

Evoluce a adaptabilita komplexních XML aplikací (a nejen XML)

Irena Holubová (Mlýnková)

Katedra softwarového inženýrství
Matematicko-fyzikální fakulta
Univerzita Karlova v Praze



holubova@ksi.mff.cuni.cz
<http://www.ksi.mff.cuni.cz/~holubova/>

Motto

***To improve is to change;
to be perfect is to change often.***

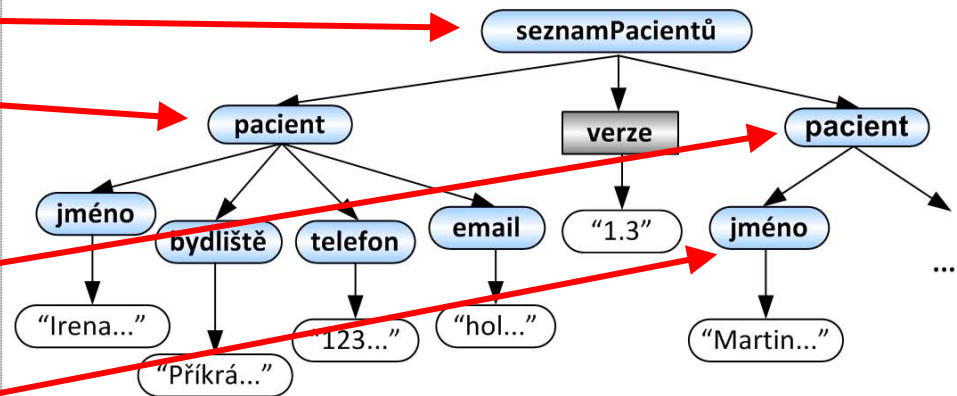
Winston Churchill



Evoluce a adaptabilita komplexních XML aplikací

```
<seznamPacientů verze="1.3">
  <pacient>
    <jméno>Irena Holubová</jméno>
    <bydliště>Příkrá 123, Praha 4</bydliště>
    <telefon>123 456 789</telefon>
    <email>holubova@ksi.mff.cuni.cz</email>
  </pacient>
  <pacient>
    <jméno>Martin Nečaský</jméno>
    <bydliště>Hlučná 567, Praha 5</bydliště>
    <zam>Malostranské nám. 25, Praha 1</zam>
    <telefon>111 222 333</telefon>
    <telefon>555 444 333</telefon>
    <email>necasky@ksi.mff.cuni.cz</email>
  </pacient>
</seznamPacientů>
```

XML dokument



Strom XML dokumentu

- eXtensible Markup Language
 - Standardní formát pro ukládání a výměnu dat
 - Standardizuje World Wide Web Consortium (W3C)
 - XML dokument = data + metadata
 - Doplněn rozsáhlou sadou XML technologií
 - Popis schématu, dotazování, transformace, aktualizace, ...

Evoluce a adaptabilita komplexních XML aplikací

```

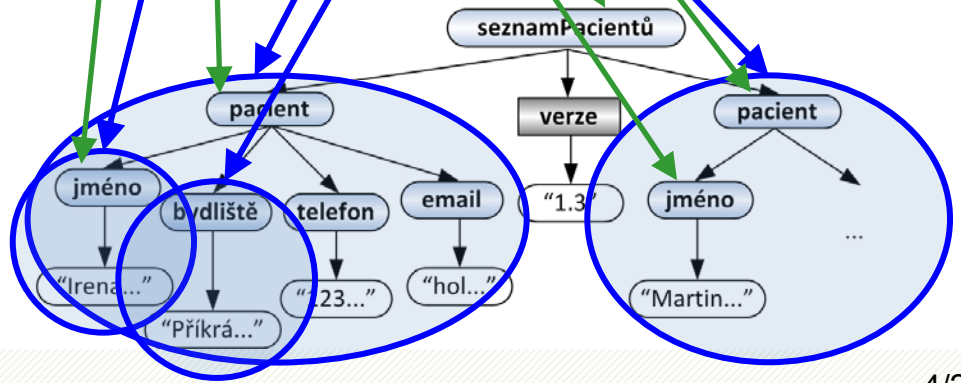
<element name="seznamPacientů">
  <complexType>
    <attribute name="verze" type="string"/>
    <sequence>
      <element name="pacient" type="Tpacient"
        maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>

<complexType name="Tpacient">
  <sequence>
    <element name="jméno" type="string"/>
    <element name="bydliště" type="string"/>
    <element name="zam" type="string"
      minOccurs="0"/>
    <element name="telefon" type="string"
      minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="email" type="string"/>
  </sequence>
</complexType>
  
```

S	→ seznamPacientů (Tverze : Tpacient+)	
Tpacient	→ pacient (Tjméno, Tbydliště, Tzam?, Ttelefon*, Temail)	
Tjméno	→ jméno (string)	
Tbydliště	→ bydliště (string)	
Tzam	→ zam (string)	(Rozšířená)
Ttelefon	→ telefon (string)	regulární
Temail	→ email (string)	stromová
Tverze	→ verze (string)	gramatika

```

for $p in /seznamPacientů/pacient
let $pocetTelefonu = count($p/telefon)
let $pocetEmailu = count($p/email)
where $pocetTelefonu = 1
order by $pocetTelefonu + $pocetEmailu ascending
return $p/jmeno $p/bydliště
  
```



XML schéma v jazyce XML Schema

Evoluce a adaptabilita komplexních XML aplikací

□ Evoluce

- Aplikace jsou typicky dynamické
 - Jen málo aplikací zůstává po implementaci a nasazení beze změn
- Důvod: nové uživatelské požadavky, změny okolního světa, s nímž aplikace komunikuje, ...

□ Adaptabilita

- Řešení problému evoluce: korektní a úplná modifikace ovlivněných částí

□ My se zaměřujeme na:

- XML aplikace = XML dokumenty, XML schémata, XML dotazy (operace), ...
- Změny v datech a jejich korektní propagaci v rámci celého systému

Evoluce a adaptabilita komplexních XML aplikací

```
<element name="seznamPacientů">  
  <complexType>  
    <attribute name="verze" type="string"/>  
    <sequence>  
      <element name="pacient" type="Tpacient"  
        minOccurs="1" maxOccurs="unbounded"/>  
    </sequence>  
  </complexType>  
</element>
```

```
for $p in /seznamPacientů/pacient  
let $pocetTelefonu = count($p/telefon)  
let $pocetEmailu = count($p/email)  
where $pocetTelefonu = 1  
order by $pocetTelefonu + $pocetEmailu ascending  
return $p/jmeno, $p/bydliště
```

Nový požadavek: Chceme strukturovat a rozšířit adresu bydliště.

```
<complexType name="Tpacient">  
  <sequence>  
    <element name="jméno" type="string"/>  
    <element name="bydliště" type="string"/>  
    <element name="zam" type="string"  
      minOccurs="0"/>  
    <element name="telefon" type="string"  
      minOccurs="0" maxOccurs="unbounded"/>  
    <element name="email" type="string"/>  
  </sequence>  
</complexType>
```

```
<seznamPacientů verze="1.5">  
  <pacient>  
    <jméno>Příkrá, Holubová</jméno>  
    <bydliště>Příkrá 123, Praha 4</bydliště>  
    <telefon>123 456 789</telefon>  
    <email>holubova@ksi.mff.cuni.cz</email>  
  </pacient>  
  <pacient>  
    <jméno>Martin Nečaský</jméno>  
    <bydliště>Hlučná 567, Praha 5</bydliště>  
    <zam>Malostranské nám. 25, Praha 1</zam>  
    <telefon>111 222 333</telefon>  
    <telefon>555 444 333</telefon>  
    <email>necasky@ksi.mff.cuni.cz</email>  
  </pacient>  
</seznamPacientů>
```

Evoluce a adaptabilita komplexních XML aplikací

```
<element name="bydliště">  
<complexType>  
  <sequence>  
    <element name="ulice" type="string"/>  
    <element name="čDomu" type="int"/>  
    <element name="město" type="string"/>  
    <element name="stát" type="string"  
      minOccurs="0"/>  
  </sequence>  
</complexType>  
</element>
```

```
<sequence>  
  <element name="jméno" type="string"/>  
  <element name="bydliště" type="string"/>  
  <element name="zam" type="string"  
    minOccurs="0"/>  
  <element name="telefon" type="string"  
    minOccurs="0"  
    maxOccurs="unbounded"/>  
  <element name="email" type="string"/>  
</sequence>  
</complexType>
```

```
for $p in /seznamPacientů/pacient  
let $pocetTelefonu = count($p/telefon)  
let $pocetEmailu = count($p/email)  
where $pocetTelefonu > 1  
order by $pocetTelefonu + $pocetEmailu ascending  
return $p/jmeno, $p/bydliště
```

```
<seznamPacientů verze="1.3">  
  <pacient>  
    <jméno>Irena Holubová</jméno>  
    <bydliště>Příkrá 123, Praha 4</bydliště>  
    <telefon>123 456 789</telefon>  
    <email>holubova@ksi.mff.cuni.cz</email>  
  </pacient>  
  <pacient>  
    <jméno>Martin Nečaský</jméno>  
    <bydliště>Hlučná 567, Praha 5</bydliště>  
    <zam>Malostranské nám. 25, Praha 1</zam>  
    <telefon>111 222 333</telefon>  
    <telefon>555 444 333</telefon>  
    <email>necasky@ksi.mff.cuni.cz</email>  
  </pacient>  
</seznamPacientů>
```

