



MINISTERSTVO VNITRA
ČESKÉ REPUBLIKY

Metodika tvorby XML schémat v oblasti informačních systémů veřejné správy

verze 1.01

březen 2009



Úvod

V současné době se stále více pozornosti v oblasti výpočetní techniky obrací směrem k distribuovaným výpočetním systémům. Tmelem, držícím takové systémy pohromadě, musí být spolehlivý a sofistikovaný formát přenosu dat. Toto postavení si postupně vydobyl formát XML [7] jako formát, nezávislý na platformě, síťovém prostředí a programovém vybavení.

XML v současné době je spíše než pouhý formát přenosu dat ucelený soubor technologií, komplexně postihující většinu aspektů přenosu dat mezi informačními systémy. Tyto technologie jsou vyjádřeny formou standardů, publikovaných standardizačními organizacemi (W3C, ISO, OASIS, IETF).

V oblasti XML je jedním z nejdůležitějších standard XML Schema [9]. Schémata definují pravidla, podle kterých se řídí struktura i obsah XML dokumentů. Schéma identifikuje třídu XML dokumentů, které vyhovují danému schématu. Můžeme také říci, že pro danou třídu XML dokumentů schéma definuje *datový slovník*. Schémata jsou nástroj, velmi usnadňující dosažení hladkého toku informací mezi všemi účastníky komunikace. Jsou materializací shody ve věci syntaxe (formátu) a sémantiky (významu) přenášených informací.

Ne všechny odkazy na normativní dokumenty, citované v tomto metodickém pokynu, jsou uvedeny v poslední verzi. Především v oblasti XML technologií (standards W3C, OASIS) je to způsobeno tím, že nejnovější standardy většinou nemají rozšířenu technologickou podporu a v mnoha případech jejich implementace v praxi je nejistá. Proto jsou citovány především stabilní, rozšířené a vyzkoušené standardy s širokou technologickou podporou.

Tento metodický pokyn má charakter doporučení.



Obsah

1. Předmět metodiky	5
1.1. <i>Struktura dokumentu</i>	5
1.2. <i>Cílová skupina uživatelů</i>	5
1.3. <i>Shoda s metodickými pokyny.....</i>	5
2. Odkazy.....	6
3. Vymezení pojmů.....	9
3.1. <i>Použité pojmy</i>	9
3.2. <i>Použité zkratky</i>	12
3.3. <i>Konformita s termíny RFC 2119.....</i>	13
4. Úvodní charakteristika XML schémat ISVS.....	14
4.1. <i>Základní datové typy ISVS.....</i>	14
4.2. <i>Referenční schémata, datové slovníky a schémata zpráv.....</i>	15
4.3. <i>Datově a textově orientovaná schémata.....</i>	16
5. Pravidla a doporučení návrhu, vývoje a správy XML schémat ISVS.....	17
5.1. <i>Návrh schémat.....</i>	17
5.1.1. <i>Jazyk schémat.....</i>	17
5.1.2. <i>Modelování dat.....</i>	18
5.1.2.1. <i>Použití UML k modelování dat.....</i>	18
5.1.2.2. <i>Modelování relačních dat</i>	18
5.1.3. <i>Design schémat</i>	19
5.1.3.1. <i>Struktura XML schémat</i>	19
5.1.3.2. <i>Použití jmenných prostorů</i>	19
5.1.3.3. <i>Formát jmenných prostorů</i>	20
5.1.3.3.1. <i>Konvence pro vytváření URN.....</i>	21
5.1.3.3.2. <i>Konvence pro vytváření URL</i>	22
5.1.3.4. <i>Použití prefixů jmenných prostorů.....</i>	22
5.1.3.4.1. <i>Použití známých prefixů.....</i>	22
5.1.3.4.2. <i>Volitelnost prefixů.....</i>	22
5.1.3.4.3. <i>Duplicitní deklarace prefixů.....</i>	23
5.1.3.5. <i>Použití elementFormDefault a attributeFormDefault kvalifikátorů.....</i>	24
5.1.3.6. <i>Globální deklarace elementů.....</i>	24
5.1.3.7. <i>Lokální deklarace atributů.....</i>	25
5.1.3.8. <i>Datové typy vs. elementy</i>	25
5.1.3.9. <i>Elementy vs. atributy.....</i>	26
5.1.3.10. <i>Použití referencí uvnitř XML dokumentů</i>	27
5.1.3.11. <i>Číselníky a jejich indikace</i>	28
5.1.3.12. <i>Hodnoty default a fixed.....</i>	29
5.1.3.13. <i>Omezení použití datových typů.....</i>	29
5.1.3.14. <i>Komentáře ve schématech.....</i>	30
5.1.3.15. <i>Usnadnění opakovaného použití schémat</i>	30
5.1.4. <i>Design komponent schémat</i>	30



5.1.4.1.	Pojmenovací konvence	30
5.1.4.1.1.	Jazyk a gramatický tvar jmen	30
5.1.4.1.2.	Case konvence	31
5.1.4.1.3.	Zkratky a akronymy	32
5.1.4.1.4.	Povolené znaky	32
5.1.4.1.5.	Jmenné konvence dle ČSN ISO/IEC 11179	33
5.1.4.1.6.	Jména lokálních elementů a atributů	33
5.1.4.1.7.	Jména elementů zpráv	34
5.1.4.1.8.	Jména jednoduchých a složených typů XML	35
5.1.4.2.	Použití základních datových typů ISVS	35
5.1.4.3.	Opakované použití ISVS typů a elementů	35
5.1.4.4.	Podpora dědičnosti typů	36
5.1.4.5.	Smíšený obsah elementů	36
5.1.4.6.	Reprezentace dat pomocí kódů nebo textu	36
5.1.5.	Dokumentace schémat a instancí XML dokumentů	37
5.1.5.1.	Dokumentace pomocí standardu Dublin Core	37
5.1.5.2.	Verzování schémat	41
5.1.5.3.	Data a metadata závislá na platformě a aplikaci	42
5.1.5.4.	Elektronický podpis schémat	42
5.1.5.5.	Použití nedokumentovaných schémat	42
5.1.6.	Doplňková validace	43
5.1.7.	Schémata a zprávy	44
5.2.	Vývoj a správa schémat	44
5.2.1.	Správce schématu, správce dat	44
5.2.2.	Publikace schémat	45
5.2.3.	Proces tvorby schémat, vazby na procesy přenosu dat	45
5.2.4.	Životní cyklus vývoje schémat	46
5.2.4.1.	Podpora starších verzí schémat	47
5.2.4.2.	Metodická podpora životního cyklu schémat	48
6.	Průvodce vývojem schématu	49
6.1.	Jednotlivé kroky vývoje schémat zpráv	49
7.	Změny	51
7.1.	Změny oproti předchozí verzi	51
7.2.	Změnové řízení	51



1. Předmět metodiky

Tento metodický materiál se podrobněji věnuje implementaci XML schémat v prostředí informačních systémů veřejné správy (dále jen „ISVS“). Je souhrnem pravidel a doporučení („Best Practices“) pro design schémat.

1.1. Struktura dokumentu

Tento dokument je členěn do několika oddílů. Nejdůležitější oddíl – *Pravidla a doporučení návrhu, vývoje a správy XML schémat ISVS 5* – popisuje různé aspekty cyklu návrhu, vývoje a údržby XML schémat. Je dále dělen na pododdíly:

- **Návrh schémat 5.1** poskytuje průvodce pro návrh a design schémat a komponent schémat, pro použití metadat ve schématech a pro dokumentaci schémat. Je *statickým* popisem požadovaných vlastností schémat.
- **Vývoj a správa schémat 5.2** popisuje požadavky na cyklus návrhu, vývoje a správy schémat jako na *dynamický* proces, probíhající v čase.

Každá část výše jmenovaných pododdílů je pak dělena do tří odstavců:

- **Pravidla** stanovují shrnutí požadavků nebo doporučení v dané problémové oblasti.
- **Vysvětlení** doplňují *Pravidla* o základní informace, proč byla příslušná pravidla přijata.
- **Příklady** jsou volitelnou částí. Mohou uvádět nezávazné příklady použití *Pravidel*.

