá reprezentace vazby mezi obsahovou položkou a typem vazby s významem:

Obsahové položky (**#CONTENT\_ITEM**)-s, které vystupují v primárního vztahu, a které jsou spojeny daným typem vazby (**#RELATIONSHIP\_TYPE**) s danou obsahovou položkou (**#CONTENT\_ITEM**) v sekundárním vztahu. / 0,M : 0,M

Objektem typu (**#EXTENSION\_ATTRIBUTE\_VALUE**) je každá reprezentace vazby mezi připojenou položkou a rozšiřujícím atributem s významem:

Obsahové položky (**#CONTENT\_ITEM**)-s, které jsou rozšířeny o daný rozšiřující atributu (**#EXTENSION\_ATTRIBUTE**). / 0,M:0,M

Objektem typu (**#CONTENT\_ITEM\_PERMISSION**) je každá reprezentace vazby mezi obsahovou položkou, operací a právem na nositele obsahu identitou vyskytující se v systému s významem:

Operace (**#OPERATION**)-s, které má právo vykonávat daná identita (**#PRINCIPAL**) na dané obsahové položce (**#CONTENT\_ITEM**). / 0,M : 0,M

Objektem typu (**#CONTENT\_TYPE\_PERMISSION**) je každá reprezentace vazby mezi typem obsahové položky, operací a právem na typ nositele identitou vyskytující se v systému obsahu s významem:

Operace (**#OPERATION**)-s, které má právo vykonávat daná identita (**#PRINCIPAL**) na daném typu obsahové položky (**#CONTENT\_TYPE**). / 0,M : 0,M

### 3.4 Vazby

1. Typ obsahové položky (**#CONTENT\_TYPE**), který je rozšířitelný o daný rozšiřující atribut (**#EXTENSION\_ATTRIBUTE**) / 1:1,0:M
2. Typ obsahové položky (**#CONTENT\_TYPE**), který může být rozšířen o daný typ vztahu (**#RELATIONSHIP\_TYPE**) v primárním smyslu. / 1:1, 0:M
3. Typ obsahové položky (**#CONTENT\_TYPE**), který může být rozšířen o daný typ vztahu (**#RELATIONSHIP\_TYPE**) v sekundárním smyslu. / 1:1, 0:M
4. Obsahová položka (**#CONTENT\_ITEM**), která je rodičem (v hierarchickém uspořádání všech obsahových položek) dané obsahové položky (**#CONTENT\_ITEM**). / 0,1 : 0,M
5. Typ obsahové položky (**#CONTENT\_TYPE**), který je typem dané obsahové položky (**#CONTENT\_ITEM**). / 1,1 : 0,M
6. Autor (**#AUTHOR**), který vytvořil daný článek (**#ARTICLE**). /1,1:0,M
7. Článek (**#ARTICLE**), který obsahuje daný komentář (**#COMMENT**). /1,1:0,M
8. Typy obsahových položek (**#CONTENT\_TYPE**)-s, které využívají danou operaci (**#OPERATION**) / 1,M:0,M
9. Operace (**#OPERATION**), po jejíž dokončení se provede daný průběh práce (**#WORKFLOW**) / 1,1 : 0,M
10. Uživatelé (**#USER**)-s, kteří jsou součástí dané skupiny (**#GROUP**) / 0,M:0,M