Evoluce a adaptabilita komplexních XML aplikací

```
<element name="bydliště">  
<complexType>  
  <sequence>  
    <element name="ulice" type="string"/>  
    <element name="čDomu" type="int"/>  
    <element name="město" type="string"/>  
    <element name="stát" type="string"  
      minOccurs="0"/>  
  </sequence>  
</complexType>  
</element>
```

```
<sequence>  
  <element name="jméno" type="string"/>  
  <element name="bydliště" type="string"/>  
  <element name="zam" type="string"  
    minOccurs="0"/>  
  <element name="telefon" type="string"  
    minOccurs="0"  
    maxOccurs="unbounded"/>  
  <element name="email" type="string"/>  
</sequence>  
</complexType>
```

```
for $p in /seznamPacientů/pacient  
let $pocetTelefonu = count($p/telefon)  
let $pocetEmailu = count($p/email)  
where $p  
return $p/jmeno, $p/ulice, $p/čDomu, $p/město  
order by $pocetTelefonu + $pocetEmailu ascending  
return $p/jmeno, $p/bydliště
```

```
<seznamPacientů verze="1.3">
```

```
<pacient>  
  <jméno>Irena Holubová</jméno>  
  <bydliště>Příkrá 123, Praha 4</bydliště>  
  <telefon>123 456 789</telefon>  
  <email>holubova@ksi.mff.cu.cz</email>  
</pacient>  
<pacient>  
  <jméno>Martin Nečaský</jméno>  
  <bydliště>Hlučná 567, Praha 5</bydliště>  
  <zam>Malostranské nám. 123</zam>  
  <telefon>111 222 333</telefon>  
  <telefon>555 444 333</telefon>  
  <email>necasky@ksi.mff.cu.cz</email>  
</pacient>  
</seznamPacientů>
```

```
<bydliště>  
  <ulice>Příkrá</ulice>  
  <čDomu>123</čDomu>  
  <město>Praha 4</město>  
  <stát>ČR</stát>  
</bydliště>
```

```
<bydliště>  
  <ulice>Hlučná</ulice>  
  <čDomu>567</čDomu>  
  <město>Praha 5</město>  
  <stát>ČR</stát>  
</bydliště>
```


Evoluce a adaptabilita komplexních XML aplikací

- Typicky obsahují aplikace **více schémat**
 - Seznam pacientů, seznam faktur, objednávka, potvrzení objednávky, seznam léků, seznam lékáren, přehled zboží na skladě, ...
- Informace ve schématech se mohou **překrývat**
 - Pacient je reprezentován v seznamu pacientů, jako odběratel na faktuře, ...
 - Lék je reprezentován v seznamu léků, v seznamu položek na faktuře, v seznamu zboží na skladě, ...
- Data mohou mít **různou sémantiku** (kontext)
 - Adresa bydliště, adresa dodavatele, adresa zaměstnavatele, adresa výrobce, ...
- ...

Evoluce a adaptabilita komplexních XML aplikací

```
<seznamPacientů verze="1.3">
  <pacient>
    <jméno>Irena Holubová</jméno>
    <bydliště>Příkrá 123, Praha 4</bydliště>
    <telefon>123 456 789</telefon>
    <email>holubova@ksi.mff.cuni.cz</email>
```

```
</pacient>
  <pacient>
    <jméno>Martin Nečaský</jméno>
    <bydliště>Hlíbová 45, Praha 4</bydliště>
    <zam>Malostranské nám. 25, Praha 1</zam>
    <telefon>111 222 333</telefon>
    <telefon>555 444 333</telefon>
    <email>necasky@ksi.mff.cuni.cz</email>
  </pacient>
</seznamPacientů>
```

```
<seznamFaktur>
  <faktura id="1" typ="vydaná">
    <datum>5.1. 2013</datum>
    <jméno>Irena Holubová</jméno>
    <adresa>Příkrá 123, Praha 4</adresa>
    <položky>
```

```
<položka>očkování</položka>
  <množství>1</množství>
</položky>
  <faktura id="2" typ="přijatá">
    <datum>13.5. 2013</datum>
    <jméno>Zentiva</jméno>
    <adresa>U Zentivy 99, Brno</adresa>
    <položky>
      <položka>dezinfekce</položka>
      <množství>100</množství>
    </položky>
  </faktura>
</seznamFaktur>
```

Nový požadavek: Chceme strukturovat a rozšířit adresy bydliště pacientů.

Evoluce a adaptabilita komplexních XML aplikací

```
<seznamPacientů verze="1.3">
  <pacient>
    <jméno>Irena Holubová</jméno>
    <bydliště>Příkrá 123, Praha 4</bydliště>
    <telefon>123 456 789</telefon>
    <email>holubova@ksi.mff.cuni.cz</email>
  </pacient>
  <pacient>
    <jméno>Martin Nečaský</jméno>
    <bydliště>Hlučná 567, Praha 5</bydliště>
    <zam>Malostranské nám. 25, Praha 1</zam>
    <telefon>111 222 333</telefon>
    <telefon>555 444 333</telefon>
    <email>necasky@ksi.mff.cuni.cz</email>
  </pacient>
</seznamPacientů>
```

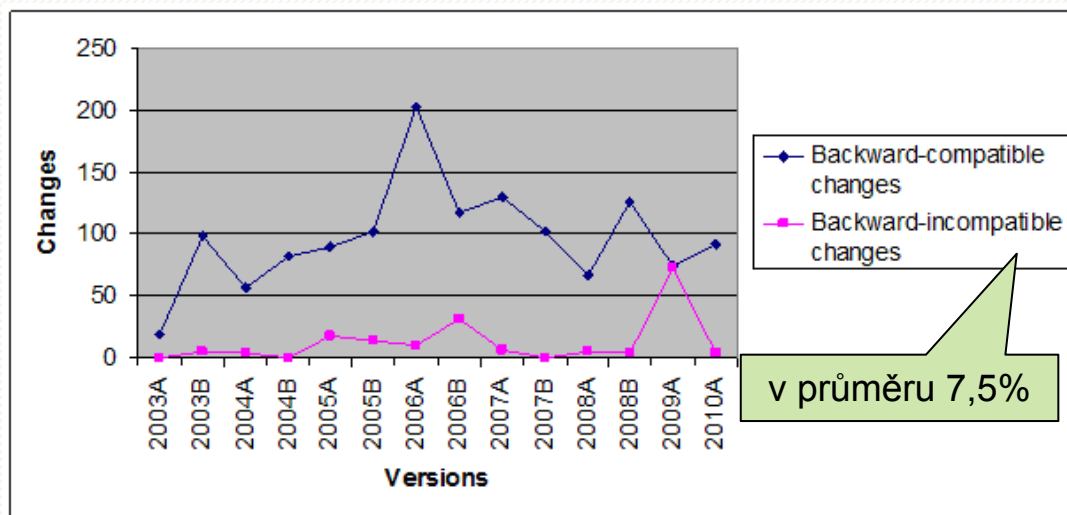
```
<seznamFaktur>
  <faktura id="1" typ="vydaná">
    <datum>5.1. 2013</datum>
    <jméno>Irena Holubová</jméno>
    <adresa>Příkrá 123, Praha 4</adresa>
    <položky>
      <položka>očkování</položka>
      <množství>1</množství>
    </položky>
  </faktura>
  <faktura id="2" typ="přijatá">
    <datum>13.5. 2013</datum>
    <jméno>Zentiva</jméno>
    <adresa>U Zentivy 99, Brno</adresa>
    <položky>
      <položka>dezinfekce</položka>
      <množství>100</množství>
    </položky>
  </faktura>
</seznamFaktur>
```

Evoluce a adaptabilita komplexních XML aplikací

- Standardní XML formáty obsahují stovky XML schémat
 - HL7 (Health Level Seven) – lékařské záznamy
 - OASIS UBL (Universal Business Language) – obchodní data
 - OpenTravel – cestovní ruch
 - Google AdWords Web Services – inzerce
 - ...



- Příklad: **OpenTravel.org**
 - Od roku 2001
 - Nyní: 333 XML schémat
 - Změny standardu 2x ročně
 - Řešení adaptability: zpětná kompatibilita
 - Složitá schémata
 - „Nepřirozené“ struktury
 - Nečitelné



Evoluce a adaptabilita komplexních XML aplikací (a nejen XML)

- Aplikace nemusí mít data reprezentována pouze v XML
 - Typicky jsou to různé formáty pro různé podproblémy

□ Např.

polyglot representation / persistence

- Relační data: pro ukládání seznamů (pacientů, léků, faktur, ...)
- XML data: pro komunikaci mezi Webovými službami (objednávky, ...)
- Grafová data: pro reprezentaci interakcí mezi léky, ...