1.2. Cílová skupina uživatelů

Cílovou skupinou uživatelů tohoto dokumentu jsou především vývojáři XML řešení v rámci ISVS. Měli by zde nalézt odpovědi na otázky návrhu struktury, vytvoření vhodného designu a návrhu spolehlivého modelu údržby XML schémat ISVS.

Předpokládá se, že čtenáři tohoto dokumentu mají znalosti XML technologií, a to zvláště XML Schema, přinejmenším na mírně pokročilé úrovni.

1.3. Shoda s metodickými pokyny

XML schémata ISVS jsou v souladu s touto metodikou, jestliže splňují podmínky uvedené v kapitole 5 této metodiky. Shoda s těmito metodickými pokyny není požadována retroaktivně. Existující řešení nejsou touto metodikou dotčena; v dalších verzích konkrétních řešení však bude žádoucí přechod k designu, které bude s touto metodikou v souladu.



2. Odkazy

[1]

Zákon č. 365/2000 Sb., o informačních systémech veřejné správy a o změně některých zákonů. <http://www.mvcr.cz/clanek/legislativa-zakon-c-365-2000-sb-o-informacnich-systemech-verejne-spravy.aspx>

[2]

ČSN ISO/IEC 11179-5: 1997

Informační technologie – Specifikace a normalizace datových prvků – Část 5: Identifikační principy a principy tvorby názvů datových prvků.

URL: <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=1677>

[3]

ČSN 97 4001-1: 1997

Výměna dat – Obecná struktura popisu datových prvků – Část 1: Popis datových prvků

[4]

RDF – Resource Description Framework (RDF) Model and Syntax Specification. *W3C Recommendation 22 February 1999*. URL: <http://www.w3.org/TR/REC-rdf-syntax/>

[5]

Key words for use in RFCs to Indicate Requirement Levels. *S. Bradner, IETF Network Working Group, March 1997*. URL: <http://www.ietf.org/rfc/rfc2119.txt>

[6]

XHTML™ 1.0: The Extensible HyperText Markup Language. A Reformulation of HTML 4 in XML 1.0. *W3C Recommendation 26 January 2000*

URL: <http://www.w3.org/TR/xhtml1>

[7]

Extensible Markup Language (XML) 1.0 (Second Edition). *W3C Recommendation 6 October 2000*.

URL: <http://www.w3.org/TR/REC-xml>

[8]

Namespaces in XML. *World Wide Web Consortium 14-January-1999*

URL: <http://www.w3.org/TR/REC-xml-names>

[9]

XML Schema Part 0: Primer. *W3C Recommendation, 2 May 2001*

URL: <http://www.w3.org/TR/xmlschema-0/>

[10]

XML Path Language (XPath) Version 1.0. *W3C Recommendation 16 November 1999*

URL: <http://www.w3.org/TR/xpath>

[11]

Dynamic Delegation Discovery System (DDDS). Part One: The Comprehensive DDDS. *IETF RFC 3401, October 2002*. URL: <http://www.ietf.org/rfc/rfc3401.txt?number=3401>

[12]

Uniform Resource Names (URN) Namespace Definition Mechanisms. *IETF RFC 3406, October 2002*. URL: <http://www.ietf.org/rfc/rfc3406.txt?number=3406>

[13]



URN Syntax. *IETF RFC 2141, May 1997.*

URL: <http://www.ietf.org/rfc/rfc2141.txt?number=2141>

[14]

XML-Signature Syntax and Processing. *W3C Recommendation, 12 February 2002.*

URL: <http://www.w3.org/TR/xmlsig-core/>

[15]

Dare Obasanjo: W3C XML Schema Design Patterns: Avoiding Complexity. *XML com, 20. November 2002.* URL: <http://www.xml.com/pub/a/2002/11/20/schemas.html>

[16]

Recommended XML Namespace for Government Organizations. *Logistics Management Institute, March 2003.*

URL: http://xml.gov/documents/completed/lmi/GS301L1_namespace.pdf

[17]

Draft Federal XML Developer's Guide. *U.S. Federal CIO Council, Architecture and Infrastructure Committee, XML Working Group, April 2002.*

URL: http://xml.gov/documents/in_progress/developersguide.pdf

[18]

DCMI Metadata Terms. DCMI Recommendation. *Dublin Core Metadata Initiative, 4-March-2003.* URL: <http://dublincore.org/documents/dcmi-terms/>

[19]

Guidelines for implementing Dublin Core in XML. DCMI Recommendation. *Dublin Core Metadata Initiative, 2-April-2003.*

URL: <http://dublincore.org/documents/2003/04/02/dc-xml-guidelines/>

[20]

Unified Modeling Language (UML). *Verze 1.04 Object Management Group 2001.*

URL: <http://www.omg.org/technology/documents/formal/uml.htm>

[21]

UN/CEFACT – ebXML Core Components. Technical Specification. *11 December 2002, Version 1.90.* URL: <http://xml.coverpages.org/CCTSv190-2002.pdf>

[22]

XSL Transformations (XSLT) Version 1.1. *W3C Working Draft 24 August 2001*

URL: <http://www.w3.org/TR/xslt11>

[23]

XML Schemas: Best Practices. *The Mitre Corporation, 17 February 2003.*

URL: <http://www.xfront.com/BestPracticesHomepage.html>

[24]

Position Paper: Code Lists. *OASIS, Proposal 09, 28 May 2002.*

URL: <http://www.oasis-open.org/committees/ubl/ndrsc/pos/p-maler-codelists-09.pdf>

[25]

Bob du Charme, John Cowan: Make Your XML RDF-Friendly. *XML com, 30 October 2002.*

URL: <http://www.xml.com/pub/a/2002/10/30/rdf-friendly.html>

[26]

Simple Object Access Protocol (SOAP) 1.1 W3C Note 08 May 2000.

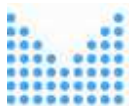
URL: <http://www.w3.org/TR/SOAP>



[27]

Uniform Resource Locators (URL). *IETF RFC 1738, December 1994.*

URL: <http://www.ietf.org/rfc/rfc1738.txt?number=1738>



3. Vymezení pojmů

3.1. Použité pojmy

Atribut – (obecně) charakteristika objektu nebo entity. Pro oblast XML strukturální XML konstrukt. Sestává se z páru jméno - hodnota, odděleného rovnítkem. Může se vyskytovat pouze uvnitř startovního tagu XML elementu.

Autorizace – udělení práv, které zahrnuje udělení přístupu na základě přístupových práv.

Autentizace – akt ověřování prohlašované identity dané entity.

CamelCase konvence – způsob konstrukce jmen, kdy každé nové slovo ve jménu složeném z více slov začíná velkým písmenem.

Cílový jmenný prostor – vlastní jmenný prostor daného XML schématu.

Číselník – seřazený seznam dvojic kód – hodnota. Kód je unikátní v rámci celého číselníku.

Datová integrita – schopnost zachovat přesnost, správnost a kompletnost dat bez ohledu na některé případné operace, prováděné s daty.

Datový prvek - (obecně) jednotka dat, která je v daném kontextu považována za nedělitelnou; (podle EDIFACT) jednotka dat, pro kterou jsou stanoveny identifikace, popis a formát hodnoty.

Datový slovník – (pro oblast XML) XML schéma, obsahující deklarace XML struktur (elementů, atributů, složených a jednoduchých typů), popisující datové entity z jedné problémové domény. Využívá deklarací XML konstruktů z referenčních schémat.

Datový typ – (obecně) kategorie používaná pro klasifikaci souboru písmen, číslic a/nebo symbolů pro zobrazení hodnot datového prvku, založená na operacích, které lze s datovým prvkem provádět (ČSN 974001-1)

Datový typ – (pro oblast XML) deklarace třídy XML struktur.

Datový typ jednoduchý - deklarace třídy XML struktur, definující pouze povolenou hodnotu. V XML schématech definováno pomocí klauzule `<xs:simpleType>`. Datový typ jednoduchý je možno aplikovat jak na elementy, tak na atributy.

Datový typ složený - deklarace třídy XML struktur, sestávající se z rodičovského elementu, obsahujícího dceřinné elementy a/nebo atributy. V XML schématech definováno pomocí klauzule `<xs:complexType>`. Datový typ složený je možno aplikovat pouze na elementy.

Dceřinný element – element, hierarchicky vnořený do jiného (rodičovského) elementu

Defaultní jmenný prostor – jmenný prostor, který pro dané XML schéma byl definován bez prefixu jmenného prostoru. Obvykle je za defaultní jmenný prostor zvolen cílový jmenný prostor schématu.