## 4 Druhá fáze

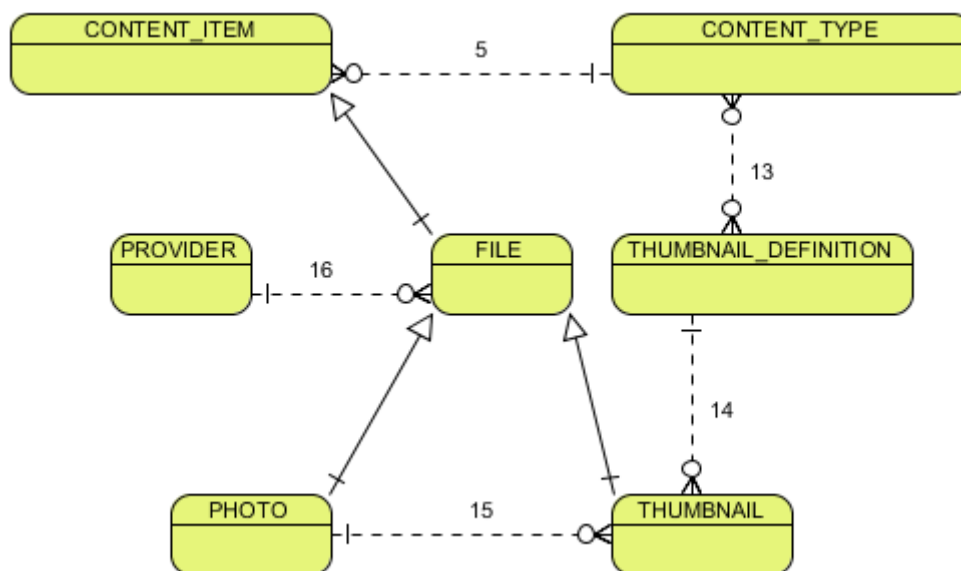
Analytické vzory jsme použili již v první fázi, abychom pomocí nich docílili jednoduché rozšiřitelnosti obsahových entit, což byl jeden z hlavních cílů našeho CMS.

Proto jsme se rozhodli navrhovaný model CMS rozšířit o 2 nové subsystemy, které se v rámci CMS velice často vyskytují - uložení souborů a správa fotografií, včetně automatického generování náhledů. Současně jsme si tímto rozšířením chtěli vyzkoušet, že navržené jádro bude vyžadovat jen minimum změn při dalším rozšíření funkcionality.

Při návrhu systému pro uložení souborů jsme využili již navržený mechanismus pro správu univerzálního obsahu, který jsme pouze rozšířili. Tím jsme všechny jeho předchozí vlastnosti s minimem úsilí přenesli i do tohoto nového modulu.

Při návrhu modulu pro správu fotografií jsme si uvědomili, že fotografie je zvláštním druhem souboru, čemuž jsme navrhovaný systém přizpůsobili.

### 4.1 ER diagram



## 4.2 Kernel entity

Objektem typu (**#PROVIDER**) je každé aplikační rozhraní poskytované svou specifickou formou, zprostředkávající služby interního systému nebo externí služby, které umožňují provádět množinu základních operací nad souborem.

Objektem typu (**#FILE**) je každá obsahová položka, která reprezentuje odkaz na pojmenovanou posloupnost bytů, doplněnou o pevně definovaná metadata, uloženou v systému poskytovatele.

*Pozn.: Specializace kernel entity (**#CONTENT\_ITEM**)*

Objektem typu (**#PHOTO**) je každý soubor, který obsahuje výhradně statická obrazová data ve zpracovatelném formátu pomocí dostupných prostředků.

*Pozn.: Specializace kernel entity (**#FILE**)*

Objektem typu (**#THUMBNAIL**) je každý soubor, který obsahuje výhradně statická obrazová data a vznikl v rámci automatického procesu zpracování.

*Pozn.: Specializace kernel entity (**#FILE**)*

Objektem typu (**#THUMBNAIL\_DEFINITION**) je každý automaticky zpracovatelný popis operací nad statickými obrazovými daty, který vytváří ze vstupních obrazových dat nová výstupní obrazová data stejného významu, ale jiné kvality či rozměrů

## 4.3 Vazby

13. Identifikace typů obsahu (**#CONTENT\_TYPE**)-s, které používají danou definici náhledu (**#THUMBNAIL\_DEFINITION**). / 0,M : 0,M
14. Definice náhledu (**#THUMBNAIL\_DEFINITION**), pomocí kterého byl vytvořen daný náhled (**#THUMBNAIL**). / 1,1 : 0,M
15. Obrázek (**#PHOTO**), který slouží jako podklad pro vytvoření daného náhledu (**#THUMBNAIL**) / 1,1:0,M
16. Poskytovatel (**#PROVIDER**), u kterého je uložen daný soubor (**#FILE**). / 1,1 : 0,M

## 5 Výsledný ER diagram