Pacienti

id	jméno	adresa
1	Irena Holubová	Příkrá 123
2	Martin Nečaský	Hlučná 567

Léky

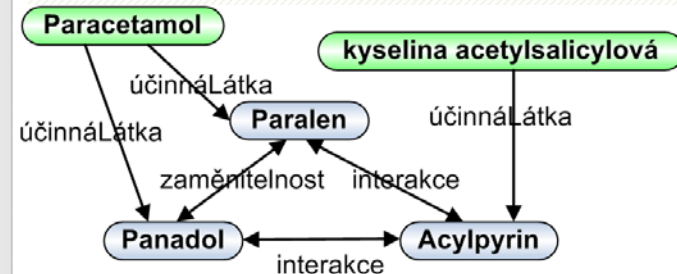
id	název	výrobce
1	Paralen	Zentiva
2	Panadol	GS

Uživá

idPacienta	idLéku
1	1
2	1

```
<objednávka id="5492013">
  <názevFirmy>U teploměru</názevFirmy>
  <adresa>U mostu 1, Praha 1</bydliště>
  <telefon>123 444 555</telefon>
  <položka>
    <název>Paralen</název>
    <množství>1000</množství>
    <cena>10Kč</cena>
  </položka>
</objednávka>
```

```
<potvrzeníObjednávky id="5492013">
  <názevFirmy>U teploměru</názevFirmy>
  <adresa>U mostu 1, Praha 1</bydliště>
  <telefon>123 444 555</telefon>
  <termínDodání>25.10. 2013</termínDodání>
</potvrzeníObjednávky >
```

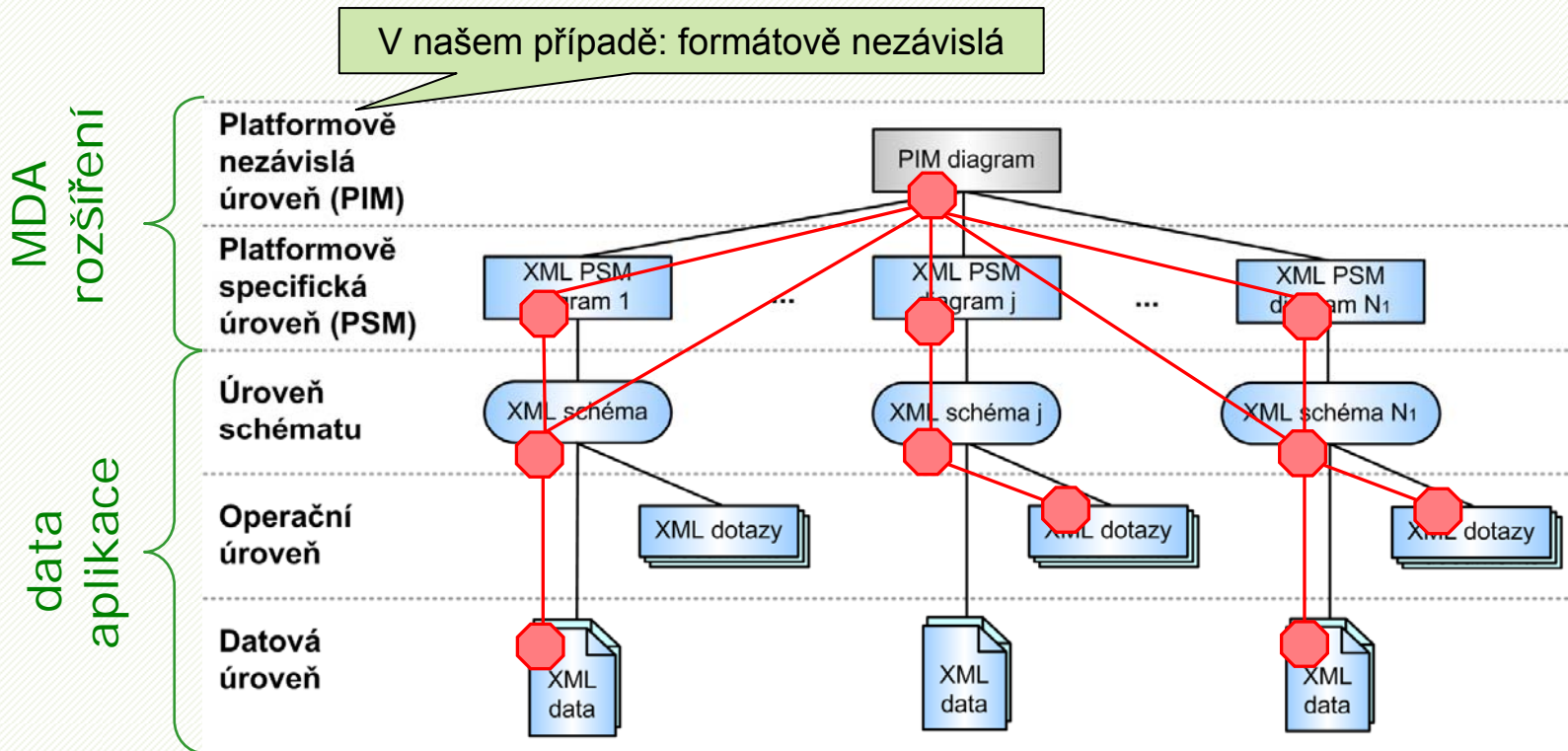


Specifikace problému

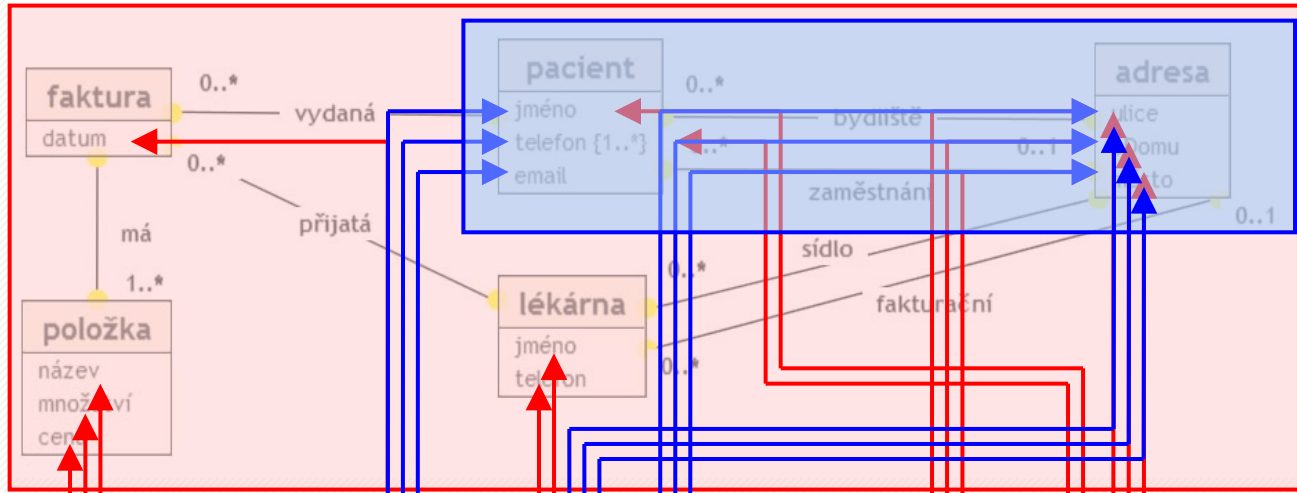
- Máme:
 - N datových **formátů**
 - Např. XML, relační, RDF, ...
 - Pro každé $i \in [1, N]$ máme N_i **schémat**
 - Např. pro XML formát máme schéma pro objednávku, potvrzení objednávky, výpis seznamu pacientů, ...
 - Pro každé $k \in [1, N_i]$ máme M_k **instancí** schématu k
 - Např. pro XML schéma objednávky máme M_k XML dokumentů obsahujících data o konkrétních objednávkách
 - Pro každé $k \in [1, N_i]$ máme O_k **operací**
- Problém: Chceme zajistit korektní a úplnou propagaci změn
 - Schémat jsou stovky, překrývají se, objekty mají různou sémantiku, ...
→ **manuální úpravy** jsou **náročné**, náchylné k chybám, nepřesnostem, neúplnostem, ... **nemožné**
- Pro jednoduchost: Uvažujme nejprve, že $N = 1$ + uvažovaným formátem je XML