Doména – též problémová doména – relativně ohraničená oblast reality, popsitelná pomocí datového modelování (například účetnictví, identifikace osoby).



DTD – formát popisu XML dokumentu. Jedná se o popis struktury XML dokumentu v proprietární non-XML syntaxi. Je dlouhodobě používán již od doby SGML.

Element - strukturální XML konstrukt. XML element se skládá ze startovního tagu, koncového tagu a informace, nesené mezi tagy neboli obsahu elementu.

Globální element (atribut) – element či atribut, deklarovaný v XML schématu takovým způsobem, že je opakovaně samostatně použitelný v jiných schématech pomocí importu schématu.

Hierarchický model – datový model, kdy jednotlivé entity zahrnuté v modelu jsou uspořádány v přirozené stromové struktuře proměnlivé hloubky.

Identifikační schéma – sada jednoznačných identifikátorů, popisující obvykle nějakou oblast reálného světa.

Import XML schémat – metoda opakovaného využití XML struktur (elementů, atributů i datových typů), definovaných v jednom schématu a příslušejících jednomu jmennému prostoru, v jiném schématu.

Informační systém – funkční celek nebo jeho část zabezpečující cílevědomou a systematickou informační činnost. Každý informační systém zahrnuje data, která jsou uspořádána tak, aby bylo možné jejich zpracování a zpřístupnění, a dále nástroje umožňující výkon informačních činností.

Informační systémy veřejné správy (ISVS) jsou souborem informačních systémů, které slouží pro výkon veřejné správy. Jsou jimi i informační systémy zajišťující činnosti podle zvláštních zákonů.

Instance XML – vlastní XML dokument, přenášející data.

Klientská aplikace - aplikace, žádající službu o nějakou akci (poskytnutí dat, vložení zaslaných dat apod.). Klientská aplikace, na rozdíl od služby, pouze vznáší požadavky na službu a nenaslouchá ostatním účastníkům komunikace. Při komunikaci se službou musí striktně dodržovat pravidla, stanovená službou.

Komunikace – přenos informací mezi dvěma či více stranami.

Kvalifikované entity – takové entity uvnitř XML dokumentů (atributy, elementy či datové typy), které jsou označeny prefixem jmenného prostoru.

Lokální element (atribut) - element či atribut, deklarovaný v XML schématu takovým způsobem, že není samostatně opakovaně použitelný v jiných schématech pomocí importu schématu, je však možno jej opakovaně použít v rámci rodičovského elementu.

Metadata – data, popisující jiná data. Typickým příkladem jsou data XML schémat, popisující data instancí XML dokumentů.

Metainformační systém - informační systém, který uchovává mimo jiné i popisy datových a funkčních rozhraní všech aplikací a služeb dostupných prostřednictvím Referenčního rozhraní.

Model – zjednodušený a schematizovaný popis reality.



Nekvalifikované entity - takové entity uvnitř XML dokumentů (atributy, elementy či datové typy), které nejsou označeny prefixem jmenného prostoru.

Parser – software, schopný načíst XML soubor a vystavit aplikační programové rozhraní pro práci nad tímto XML souborem.

Referenční integrita – způsob udržení konzistence dat v relačním modelu dat pomocí relací primární klíč – cizí klíč.

Referenční rozhraní – souhrn právních, technických, organizačních a jiných opatření vytvářejících jednotné integrační prostředí ISVS, které poskytuje kvalitní soustavu společných služeb, včetně služeb výměny oprávněně vyžadovaných informací mezi jednotlivými informačními systémy orgánů veřejné správy a dalšími subjekty, a to i se systémy mimo Českou republiku.

Referenční schéma - XML schéma, obsahující deklarace obecných XML struktur (elementů, atributů, složených a jednoduchých typů), popisující datové entity společné více problémovým doménám.

Relační model - datový model, kdy jednotlivé entity zahrnuté v modelu jsou uspořádány v tabulkách a sloupcích tabulek metodou normalizace dat. Vztahy mezi entitami jsou postiženy pomocí relací primární klíč – cizí klíč.

Rodičovský element – element, nadřazený v hierarchii XML dokumentu sledované entitě (elementu, atributu apod.).

Sémantika – význam, smysl či faktický obsah nějakého konceptu (např. XML elementu).

Schéma zprávy – typ XML schématu, popisující strukturu konkrétních přenášených zpráv ISVS. Využívá deklarací XML konstruktů z datových slovníků a referenčních schémat.

Služba - obecně aplikace, provádějící na požádání nějakou akci (např. poskytující data z registru, přebírající zasláná data a vkládající je do databáze apod.). Charakteristickou vlastností služby je (zpravidla permanentní) naslouchání požadavkům klientů a plnění jejich požadavků.

Správce dat – Subjekt, zodpovědný za publikaci schématu a za dostupnost schématu.

Správce schématu – Subjekt, zodpovědný za vývoj schématu a za faktickou správnost schématu včetně shody s touto metodikou tvorby schémat.

Syntaxe – způsob zápisu určité informace. Např. sémanticky identický popis datové struktury může být zapsán v syntaxi XML Schema nebo v syntaxi relačního modelu.

UML – Unified Modelling Language – jazyk, vytvořený modelování rozsáhlých oblastí reálného světa – např. datových struktur, objektů, procesů a podobně. Je používán při vývoji XML schémat pro syntakticky neutrální způsob zápisu.

XML dokument – dokument, obsahující data, značkováná v souladu se standardem XML.

XML Schema - formát popisu XML dokumentu. Jedná se o popis struktury XML dokumentu v XML syntaxi. V současné době nejrozšířenější formát pro deklaraci tříd XML dokumentů. Oproti DTD poskytuje výrazně vyšší možnosti detailního popisu struktur a omezení platnosti dat.



XPath – jazyk, kterým lze adresovat části XML dokumentu. Má proprietární syntaxi; vrací seznam částí XML dokumentu, vyhovujících dotazu. Poskytuje také základní funkce pro manipulaci s daty (řetězcovými, numerickými a logickými) v XML dokumentech.

XSLT – jazyk, umožňující pomocí jazyka XPath manipulace s částmi XML dokumentu. Je možno vytvářet jiné XML dokumenty transformací stávajících nebo např. generovat čistě textové výstupy. Jazyk XSLT používá XML syntaxi.

XSLT šablona – soubor se sadou XSLT instrukcí k provedení transformace XML dokumentu.

3.2. Použité zkratky

ČSN	Česká technická norma
DCMI	Dublin Core Metadata Initiative
DSDL	Document Schema Definition Language
DTD	Document Type Definition
ebXML	Electronic Business using eXtensible Markup Language
HTML	Hypertext Markup Language
IETF	Internet Engineering Task Force
IS	Informační systémy
ISDP	Informační systém o datových prvcích
ISO	International Organization for Standardization
ISVS	Informační systémy veřejné správy
MS ISVS	Metainformační systém informačních systémů veřejné správy
NID	Namespace Identifier
NSS	Namespace Specific String
OASIS	Organization for the Advancement of Structured Information Standards
RDDL	Resource Directory Description Language
RDF	Resource Description Framework
RFC	Request For Comments
SGML	Standard Generalized Markup Language
UBL	Universal Business Language
UML	Unified Modelling Language
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
URI	Uniform Resource Identifier



URL	Uniform Resource Locator
URN	Uniform Resource Name
W3C	World Wide Web Consortium
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language
XMI	XML Metadata Interchange
XML	Extensible Markup Language
XPath	XML Path Language
XSLT	Extensible Stylesheet Language for Transformations

3.3. Konformita s termíny RFC 2119

V celém dokumentu je v odstavcích **Pravidlo** používán termín „musí“ nebo „nesmí“ pro vyjádření závaznosti příslušného ustanovení, termín „bude“ nebo „nebude“ vyjadřuje prohlášení o závazném cíli nebo úmyslu, termíny „měl by“, „doporučeno“, „neměl by“ a „nedoporučeno“ vyjadřují doporučení a termíny „může“ a „volitelné“ vyjadřuje směr činnosti v rámci přípustných limitů definovaných tímto dokumentem. Výše uvedené termíny mohou být použity v kterémkoli z gramatických tvarů. Výše uvedená pravidla platí pouze pro slova, uvedená KAPITÁLKAMI.

