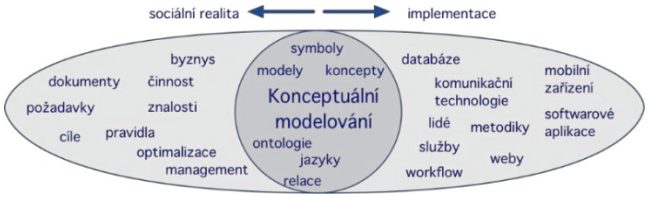
PODPORA KONCEPTUÁLNÍHO MODELOVÁNÍ PRO JAZYK HASKELL

Ing. Marek Suchánek **Ing. Robert Pergl, Ph.D.** (vedoucí práce)

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

MOTIVACE

Pro **pochopení problémové domény** nejen za účelem tvorby informačních systémů používáme konceptuální modelování. Konceptuální model však bývá oddělen od implementace a mohou tak vznikat **problémy s udržováním konzistence** vlivem různých změn jak úrovni domény, tak i implementace, a to i při použití MDD. [1]



Některé současné vyšší programovací jazyky disponují takovými vlastnosti, že se jeví jako vhodný prostředek pro zápis konceptuálních modelů. To by mohlo zmíněný problém odstranit, **zaručit konzistenci modelu s implementací** a přinést mnohé další výhody. Mezi takové jazyky se řadí svou expresivitou a propracovaným typovým systémem právě **čistě funkcionální jazyk Haskell**.

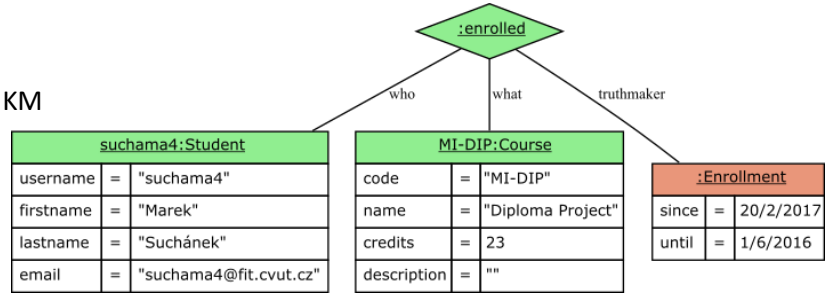
CÍLE A OBSAH PRÁCE

- Rešerše konceptuálního modelování včetně validace, verifikace a významu v SW inženýrství
- Analýza současných přístupů a metod
- Stanovení požadavků na systém pro kódování KM
- Návrh systému pro kódování KM v Haskell
- Implementace prototypu systému
- Demonstrace prototypu na případové studii
- Ekonomicko-manažerské zhodnocení

KONCEPTUÁLNÍ MODELOVÁNÍ S hCM

Systém pro podporu konceptuálního modelování v jazyce Haskell **hCM** navržený na základě rešerše obecných postupů, grafických notací (UML, OntoUML, ORM, …) i metod formální specifikace (Alloy, B, OCL, VDM, Z, …) umožňuje zachycení konceptuálního modelu **přímo v kódu** a provádět verifikaci, validaci a vizualizaci. Systém zahrnuje navíc tzv. **instanční modelování**, které je důležité právě pro proces validace, kdy se zjišťuje, zda instance modelu odpovídají instancím v reálném světě.

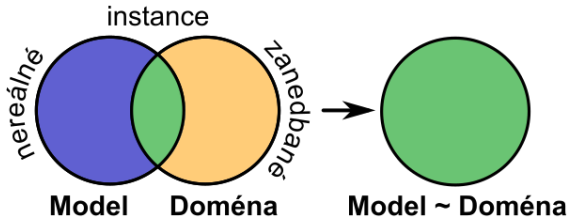
1. Zápis modelu za pomoci připravených typových tříd, datových typů a funkcí v rámci tzv. **kompilátorem řízeného modelování** (kompilátor provádí uživatele tvorbou modelu). Model tvoří specifikace struktury, chování a omezení na různých úrovních.
2. Vytváření instancí modelu či specifikace **pseudonáhodného generování frameworkem QuickCheck** (je možné použít i ve fázi testování implementace).
3. **Vizualizace** konceptuálního **modelu struktury** (koncepty, atributy, relace) jazykem DOT **i instancí s indikací přípustnosti** dle specifikovaných omezení.
4. **Validace modelu** pomocí připravených instancí se zákazníkem **dle metodiky** a případná úprava modelu.



VÝSLEDNÉ ŘEŠENÍ

Knihovnu **hCM** lze velmi snadno použít běžnou cestou nástroji *stack* či *cabal* vyžívajících uložiště Haskell balíčků **Hackage**. Díky své jednoduchosti, licenci, dokumentaci a umístění na **GitHub** lze knihovnu i případovou studii dále rozvíjet a rozšiřovat podle vlastních potřeb.

Zápis modelu přímo v kódu přináší při tvorbě informačních systémů možné **úspory ve formě času a finančních prostředků projektu** i **kvalitnější výsledek reflektující reálné potřeby** dané problémové domény a **s lepší evolvabilitou**. Zjištění a odstranění chyb při analýze a návrhu bývá i **100× levnější** než až po nasazení implementace (v rámci změnových řízení) [2].



A CO DÁL?

Na práci lze navázat rozvojem knihovny, návrhem a vývojem podpůrných nástrojů (GUI, částečná automatizace validace, překlad do Alloy, OntoUML a dalších, …), výzkumem použitelnosti dalších programovacích jazyků s jinými vlastnostmi či využitím metamodelů z vyšších ontologií, jako jsou například UFO, BFO nebo DOLCE.

Autor bude dále pokračovat v DSP s dizertační prací „Vztah konceptuálního modelování k přesným technickým specifikacím“ na FIT ČVUT v Praze.

Tato práce vnikla ve spolupráci s Centrem pro konceptuální modelování a implementace, FIT ČVUT v Praze (ccmi.fit.cvut.cz)

[1] ROBINSON, Stewart et al.: ***Conceptual modeling: Definition, purpose and benefits***. In: 2015 WSC, 2015, IEEE, s. 2812–2826
[2] BOEHM, Barry a Victor R. BASILI: ***Software Defect Reduction Top 10 List***, In: Computer č. 34, leden 2001, IEEE, s. 135–137

ACM IT SPY
29. 7. 2017