Řešení: využití principů modelem řízeného návrhu

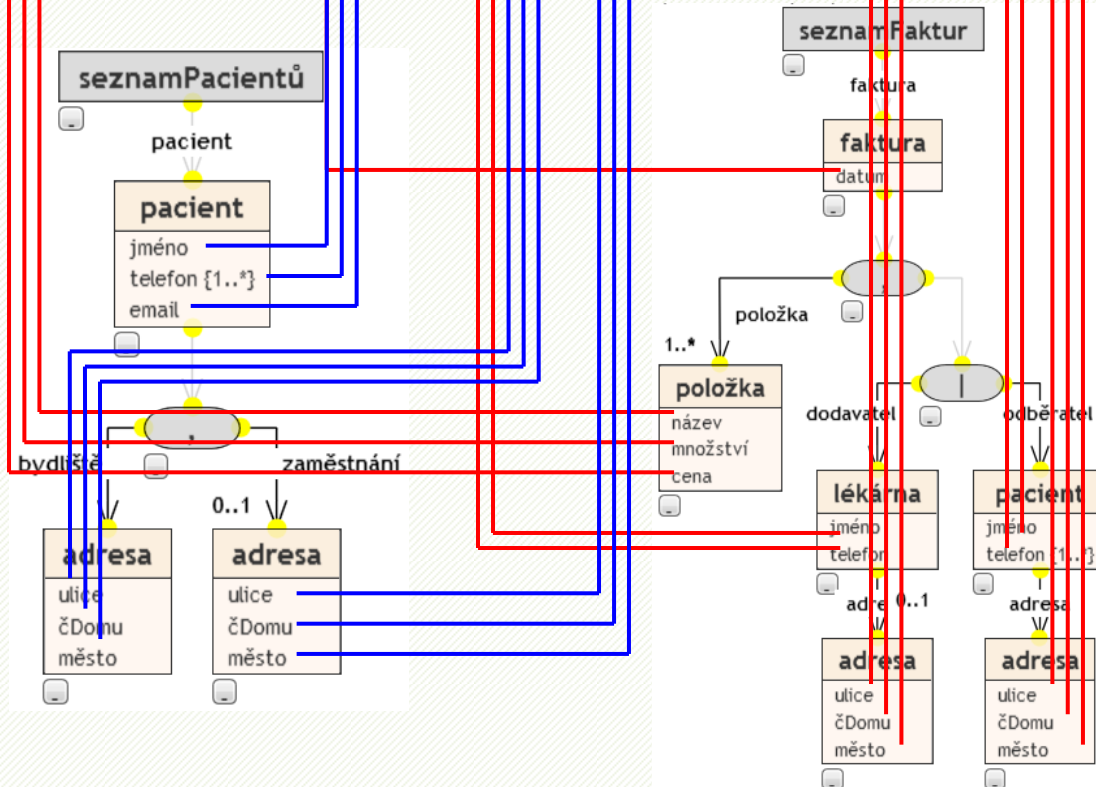
- Modelem řízený návrh (Model-Driven Architecture - MDA)
 - Původně pro objektově-orientované programování
 - Několik úrovní specifikace návrhu
 - Cíl: oddělit aplikační logiku od technologické platformy



PIM úroveň

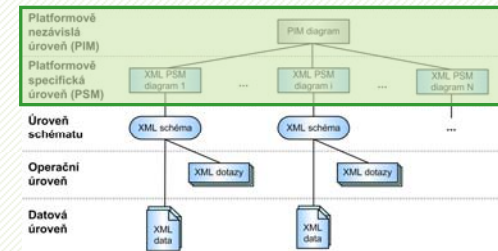


PSM úroveň

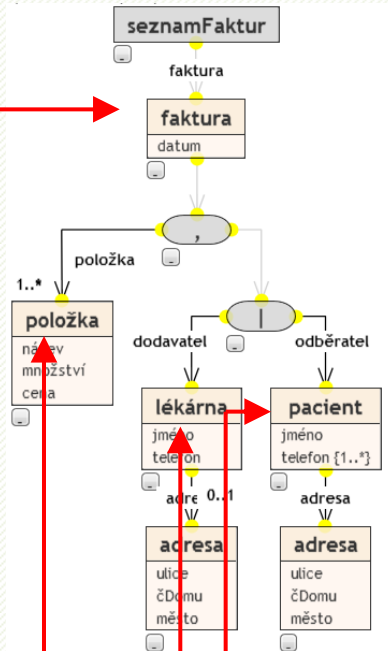
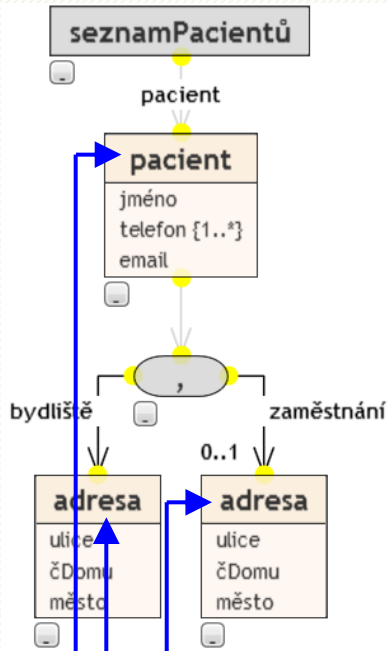


- Koncepty reálného světa
- Vztahy mezi nimi
- **Bez ohledu na zvolený datový formát**

- Reprezentace zvolené části reality v modelu XML formátu
- **Mapování** mezi úrovní PIM a úrovní schématu



PSM úroveň



- Reprezentace zvolené části reality v modelu XML formátu

Úroveň schématu

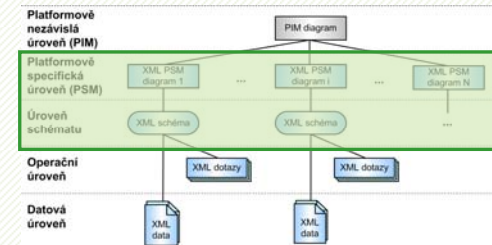
```

<element name="seznamPacientů">
  <complexType>
    <attribute name="verze" type="string"/>
    <sequence>
      <element name="pacient" type="Tpacient"
        maxOccurs="unbounded">
      </element>
    </sequence>
  </complexType>
</element>
<complexType name="Tpacient">
  <sequence>
    <element name="jméno" type="string"/>
    <element name="bydliště" type="string"/>
    <element name="zam" type="string"
      minOccurs="0"/>
    <element name="telefon" type="string"
      minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="email" type="string"/>
  </sequence>
</complexType>
  
```

```

<element name="seznamFaktur">
  <complexType>
    <sequence>
      <element name="faktura" type="Tfaktura"
        maxOccurs="unbounded">
      </element>
    </sequence>
  </complexType>
</element>
<complexType name="Tfaktura">
  <sequence>
    <element name="datum" type="string"/>
    <element name="položka" type="Tpoložka"/>
    <choice>
      <element name="lékárna" type="Tlékárna"/>
      <element name="pacient" type="Tpacient"/>
    </choice>
  </sequence>
</complexType>
<complexType name="Tlékárna">
  ...
</complexType>
<complexType name="Tpacient">
  ...
</complexType>
  
```

- Vyjádření XML modelu v XML Schema



Úroveň schématu

```

<element name="seznamPacientů">
  <complexType>
    <attribute name="verze" type="string"/>
    <sequence>
      <element name="pacient" type="Tpacient"
        maxOccurs="unbounded">
      </sequence>
    </complexType>
  </element>
  <complexType name="Tpacient">
    <sequence>
      <element name="jméno" type="string"/>
      <element name="bydliště" type="string"/>
      <element name="zam" type="string"
        minOccurs="0"/>
      <element name="telefon" type="string"
        minOccurs="0"
        maxOccurs="unbounded"/>
      <element name="email" type="string"/>
    </sequence>
  </complexType>

```

XML schéma

XML dotazy

```

for $p in /seznamPacientů/pacient
let $pocetTelefonu = count($p/telefon)
let $pocetEmailu = count($p/email)
where $pocetTelefonu = 1
order by $pocetTelefonu + $pocetEmailu ascending
return $p/jmeno, $p/bydliště

```

Datová úroveň

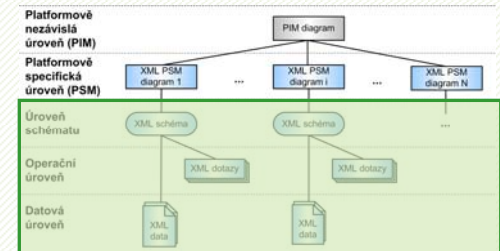
```

<seznamPacientů verze="1.3">
  <pacient>
    <jméno>Irena Holubová</jméno>
    <bydliště>Příkrá 123, Praha 4</bydliště>
    <telefon>123 456 789</telefon>
    <email>holubova@ksi.mff.cuni.cz</email>
  </pacient>
  <pacient>
    <jméno>Martin Nečaský</jméno>
    <bydliště>Hlučná 567, Praha 5</bydliště>
    <zam>Malostranské nám. 25, Praha 1</zam>
    <telefon>111 222 333</telefon>
    <telefon>555 444 333</telefon>
    <email>necasky@ksi.mff.cuni.cz</email>
  </pacient>
</seznamPacientů>