Tyto termíny jsou interpretovány v souladu s anglickými originály termínů, uvedenými v IETF „Request For Comments“ č. 2119 [5].



4. Úvodní charakteristika XML schémat ISVS

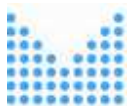
Schémata je možno logicky rozčlenit na několik skupin. Užitečné je dělení schémat do vrstev, které odpovídají míře obecnosti daného schématu.

4.1. Základní datové typy ISVS

Architektura schémat ISVS vychází z metodologie a modelu popsaného v publikaci „ebXML Core Components. Technical Specification“ [21]. Z této metodologie byl převzat model základních typů (Core Component Types). Základní typy vytvářejí primární vrstvu datových typů ISVS, ze kterých jsou pak odvozeny všechny deklarace datových typů ve schématech ISVS.

Existuje omezená množina základních datových typů ISVS:

Jméno typu	Popis základního typu	Atributy
CenaType	Počet peněžních jednotek ve specifikované či implicitní měně; volitelně spolu s doplňkovou informací o použité měnové jednotce.	menaKod menaCiselnikID
KodType	Znakový řetězec (písmena, číslice nebo symboly) které mohou z důvodů jasnosti a jazykové nezávislosti nahradit definitivní hodnotu nebo text; volitelně spolu s doplňkovými informacemi o údaji.	ciselnikID ciselnikSpravceID ciselnikVerzeID kodJmeno jazykKod
DatumCasType	Hodnota konkrétního bodu v časové ose volitelně spolu s doplňkovou informací o formátu údaje. Používá se pro datum a/nebo čas.	_
IdentifikatorType	Znakový řetězec jednoznačně identifikující instanci příslušného objektu od ostatních objektů v rámci jednoho identifikačního schématu; volitelně spolu s doplňkovými informacemi o údaji.	schemaID schemaSpravceID jazykKod
IndikatorType	Seznam dvou a pouze dvou hodnot, indikujících stavy jako zapnuto/vypnuto, pravda/nepravda apod. Je synonymem pro datový typ “Boolean”; volitelně spolu s doplňkovou informací o formátu údaje.	indikatorFormat
MereniType	Hodnota velikosti, hmotnosti, objemu apod., získaná pomocí procesu měření; volitelně spolu s doplňkovou informací o jednotce, v níž je údaj vyjádřen.	jednotkaKod jednotkaCiselnikID
MnozstviType	Hodnota množství nebo rozsahu; volitelně spolu s doplňkovou informací o jednotce, v níž je údaj vyjádřen.	jednotkaKod jednotkaCiselnikID
PocetType	Hodnota počtu; volitelně spolu s doplňkovou informací o jednotce, v níž je údaj vyjádřen.	jednotkaKod jednotkaCiselnikID



NumerickaHodnotaType	Numerická informace, získaná např. výpočtem nebo počítáním. Nevyžaduje nutně jednotku kvantity nebo jednotku měření.	–
BinarniDataType	Binární data kódovaná ve formátu base64 ; volitelně spolu s doplňkovou informací o formátu původních dat.	dataFormat souborJmeno encoding characterSet kodMIME
TextType	Znakový řetězec, obecně ve formátu volného textu i více slov; volitelně spolu s doplňkovou informací o jazyku.	jazykKod

Jak je zřejmé, pro základní datové typy byla převzata pouze metodologie ebXML. Vlastní schéma základních datových typů není nijak svázáno s implementací schémat ebXML např. v datovém slovníku UBL. Důvodem pro toto řešení je rozhodnutí nevázat řešení ISVS ČR na žádné externí entity, a to především z důvodu požadavku plné kontroly nad celým systémem datových typů ISVS.

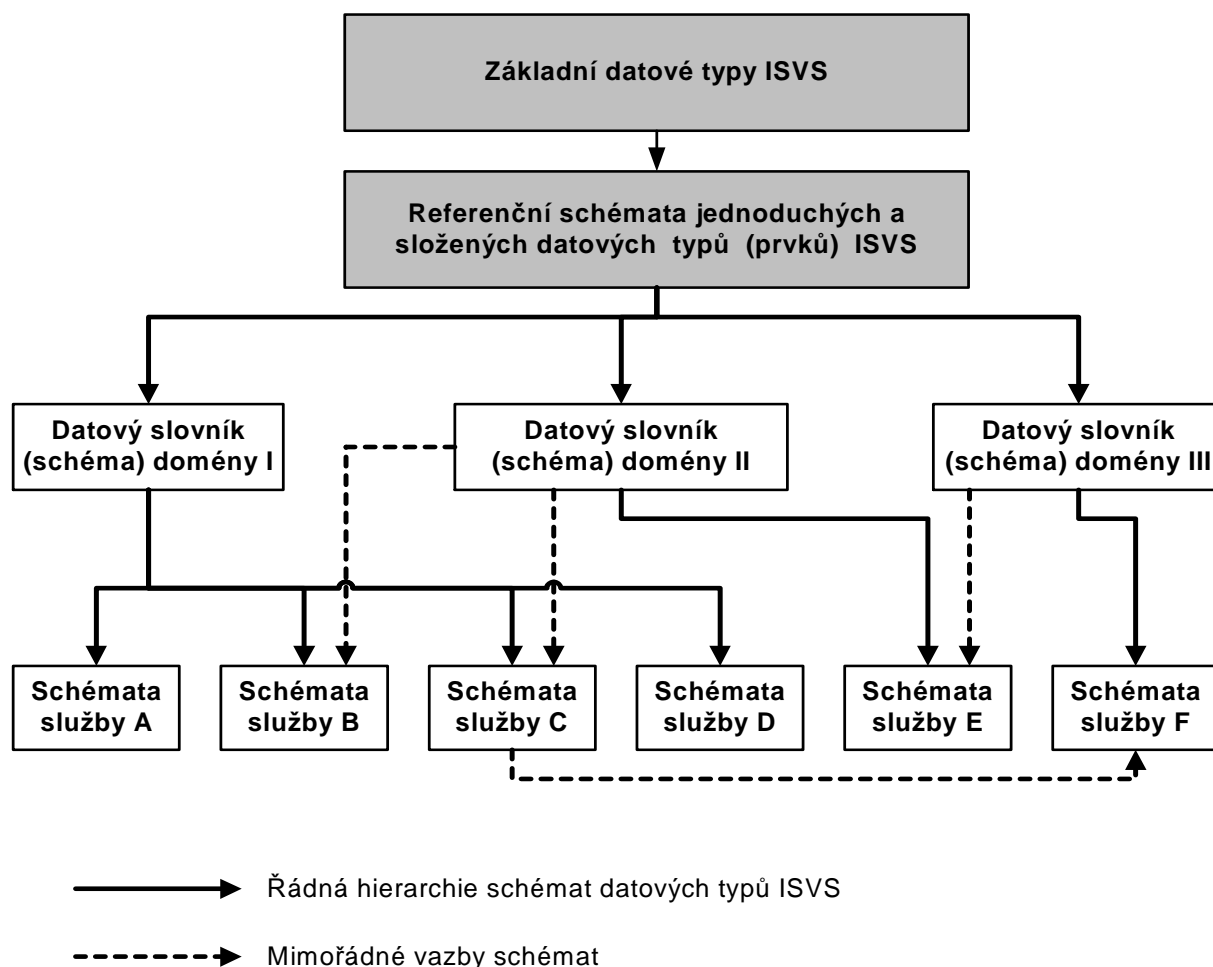
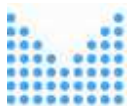
4.2. Referenční schémata, datové slovníky a schémata zpráv

Další vrstvu představují **referenční schémata** datových typů. Tato schémata popisují obecné datové prvky, používané obecně při komunikaci pomocí rozhraní ISVS. Všechny typy v referenčních schématech jsou odvozeny ze základních datových typů.

Nad nezbytnou vrstvou standardních schémat datových typů ISVS jsou vytvořena schémata (**datové slovníky**), představující standardizaci dat v jedné problémové oblasti (doméně).

Datové slovníky z problémové domény mohou využívat konkrétní služby, poskytující data v rámci této domény klientským aplikacím. Služby budou definovat datové typy, specifické pro jednotlivé služby, a skládat takové datové typy do konkrétních **schémat zpráv**. Zprávy jsou pak přenášeny pomocí komunikační infrastruktury ISVS.

Možné jsou i mimořádné vazby schémat, kdy např. schéma z jedné problémové domény využívá (importuje) vhodné datové typy, definované schématem v jiné problémové doméně. Všechna schémata v rámci ISVS pak musejí využívat standardní datové typy.



OBR.1 – Skupiny XML schémat, možné vazby schémat

4.3. Datově a textově orientovaná schémata

Při vytváření schémat je nutno přihlížet k některým skutečnostem, které mají vliv na konečnou podobu schémat. V první řadě je třeba odlišovat *datově* a *textově* orientovaná schémata, aplikace i dokumenty.

Datově orientované XML aplikace mohou mít diametrálně odlišné požadavky na strukturu XML souborů oproti textově orientovaným aplikacím. Datově orientovaná schémata mohou být optimalizována například s ohledem na velikost přenášených dat a měla by splňovat striktní požadavky datové a případně i referenční integrity. Dokumenty nutně nemusejí být čitelné člověkem (*human-readable*), neboť s výjimkou vývojářů je nebudou uživatelé prohlížet.

Textově orientované XML dokumenty mají obvykle uživatele-člověka, který může editovat jejich obsah i bez specializovaných nástrojů pro editaci XML dokumentů. Jejich struktura musí být jasně srozumitelná především pro autory XSLT šablon, kteří prezentují data, obsažená v dokumentech.



5. Pravidla a doporučení návrhu, vývoje a správy XML schémat ISVS

Následující kapitola je přehledem pravidel implementace a vytváření skupin schémat (datových slovníků) v oblasti ISVS.

5.1. Návrh schémat

5.1.1. Jazyk schémat

Pravidla

Pro vytváření nových schémat ISVS MUSÍ být použito jazyku XML Schema. I v případech, kdy existuje dlouhodobá tradice použití existujících DTD, MĚLI BY vývojáři, zvláště pak datově orientovaných dokumentů, přejít k XML Schema. Vývojáři textově orientovaných dokumentů MOHOU pokračovat v použití DTD. Jazyky schémat jiné než XML Schema a DTD NESMĚJÍ být při komunikaci mezi ISVS používány.

V případě, kdy v budoucnosti některý z jazyků popisu formátu XML dokumentu dospěje do fáze standardu (W3C Recommendation), MŮŽE dojít k revizi tohoto pravidla.

Vysvětlení

DTD (*Document Type Definition*) je formát popisu dokumentu, pocházející ještě z éry SGML. Jeho výhodou je dlouhá tradice použití a obeznámenost XML komunity s tímto formátem. Proto ve formátu DTD existuje již značné množství vytvořených schémat. Nevýhodou je pak jednak odlišný jazyk pro vytváření DTD (DTD není XML dokument a není možno jej zpracovat pomocí parseru), jednak poměrně slabá možnost deklarace podrobnějších pravidel strukturování. DTD nepodporují deklaraci datových typů pro přenášená data, neumožňují použití jmenných prostorů a neumožňují import deklarací struktur a datových typů z jiných DTD.

XML Schema [9] je standard W3C. XML Schema nemá nedostatky DTD; jedná se o deklaraci typu dokumentu ve formátu dokumentu XML (je tedy čitelný parserem). Obsahuje množství vestavěných datových typů a snadno lze definovat i vlastní datové typy. Podporuje jmenné prostory a import dříve deklarovaných datových typů, proto je možno pro jeden dokument využít více schémat (modularita). Nejsilnější stránkou XML schéma je excelentní podpora datových typů, a to jak jednoduchých, tak složených.

Jazyky jiné než DTD a XML Schema v současné době nejsou stabilními standardy. Použití dalších jazyků schémat by také vnášelo do komunikace systémů ISVS komplikace v podobě nutnosti konformity rozhraní se všemi povolenými formáty. Proto není použití těchto jazyků v rámci ISVS povoleno.

Kandidátem na zařazení do seznamu povolených jazyků schémat je v současné době návrh standardu DSDL (Document Schema Definition Language). Standard DSDL by měl komplexně postihnout problematiku popisu i validace XML dokumentů.



5.1.2. Modelování dat

5.1.2.1. Použití UML k modelování dat

Pravidla

Při návrhu složených datových struktur JE DOPORUČENO nejprve navrhnout model, nezávislý na konečné syntaxi XML Schema. Tento model BY MĚL být ve formátu UML. Dokumentace datových slovníků a referenčních schémat, obsahujících složené struktury, BY MĚLA obsahovat UML model příslušných složených struktur. Dokumentace schémat zpráv MOHOU obsahovat UML modely zpráv.

Pokud bude dokumentace schémat obsahovat UML model, JE DOPORUČENO, aby tento model byl zapsán v syntaxi XMI.

Vysvětlení

Při návrhu datových struktur ISVS je vhodné nejprve vytvořit model, který nebude primárně orientován na konkrétní předpokládanou koncovou syntaxi (XML Schema, relační model). Výhodou tohoto přístupu je možnost opakované využitelnosti takových nezávislých modelů v budoucnosti, kdy by mohlo dojít ke změně cílové syntaxe schémat.

UML (Unified Modelling Language) je jazyk, vytvořený mimo jiné i pro návrh a popis složených datových struktur. Je široce používán při vývoji XML Schema.

Pro schémata s datovými modely s předpokládaným širokým využitím (referenční schémata, datové slovníky) je velmi vhodné do dokumentace přiložit UML modely deklarovaných struktur, přinejmenším z důvodu přehlednosti grafické reprezentace UML modelů. Pro schémata koncových zpráv je použití UML volitelné; předpokládá-li se však opakovaná použitelnost struktur i jinými informačními systémy, je vhodné UML modely zařadit.

5.1.2.2. Modelování relačních dat

Pravidla

Pro modelování struktur, přenášejících relační data, JE DOPORUČENO vytvořit datový model nezávislý na případné výchozí databázi. Pouze v případě proprietárních řešení bez předpokladu opakovaného využití navržených datových struktur MOHOU být data modelována v souladu s výchozí relační strukturou.

Vysvětlení

Pokud se pro navrhované datové struktury předpokládá široké opakované využití blíže neurčenou a víceméně neohrazenou komunitou uživatelů (např. pro datové slovníky či referenční schémata), je vhodné data modelovat „přirozeným“ objektovým způsobem. Vnucení relačního modelu, převzatého z výchozí databáze některého z účastníků, celé komunitě není žádoucí. Objektový či hierarchický model je samozřejmě vhodný také pro případy, kdy žádná výchozí databáze neexistuje.

Jiná situace nastane v případě, kdy jsou vyměňována data uvnitř relativně uzavřené skupiny uživatelů (aplikací). V tomto případě je možno (po společné dohodě účastníků komunikace)



navrhnout datový model pro přenášená data v souladu s relačním modelem výchozí databáze některého z účastníků.

5.1.3. Design schémat

5.1.3.1. Struktura XML schémat

Pravidla

Vývojáři ISVS BY NEMĚLI při návrhu XML schémat používat nepoužívaná, neobvyklá či příliš obtížně čitelná řešení.

Vysvětlení

Každý designér XML schémat v rámci ISVS by měl mít v první řadě na zřeteli cílovou skupinu uživatelů vyvíjeného schématu. Použití komplikovaných (i když elegantních a sofistikovaných) řešení na úkor čitelnosti a srozumitelnosti schématu není žádoucí. Pokud je nutno volit mezi řešením elegantním, ale obtížně čitelným a řešením primitivním, ale snadno dešifrovatelným (za předpokladu stejné kvality schémat s ohledem na schopnosti popisu struktury a validace dat), je lépe zvolit druhou variantu.

Dalším důvodem pro jistou primitivnost schémat je také parser od parseru rozdílná implementace standardu XML Schema, kde interpretace některých komplexních řešení se pro různé parsery liší. Z toho mohou také vyplynout i odlišné výroky o validitě schémat.

5.1.3.2. Použití jmenných prostorů

Pravidla

Všechna schémata ISVS BY MĚLA obsahovat atribut cílového jmenného prostoru `targetNamespace`. JE DOPORUČENO, aby cílový jmenný prostor byl shodný s defaultním jmenným prostorem schématu.