```

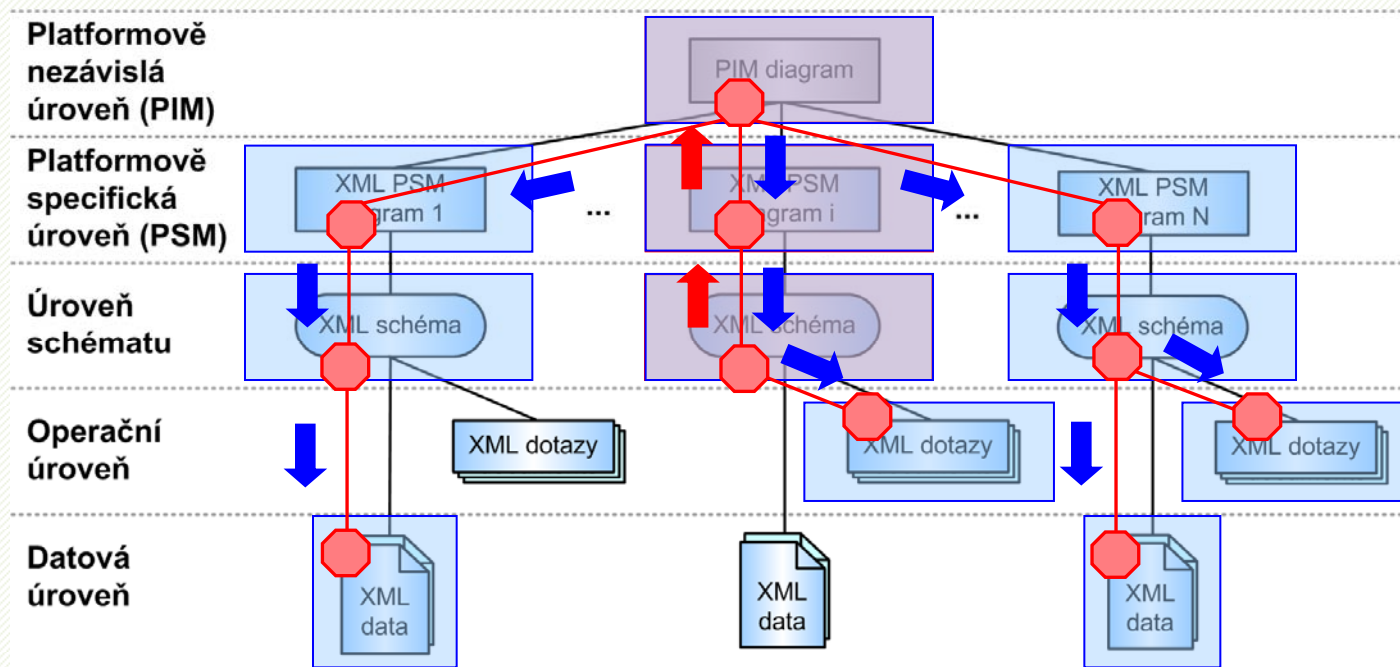
XML dokumenty

Operační úroveň



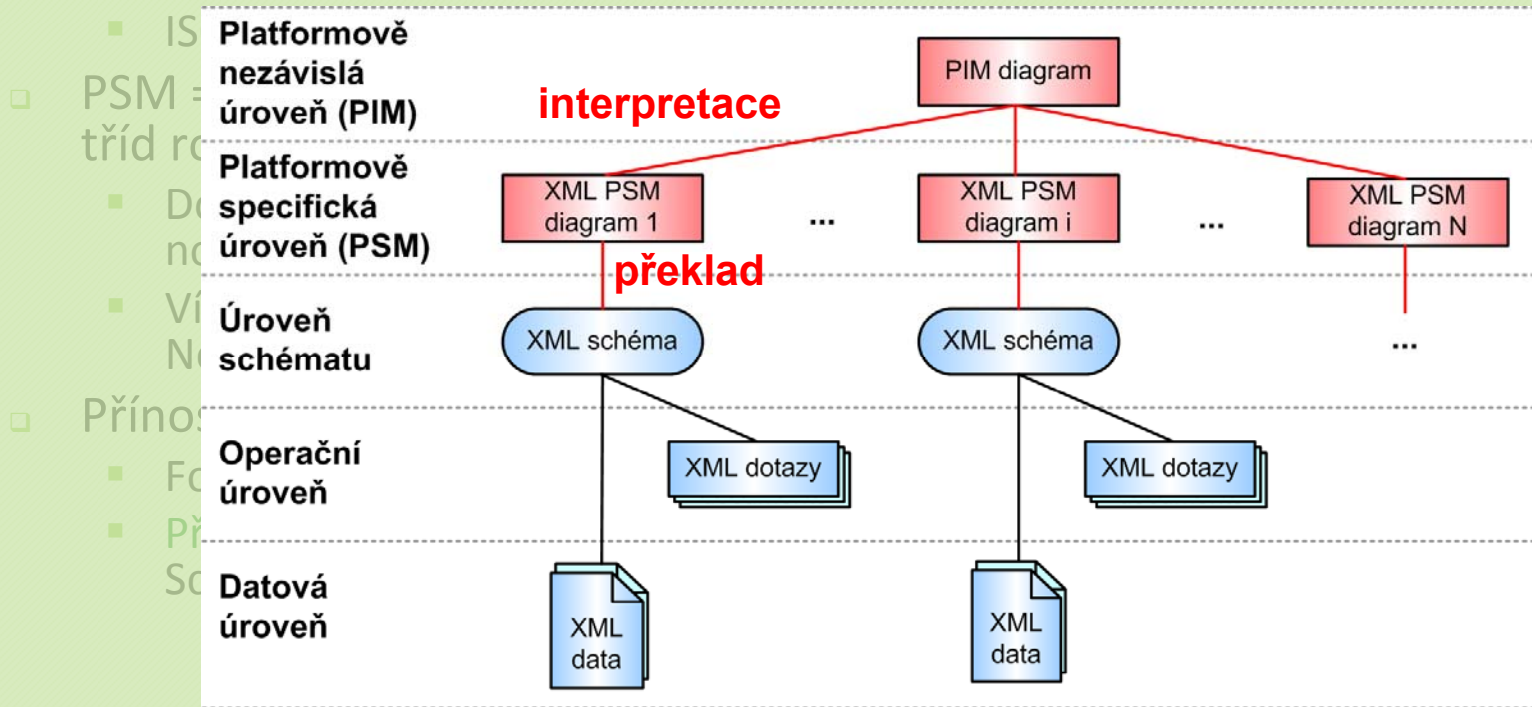
Jak probíhá propagace změn?

- Fáze I. **Vzestupná propagace**
 - Hledáme část reality, která je ovlivněna změnou
- Fáze II. **Sestupná propagace**
 - Hledáme všechna místa, kde se s danou částí reality pracuje



Problém 1. Korektní definice modelů a mapování mezi nimi

- PIM = podmnožina UML diagramu tříd



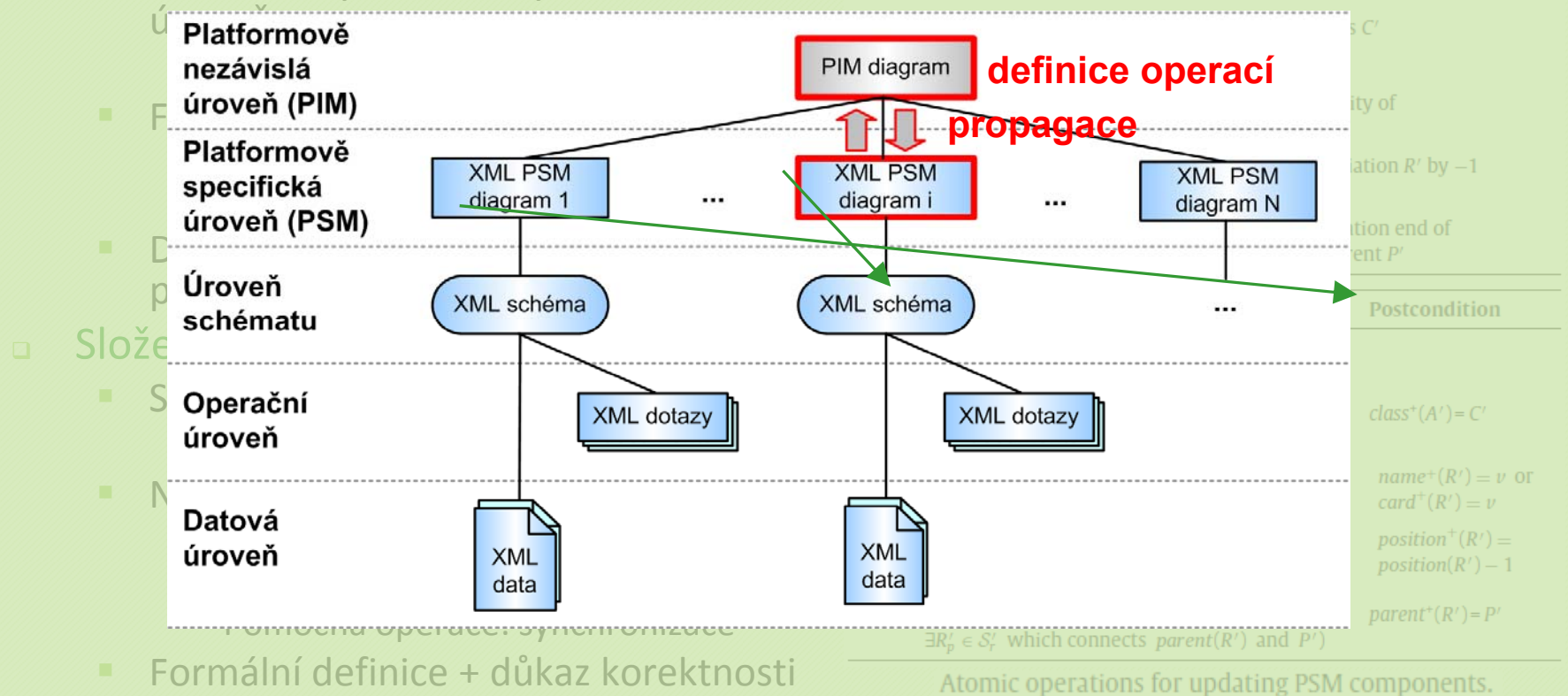
- Příno...
 - Fo...
 - Př...
 - Sc...
- Definice interpretace = formální definice mapování PSM ↔ PIM