Vysvětlení

Tzv. *chameleon design* (design, kdy schémata nemají `targetNamespace`) je někdy používán pro schémata, u kterých se předpokládá zahrnutí do více koncových schémat s různými jmennými prostory. Takové schéma bez cílového jmenného prostoru se pak chová jako chameleon – datové typy v něm deklarované přebírají jmenný prostor příslušného importujícího schématu.

Problém nastává v okamžiku, kdy je třeba se uvnitř *chameleon* schématu odkazovat pomocí XPath výrazu použitého v `xs:key`, `xs:keyref`, a `xs:unique` omezeních. V případě, kdy takové schéma zahrneme do koncového schématu, XPath výraz selže.

Dalším negativem *chameleon* schémat je zvýšení pravděpodobnosti změny uvnitř komplexu schémat, které budou součástí výsledného jmenného prostoru, a tedy nutnost změn verzí schémat.

Nejzávažnějším argumentem proti použití *chameleon designu* je ale nutnost „jednovrstevné“ deklarace typů uvnitř *chameleon* schématu. Jednovrstevná deklarace znamená, že typy uvnitř *chameleon* schématu mohou být odvozeny výhradně z datových typů XML Schema nebo z typů z importovaného schématu s deklarovaným cílovým jmenným prostorem. Pokud není importující schéma deklarováno s vlastním cílovým jmenným prostorem jako defaultním, případné odkazy



mezi zahrnutými chameleon schémata selžou (entity chameleon schémat se stávají členy jmenného prostoru s prefixovanými entitami, ale mezi sebou se odkazují bez prefixů). Totéž nastane i v případě, kdy chameleon schéma je pouze jedno, ale uvnitř chameleon schématu se typy odvozují jeden ze druhého.

Pravidlo pro shodu `targetNamespace` a `defaultNamespace` je zvoleno z důvodu čitelnosti schémat; entity ze jmenného prostoru XML Schema budou v tomto případě vždy kvalifikované.

Příklad

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="Root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="osoba" type="OsobaType" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
    <xs:key name="OsobaKey">
      <xs:selector xpath="osoba"/>
      <xs:field xpath="@jmeno"/>
    </xs:key>
    <xs:keyref name="zamestnani" refer="OsobaKey">
      <xs:selector xpath="osoba"/>
      <xs:field xpath="@ zamestnani"/>
    </xs:keyref>
  </xs:element>
  <xs:complexType name="OsobaType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="zamestnani" type="xs:string"/>
        <xs:attribute name="jmeno" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>
```

Zahrneme-li toto schéma pomocí příkazu `include` do jiného schématu s prefixovaným cílovým jmenným prostorem, výše uvedené odkazy selžou.

5.1.3.3. Formát jmenných prostorů

Pravidla pro cílový jmenný prostor schématu ISVS

Pojmenování jmenného prostoru v sobě NESMÍ mít zabudovány žádnou další logiku zpracování a MUSÍ sloužit pouze jako jedinečný trvalý identifikátor zdroje. JE DOPORUČENO, aby jmenný prostor byl ve formátu URN.

Vysvětlení

Jmenný prostor je primárně pouze identifikátor sady deklarácí datových objektů. Vkládání jakékoli další logiky do pojmenování jmenného prostoru může vnášet nežádoucí boční efekty do tohoto primárního určení a není tedy povoleno.

Formát URN pro jmenné prostory byl zvolen pro jasnou indikaci charakteru jmenného prostoru jako identifikátoru a dále pak pro trvalý charakter URN identifikátorů. Zatímco URL adresa zdroje se může změnit, URN má perzistentní charakter.



5.1.3.3.1. Konvence pro vytváření URN

Pravidla

URN jmenného prostoru XML schémat ISVS MUSÍ být vytvořen v souladu s IETF RFC 2141 (URN Syntax). URN obsahuje:

- Vedoucí sekvenci znaků „urn:“
- Identifikátor jmenného prostoru URN (Namespace Identifier – NID)
- Specifický řetězec jmenného prostoru URN (Namespace Specific String – NSS)

NID pro jmenné prostory schémat ISVS MUSÍ být řetězec „cz:“. Specifický řetězec jmenného prostoru MUSÍ začínat řetězcem „isvs:“, následovaným řetězcem, charakterizujícím oblast veřejné správy (resort). Řetězec oblasti veřejné správy BY MĚL být shodný s registrovaným doménovým jménem dané instituce a MUSÍ být ukončen dvojtečkou (:). Řetězec oblasti MUSÍ být následován řetězcem „schemas:“. Další část NSS JE VOLITELNÁ a její syntaxe je plně v kompetenci navrhovatele URN; MUSÍ však být v souladu s IETF RFC 2141.

Je doporučeno, aby vedoucí sekvence „urn“ a identifikátor jmenného prostoru byly uvedeny malými písmeny.

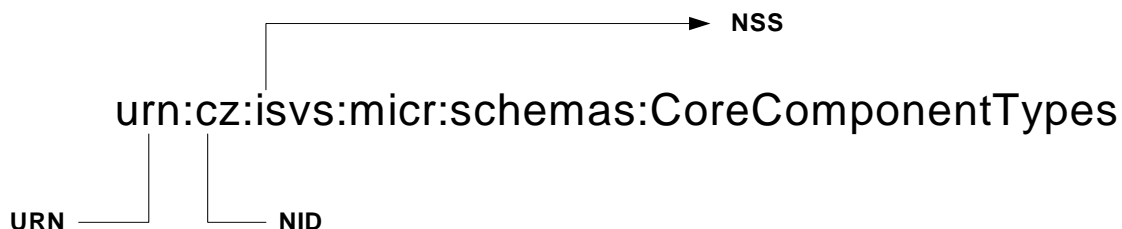
Vysvětlení

Použití jmenných prostorů ve formátu URN je v souladu s pojetím jmenného prostoru jako trvalého identifikátoru. Jmenné prostory ve formátu URL implikují nějaký zdroj, umístěný na dané URL adrese, což nemusí být v mnoha případech pravda.

I když IETF RFC 2141 uvádí, že vedoucí sekvence „urn“ a NID jsou case-insensitive, bude vhodné dodržovat konvenci o uvádění těchto částí malými písmeny. Důvodem je, že jmenné prostory XML jsou case-sensitive. Dva řetězce URN, kde bude např. NID uveden jednou velkými a podruhé malými písmeny, budou sice z pohledu RFC 2141 identická jména, ale z pohledu XML Schema to budou dva různé jmenné prostory.

Příklad

Uveďme si příklad pro URN cílového jmenného prostoru schématu základních datových typů ISVS v působnosti Ministerstva informatiky ČR:





5.1.3.3.2. *Konvence pro vytváření URL*

Pravidla

URL jmenného prostoru XML schémat ISVS MUSÍ být vytvořeno v souladu s IETF RFC 1738 (URL). URL NESMÍ být ve formátu relativní adresy.

Vysvětlení

Použití jmenných prostorů ve formátu URL není doporučeno, nelze jej však zcela zakázat, a to především z historických důvodů. Proto jsou zařazena i tato pravidla, upřesňující použití URL.

5.1.3.4. Použití prefixů jmenných prostorů

5.1.3.4.1. *Použití známých prefixů*

Pravidla

JE DOPORUČENO, aby vývojáři ISVS při vytváření XML dokumentů používali pro známé a často užívané jmenné prostory známé a dobře definované prefixy.

Vysvětlení

Použití obecně známých prefixů jmenných prostorů usnadňuje uživatelům XML schémat a dokumentů orientaci v takových dokumentech a přispívá k čitelnosti řešení.

Příklad

Mějme například následující fragment schématu:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://example.com"> ,
```

kde používáme dobře známý prefix xs: k přiřazení jmenného prostoru XML Schema. Vyloučena by tedy měla být například takováto deklarace:

```
<abc:schema xmlns:abc="http://www.w3.org/2001/XMLSchema" targetNamespace="http://example.com">
```

5.1.3.4.2. *Volitelnost prefixů*

Pravidla

JE DOPORUČENO, aby při čtení instancí XML dokumentů aplikačními nástroji vývojáři očekávali jakékoli prefixy, a to i pro známé jmenné prostory. Vývojáři NESMĚJÍ požadovat použití konkrétních prefixů jmenných prostorů.