Necasky, M. - Mlynkova I. - Klimek, J. - Maly, J.: **When Conceptual Model Meets Grammar: A Dual Approach to XML Data Modeling.** *International Journal on Data & Knowledge Engineering*, volume 72, pages 1 - 30. Elsevier, 2012. ISSN 0169-023X.
[IF: 1.422, 5-Year IF: 1.444]

Problém 2. Definice operací a jejich propagace

Atomické operace

- Pro všechny konstrukty PIM i PSM



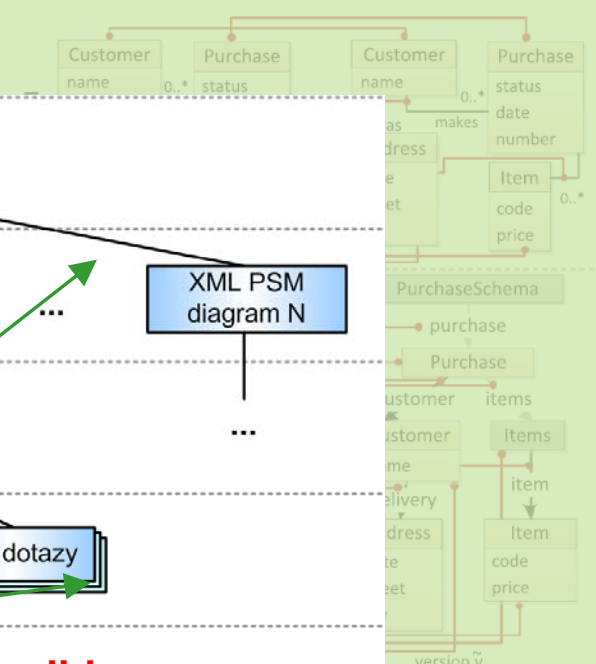
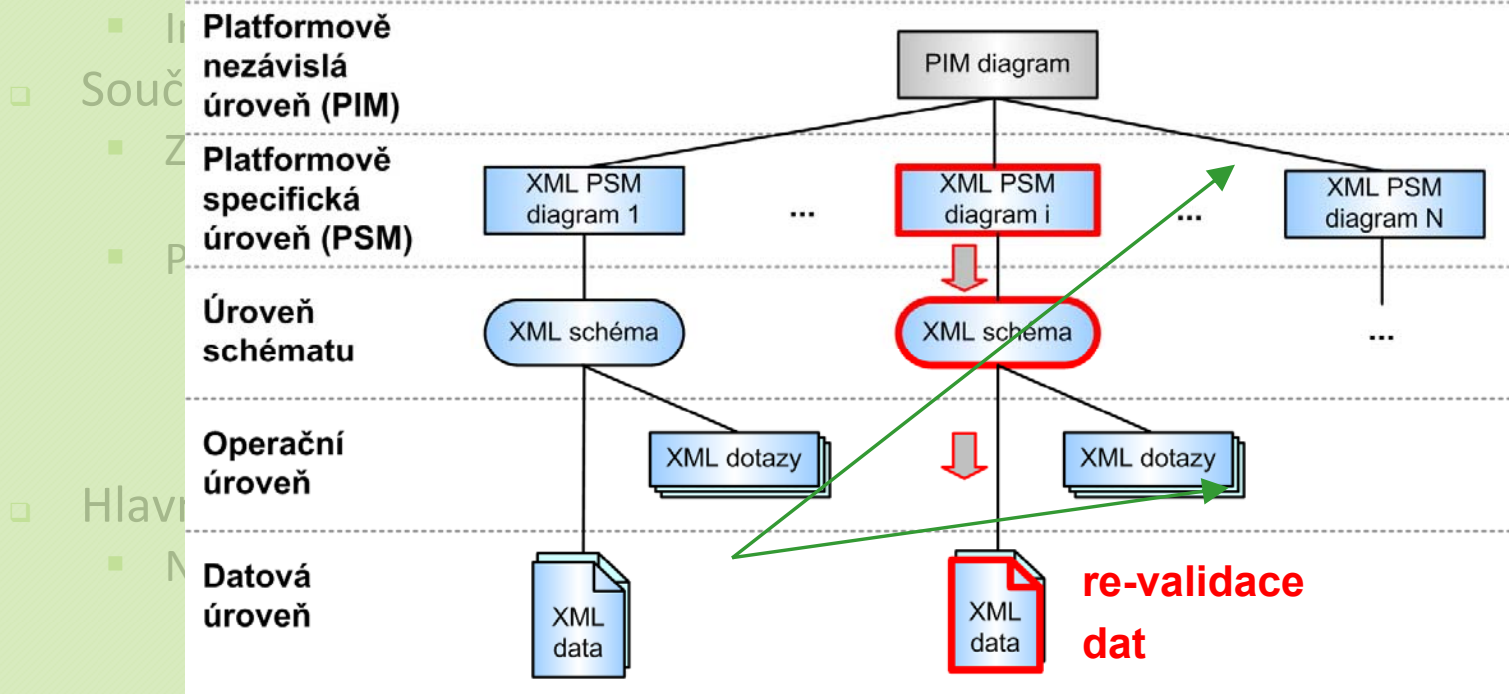
Složení

- Operační úroveň
- Datová úroveň

- Formální definice + důkaz korektnosti

Problém 3. Propagace změn do XML dokumentů

- Vstup: změny v XML schématu (PSM diagramu)
- Výstup: změny v XML dokumentech



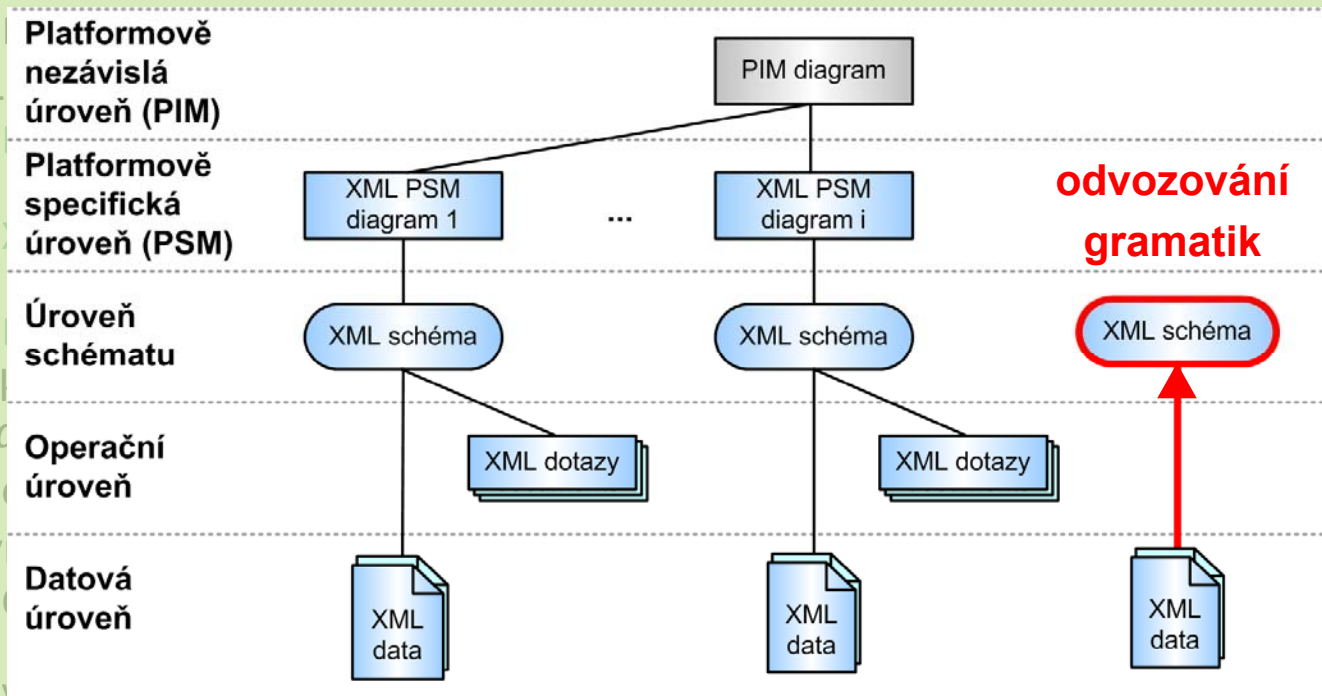
– Porovnáváme dvě libovolné verze schématu

- Automatické generování XSLT skriptu pro adaptaci (re-validaci) XML dat
 - Uchování sémantiky dat (vztah k PIM modelu)

Problém 4. Odvozování XML schémat (Reversní inženýrství)

Analýzy reálných XML dat (XML schémat + XML dokumentů):

- 52% XML dokumentů nemá XML schéma



využití datových vstupních informací

- XML operace, zastaralé schéma

Odvozování konstruktů jazyka Schematron

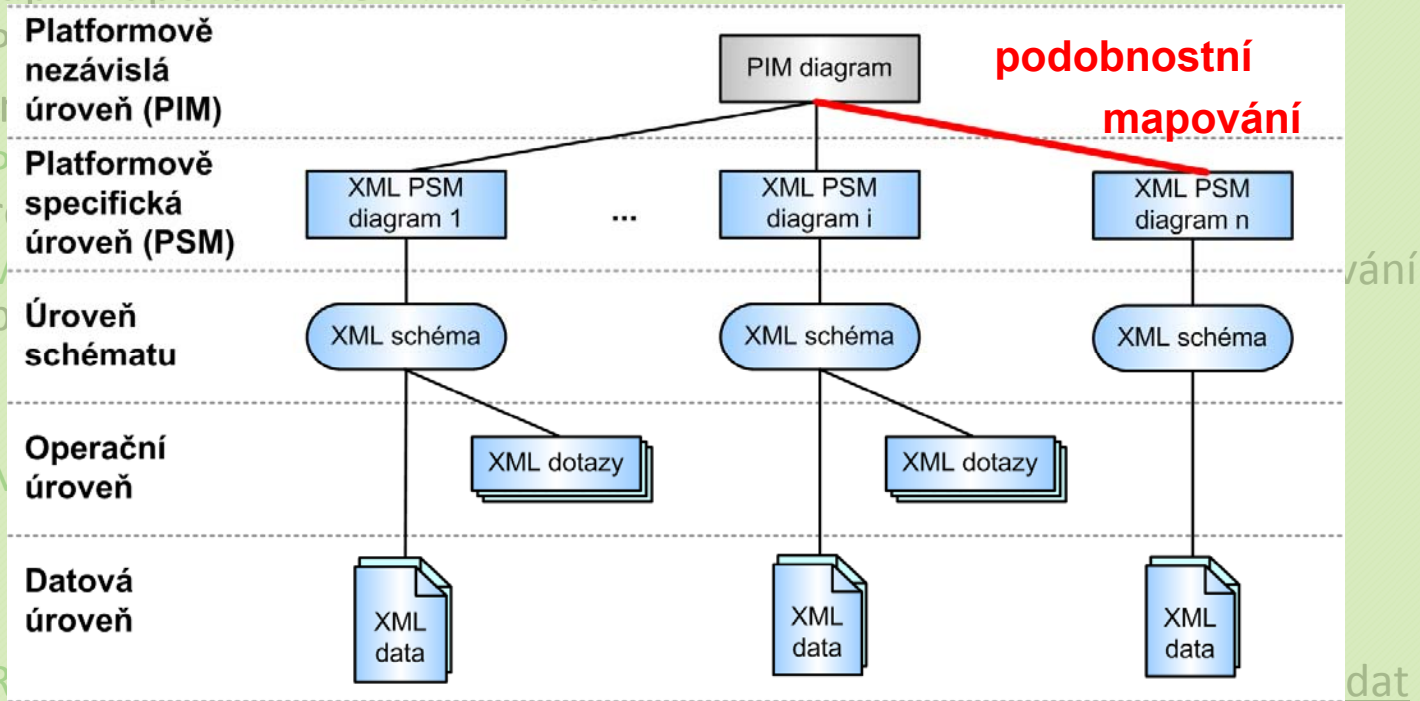
- Gramatiky → sada pravidel v XPath

kap. 4
habilitace

Mlynkova, I. - Necasky, M.: **Heuristic Methods for Inference of XML Schemas: Lessons Learned and Open Issues.** *Informatica*. IOS Press, 2012. ISSN 0868-4952. (In Press.)
[IF: 1.627, 5-Year IF: 1.074]

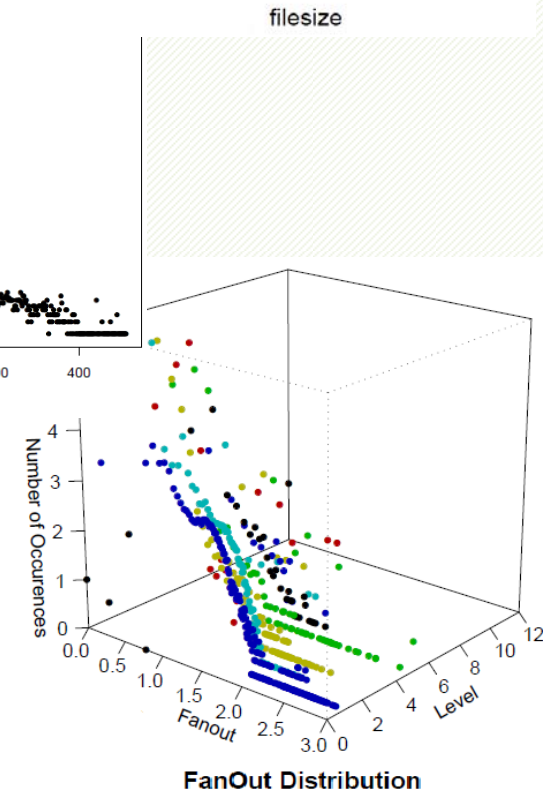
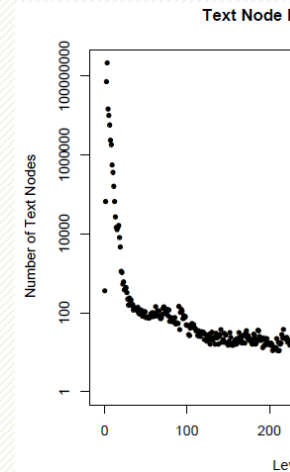
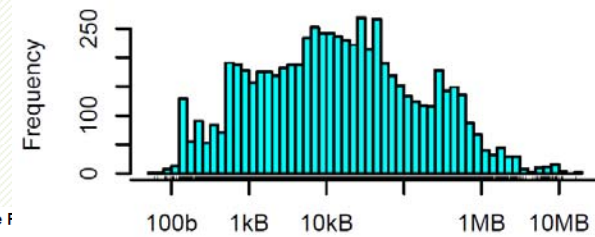
Problém 5. Mapování PSM na PIM (Reversní inženýrství)

- Vstup: PIM a PSM schéma
- Výstup: mapování mezi PIM a PSM



Problém 6: Analýzy reálných XML dat

- Analýzy reálných XML dat – hlavní přínosy:
 - Potvrzeno: průměrná hloubka XML dokumentů < 6
 - Klasická optimalizace
 - Vyvráceno: rekurze / smíšený obsah nejsou časté
 - Klasifikace dle složitosti struktury
 - Vyvráceno: XML schémata se téměř nepoužívají
 - Klasifikace využití dle typu dat
 - ...
- Analyzer** – modulární systém pro analýzy reálných (XML) dat
 - Cíl: opakovatelnost, jednoduchost, rozšiřitelnost
 - Systém nabízí: crawling dat, opravy chyb, analýzy dat, vizualizace výsledků
 - Analýzy XML dotazů



Mlynkova, I. - Toman, K. - Pokorný, J.: **Statistical Analysis of Real XML Data Collections**. *COMAD '06*, pages 20 - 31, New Delhi, India, December 2006. McGraw-Hill Publishing, 2006. ISBN 0-07-063374-6.