Vysvětlení

V rámci řešení ISVS nesmějí být požadovány konkrétní prefixy jmenných prostorů. Pokud by bylo takové pravidlo přijato, jednalo by se o zavedení komplikovaných a nepříliš dobře spravovatelných mechanismů do návrhu schémat; v rámci ISVS by např. musela být pravděpodobně zavedena centrální evidence povolených prefixů včetně pravidel jejich verzování.

Příklad

Pokud máme deklarováno např. schéma

```
<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns="test" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">  
  <xs:element name="root">
```



```
<xs:complexType>
  <xs:sequence>
    <xs:element ref="text"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="text" type="xs:string"/>
</xs:schema>
```

je požadováno, aby bylo možno validovat jak dokument s defaultní deklarací (bez prefixu):

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns:aaa="test" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="test
C:\temp\test.xsd">
  <text>pokus</text>
</root>
```

stejně jako dokument se změněným prefixem jmenného prostoru

```
<?xml version="1.0" encoding="UTF-8"?>
<aaa:root xmlns:aaa="test" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="test
C:\temp\test.xsd">
  <aaa:text>pokus</aaa:text>
</aaa:root>
```

5.1.3.4.3. Duplicitní deklarace prefixů

Pravidla

JE DOPORUČENO, aby vývojáři ISVS nepoužívali v instancích XML dokumentů přiřazení více prefixů jednomu jmennému prostoru. Stejně tak není doporučeno přiřadit jeden prefix více jmenným prostorům.

Vysvětlení

Třebaže přiřazení více prefixů jednomu jmennému prostoru či přiřazení jednoho prefixu více jmenným prostorům není fatální chybou, je velmi vhodné vyvarovat se takových řešení. Důvodem je především čitelnost a přehlednost řešení.

Příklad

Je nutno vyvarovat se přiřazení více prefixů jednomu jmennému prostoru:

```
<?xml version="1.0" encoding="UTF-8"?>
<aaa:root xmlns:aaa="test" xmlns:bbb="test" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="test C:\temp\test.xsd">
  <bbb:text>pokus</bbb:text>
</aaa:root>
```

stejně jako přiřazení jednoho prefixu více jmenným prostorům (i když toto řešení není v některých parserech přípustné):

```
<?xml version="1.0" encoding="UTF-8"?>
<aaa:root xmlns:aaa="test" xmlns:aaa="http://www.w3.org/2001/XMLSchema-instance" aaa:schemaLocation="test
C:\temp\test.xsd">
  <aaa:text>pokus</aaa:text>
</aaa:root>
```



5.1.3.5. Použití *elementFormDefault* a *attributeFormDefault* kvalifikátorů

Pravidla

JE DOPORUČENO, aby vývojáři ISVS nepoužívali v instancích XML dokumentů přiřazení více prefixů jednomu jmennému prostoru. Stejně tak není doporučeno přiřadit jeden prefix více jmenným prostorům.

Vysvětlení

Třebaže přiřazení více prefixů jednomu jmennému prostoru či přiřazení jednoho prefixu více jmenným prostorům není fatální chybou, je velmi vhodné vyvarovat se takových řešení. Důvodem je především čitelnost a přehlednost řešení.

Příklad

Je nutno vyvarovat se přiřazení více prefixů jednomu jmennému prostoru:

```
<?xml version="1.0" encoding="UTF-8"?>
<aaa:root xmlns:aaa="test" xmlns:bbb="test" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="test C:\temp\test.xsd">
  <bbb:text>pokus</bbb:text>
</aaa:root>
```

stejně jako přiřazení jednoho prefixu více jmenným prostorům (i když toto řešení není v některých parserech přípustné):

```
<?xml version="1.0" encoding="UTF-8"?>
<aaa:root xmlns:aaa="test" xmlns:aaa="http://www.w3.org/2001/XMLSchema-instance" aaa:schemaLocation="test
C:\temp\test.xsd">
  <aaa:text>pokus</aaa:text>
</aaa:root>
```

5.1.3.6. Globální deklarace elementů

Pravidla

JE DOPORUČENO, aby vývojáři ISVS deklarovali jako globální pouze takové komponenty schémat, které budou splňovat některá z následujících kritérií:

- Jsou opakovaně používány uvnitř schématu
- Jsou určeny pro opakované použití jinými schématy
- Jsou zamýšleny jako kořenový element instancí XML dokumentů.

Vysvětlení

Elementy a složené datové typy, které jsou deklarovány jako dceřinné struktury elementu `xs:schema`, jsou globální struktury a jako takové mohou být opakovaně využity jak uvnitř schématu, tak i v jiném schématu, které stávající schéma importuje či zahrne. Také členy substitučních skupin je nutno deklarovat globálně.

Každý z globálních elementů pak může být použit jako kořenový element validního dokumentu.



Elementy a složené datové typy, jejichž deklarace je umístěna uvnitř deklarací složených datových typů, jsou lokálně deklarovány. V jednom schématu se může vyskytnout více lokálních deklarací komponent nejen se stejným názvem, ale i s různými datovými typy.

Je dále nutno si uvědomit, že pouze globálně deklarované elementy je možno opakovaně využít v jiných schématech, do kterých bude dané schéma importováno.

5.1.3.7. Lokální deklarace atributů

Pravidla

JE DOPORUČENO, aby vývojáři schémat ISVS deklarovali atributy lokálně. Pouze v indikovaných případech potřeby globálních a opakovaně použitelných atributů MOHOU schémata obsahovat globální deklarace atributů.

Vysvětlení

Atributy, které jsou deklarovány jako dceřiné struktury elementu `xs:schema`, jsou globální atributy a mohou být opakovaně využity jak uvnitř schématu, tak i v jiném schématu, které stávající schéma importuje či zahrne. Atributy deklarované uvnitř deklarací složených typů jsou lokální atributy.

Je dále nutno si uvědomit, že pouze globálně deklarované atributy či skupiny atributů je možno opakovaně využít v jiných schématech, do kterých bude dané schéma importováno.

5.1.3.8. Datové typy vs. elementy

Pravidla

Referenční schémata a schémata **datových slovníků** MUSEJÍ být navržena tak, že budou globálně deklarovat *sadu* elementů, určených k opakovanému využití. Spolu s elementy MUSEJÍ být globálně deklarovány i příslušné datové typy.

Schémata zpráv MUSEJÍ obsahovat pouze *jeden* globální element, deklarující kořenový element instance XML dokumentu; dále pak MOHOU obsahovat globální deklarace některých datových typů, určených k opakovanému použití.

Vysvětlení

Navržený design schémat (tzv. *Garden of Eden* design) vychází z předpokladu, že:

- Je nutno zajistit, aby jména elementů zůstala nezměněná během importu schémat do jiných schémat. Proto je nutno deklarovat globální elementy, opakovaně použitelné referencí.
- Pro opakované využití komponent mimo původní problémovou doménu, kdy je často třeba použít jiná jména elementů a tedy využívat datové typy, je doporučeno deklarovat datové typy použité pro definici elementů globálně.

Příklad

V prvním příkladu nalezneme globální deklaraci elementu `Osoba`:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://example.com">  
  <xs:element name="Osoba">
```



```
<xs:complexType>
  <xs:sequence>
    <xs:element name="Jmeno" type="xs:string"/>
    <xs:element name="Prijmeni" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

V tomto případě lze opakovaně využít v jiném schématu pouze element `Osoba`, a to bez možnosti přejmenování.

Další příklad uvádí globální deklaraci typu `OsobaType`:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://example.com">
  <xs:complexType name="OsobaType">
    <xs:sequence>
      <xs:element name="Jmeno" type="xs:string"/>
      <xs:element name="Prijmeni" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Datový typ lze opakovaně využít a vytvořený element libovolně pojmenovat.

V posledním příkladu je uveden navrhovaný design, kdy v jednom schématu jsou přítomny globální deklarace jak elementu, tak typu:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://example.com"
  xmlns="http://example.com">
  <xs:complexType name="OsobaType">
    <xs:sequence>
      <xs:element name="Jmeno" type="xs:string"/>
      <xs:element name="Prijmeni" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Osoba" type="OsobaType"/>
</xs:schema>
```

5.1.3.9. Elementy vs. atributy

Pravidla

Design ISVS schémat MUSÍ být navržen tak, že elementy budou hlavní strukturou přenosu informace v instancích XML dokumentů. Pokud element obsahuje atributy, význam kteréhokoli z těchto atributů BY MĚL být aplikovatelný na všechny dceřiné elementy daného elementu.