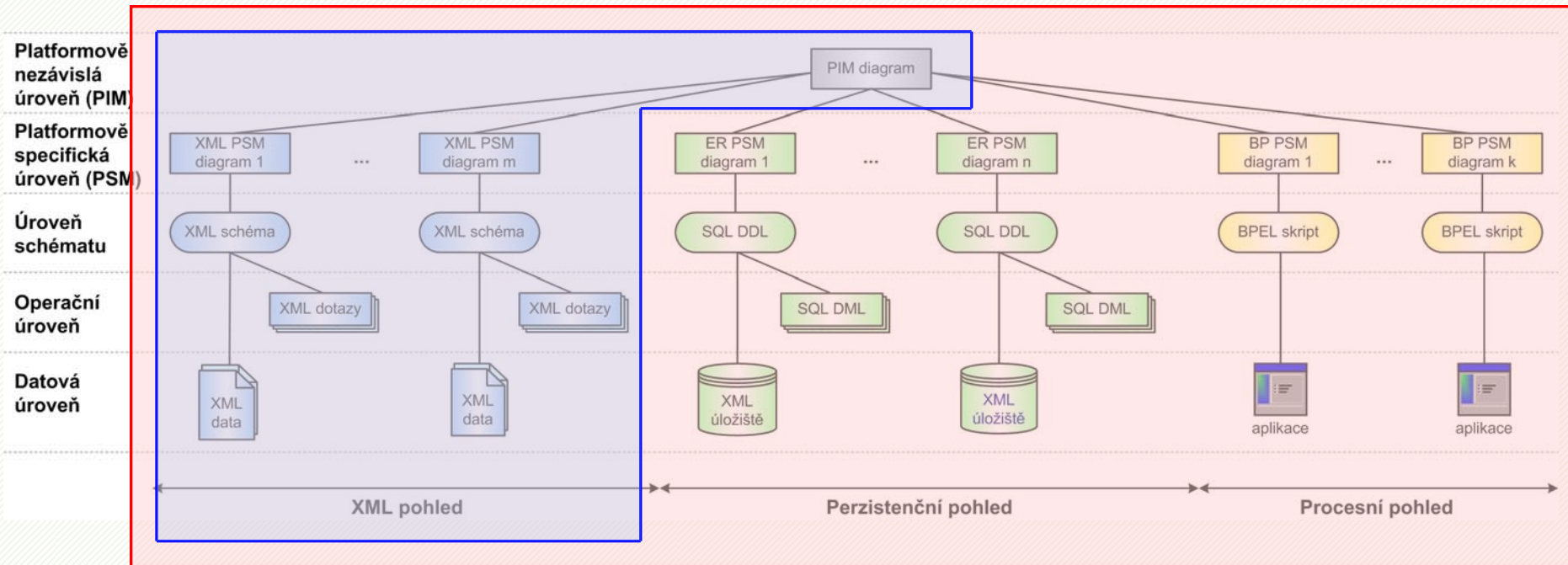
The Best Student Paper Award

kap. 6 habilitace

Starka, J. - Svoboda, M. - Sochna, J. - Schejbal, J. - Mlynkova, I. - Bednarek, D.: **Analyzer - A Complex System for Data Analysis**. *The Computer Journal*, volume 55, issue 5, pages 590 - 615. Oxford University Press, 2012. ISSN 0010-4620.

[IF: 0.785, 5-Year IF: 0.943]

Evoluce a adaptabilita komplexních XML aplikací (a nejen XML)



XCASE



Necasky, M. - Klimek, J. - Maly, J. - Mlynkova, I.: **Evolution and Change Management of XML-based Systems**. *Journal of Systems and Software*, volume 85, issue 3, pages 683–707. Elsevier, February 2012. ISSN 0164-1212.

[IF: 0.836, 5-Year IF: 1.117]

- Navrženo v úvodním článku
- Realizováno později

Současná práce a další plány

- Současné rozšiřování:
 - Propagace do **relačního modelu**
 - Množina stromů → množina relací
 - Propagace do **business proces modelu**
 - Model dat → model toho, jak a kde se s daty pracuje
- Hlavní plán: propagace do grafových dat
 - Stromy → grafy
 - Use-cases:
 - **Grafové databáze** \subset NoSQL databáze
 - Nová přednáška: Big Data management a NoSQL databáze (NDBI040)
 - **Linked Data**
 - TAČR INTLIB, FP7 LOD2 (XRG doktorandi, pod VŠE)
 - Podané granty:
 - GAČR – VŠB TU Ostrava
 - BMBF (Ministerstvo školství a výzkumu, SRN) – grant podporující přípravu grantů pro Horizont 2020
 - Georg-August-Universität, Göttingen
 - Institute Mihailo Pupin, Belgrade
 - Webdata Solutions GmbH, Leipzig

Děkuji za pozornost

Poděkování:

Členům týmu **XRG** (J. Pokorný, M. Nečaský, J. Klímek, J. Malý, J. Stárka, M. Svoboda, T. Knap, M. Polák)

Partnerským týmům v projektech **GAČR 201/06/0756** a **201/09/0990**
(FEL ČVUT, FEI VŠB TUO)

Partnerským týmům v projektu **TAČR TA02010182** (ÚFAL MFF UK, Sysnet s.r.o.)

Dr. Eric Pardede, Department of Computer Science & Computer Engineering, La Trobe University, Melbourne, Australia

Dr. Sherif Sakr, National ICT Australia (NICTA) Research Centre of Excellence, ATP lab, Sydney, Australia

Podíly na předkládaných výsledcích

□ Člen týmu

■ Témata:

- Návrh a formalizace pěti-úrovňového systému a souvisejících operací ([Problém 2](#))
 - Pozn. Problém 1 není vzhledem k menšímu podílu zahrnut v habilitační práci.
- Propagace změn do XML dat ([Problém 3](#))
 - Vedoucí diplomové práce

- Úloha: spolupráce na společném týmovém návrhu, spolupráce na přípravě vybraných částí publikací

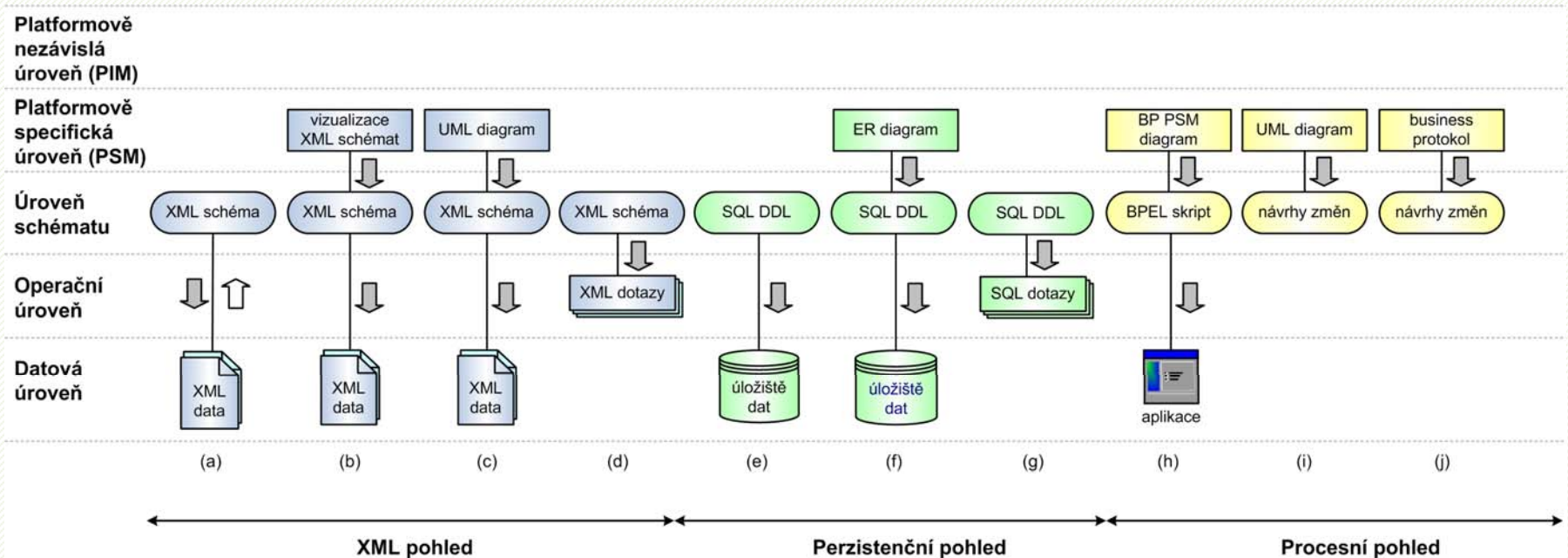
□ Klíčová úloha

■ Témata:

- Odvozování XML schémat – regulárních stromových gramatik ([Problém 4](#))
 - Několik publikací + modulární, rozšiřitelný odvozovací systém *jinfer*
- Mapování PSM na PIM ([Problém 5](#))
 - Vybrané podobnostní metody (v XRG vzniklo více návrhů)
- Propagace změn do XML operací
- Evoluce relačního modelu
- Analýzy reálných XML dat ([Problém 6](#))

- Úloha: vedoucí SW projektů / diplomových prací, školitelka

Související práce



- ❑ Jednotlivé případy se řeší, ale pouze částečně a v omezeném kontextu
 - Komplexní řešení (více formátů, více schémat, propagace do operací) chybí
- ❑ Řešení jsou často nepřesná → nefunkční
 - Bez formálního základu