Hodnoty atributů BY MĚLY být krátké, nejlépe čísla či kódy. Atributy s dlouhými řetězci v hodnotě BY NEMĚLY být používány.

Atribut NESMÍ být použit ke kvalifikaci obsahu jiného atributu.

Vysvětlení

Jakmile je jednou pro přenos informace použito atributu, není možno již tuto informaci dále strukturovat. Proto je jako hlavní přenosová struktura uvnitř XML dokumentu navrhován element.



Atributy jsou doporučovány pouze k vyjádření kvalifikace obsahu elementu. Slouží tedy jako metadata obsahu elementu, a to nejen textového obsahu elementu, ale i dceřiných elementů. Vhodné jsou např. k vyjádření formátu souboru, jednotky měření apod.

Příklad

Následující příklad uvádí nevhodné použití atributu:

```
<osoba datumNarozeni="24.05.1959">  
  <jmeno>Pavel</jmeno>  
  <prijmeni>Benda</prijmeni>  
</osoba>
```

Atribut `datumNarozeni` nese informaci, ne metadata. Uvedené datum by mělo být tedy vhodněji nesené v elementu:

```
<osoba>  
  <datumNarozeni>24.05.1959</datumNarozeni>  
  <jmeno>Pavel</jmeno>  
  <prijmeni>Benda</prijmeni>  
</osoba>
```

Pokud bychom potřebovali zařadit metadata např. o formátu data, můžeme tak učinit pomocí atributu, například následujícím způsobem:

```
<osoba>  
  <datumNarozeni format="dd.mm.rrrr">24.05.1959</datumNarozeni>  
  <jmeno>Pavel</jmeno>  
  <prijmeni>Benda</prijmeni>  
</osoba>
```

Naprostě nepřijatelné je pak kvalifikování obsahu atributu jiným atributem:

```
<osoba datumNarozeni="24.05.1959" datumFormat="dd.mm.rrrr">  
  <jmeno>Pavel</jmeno>  
  <prijmeni>Benda</prijmeni>  
</osoba>
```

5.1.3.10. Použití referencí uvnitř XML dokumentů

Pravidla

Pro vyjádření referencí uvnitř instancí XML dokumentů JE DOPORUČENO použít odkazů pomocí `xs:key`, `xs:unique` a `xs:keyref`. Metoda odkazování pomocí datových typů ID/IDREF BY NEMĚLA být používána.

Vysvětlení

Odkazování pomocí datových typů ID/IDREF má podstatnou nevýhodu v tom, že v rámci jedné instance dokumentu je možno použít každé ID pouze jednou; ID má tedy v rámci dokumentu globální platnost. Naproti tomu `xs:key` a `xs:unique` identifikátory je možno uplatnit na libovolné lokální úrovni hierarchie dokumentu; a dále, jedinečnost klíčů `xs:key` / `xs:unique` je požadována pouze na této lokální úrovni. Použití datových typů ID/IDREF se navíc nedoporučuje (viz 5.1.3.13)



5.1.3.11. Číselníky a jejich indikace

Pravidla

Kódy číselníků nebo hodnoty identifikátorů identifikačních schémat, používané v XML schématech ISVS, BY MĚLY být deklarovány přímo ve schématech výčtem hodnot. Není-li to možné, MUSÍ být v metadatech schématu obsažena identifikace číselníku. JE DOPORUČENO pro identifikaci zdrojů použít URI. Identifikátory číselníků MOHOU být nesený i v instanci XML dokumentu pomocí vhodných struktur, například atributů základních datových typů ISVS.

Vysvětlení

Je-li to přijatelné z důvodu velikosti číselníku, je vhodné použít přímé deklarace číselníku ve schématu výčtem pomocí `xs:enumeration` uvnitř `xs:restriction` deklarace. Je-li číselník neúměrně rozsáhlý, je nutno pro daný datový typ či element deklarovat jako zdroj hodnot externí číselník či identifikační schéma. Deklaraci externího číselníku je vhodné umístit do metadat daného datového typu či elementu; vhodné je použití Dublin Core elementu `relation` s rozšířením elementem `requires` (viz 5.1.5.1).

Identifikace externího číselníku je možná i přímo v instanci XML dokumentu. Elementy mohou mít deklarovány například atributy metadat, nesoucí například informaci o textové hodnotě kódu nebo URL adresu číselníku. Tento koncept je použit v návrhu schémat základních datových typů ISVS (viz 4.1).

Příklady

Deklarace číselníku výčtem v rámci složeného typu je velmi jednoduchá:

```
<xs:complexType name="OsobaStudiumStavNazevType">
  <xs:simpleContent>
    <xs:restriction base="cct:IdentifikatorType">
      <xs:enumeration value="ukončené"/>
      <xs:enumeration value="probíhá"/>
      <xs:enumeration value="přerušené"/>
      <xs:enumeration value="nedokončené"/>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>
```

Pokud není možné zařadit výčet a je nutné deklarovat závislost na externím číselníku; doporučený způsob je pomocí metadat Dublin Core:

```
<xs:complexType name="OsobaStudiumStavNazevType">
  <xs:annotation>
    <xs:appinfo>
      <rdf:RDF>
        <rdf:Description rdf:about="http://localhost/uvis_datatypes/v_1.0.2/#STSTUD">
          <dc:creator>Pavel Benda (benda@asd-software.cz)</dc:creator>
          <dc:title>STAV STUDIA</dc:title>
          <dcterms:created>2002-12-09</ dcterms:created>
          < dcterms:valid>1. 1. 1999</ dcterms:valid>
          < dcterms:requires>http://www.ListSource.cz/StavStudia</ dcterms:requires>
        </rdf:Description>
      </rdf:RDF>
    </xs:appinfo>
  </xs:annotation>
```



```
<xs:simpleContent>  
  <xs:restriction base="cct:IdentifikatorType">  
    <xs:minLength value="1"/>  
    <xs:maxLength value="25"/>  
  </xs:restriction>  
</xs:simpleContent>  
</xs:complexType>
```

V instanci XML dokumentu je možno zařadit číselníkové informace např. následujícím způsobem:

```
<osoba>  
  <jmeno>Pavel</jmeno>  
  <prijmeni>Benda</prijmeni>  
  <stavStudiaKod ciselnikID="http://www.ListSource.cz/StavStudia" kodJmeno="ukončené">1</stavStudiaKod>  
</osoba>
```

5.1.3.12. Hodnoty default a fixed

Pravidla

Pro design XML schémat v rámci ISVS NESMÍ být použito atributů `default` a `fixed` při deklaraci datového obsahu elementů a atributů.

Vysvětlení

Základním problémem při použití `default` a `fixed` hodnot je skutečnost, že po validaci vkládají nová data do XML dokumentů. Existují tedy dva odlišné stavy téhož dokumentu, a to před validací a po validaci, přičemž dokument před validací je de facto nekompletní. Teprve validací vzniklý PSVI (Post Schema Validation Infoset) je plnohodnotná instance XML dokumentu se schématem, deklarujícím `default` a/nebo `fixed` hodnoty.

Zařazení validačního procesu jako zdroje datového obsahu dokumentu není šťastné řešení, neboť ne vždy je schéma k dispozici; navíc ne každý příjemce dokumentu musí nutně provést validaci.

Největší problémy může však použití `default` a `fixed` hodnot vnést do procesů, spojených s kryptografickými algoritmy. Je sporné, jak interpretovat elektronický podpis, provedený nad nevalidovaným dokumentem a nad PSVI.

Problémy také mohou nastat při použití datového typu `QName`, kdy prefixu jmenného prostoru elementu nebo atributu s datovým typem `QName`, deklarovaným jako `fixed` nebo `default`, může být v instanci dokumentu přidělen jiný jmenný prostor (aktuální přiřazení v `QName` není v okamžiku vytvoření dokumentu viditelné).

5.1.3.13. Omezení použití datových typů

Pravidla

Pro design XML schémat v rámci ISVS se NEDOPORUČUJE použití datových typů `ID`, `IDREF`, `IDREFS`, `ENTITY`, `ENTITIES`, `NOTATION`, `NMTOKEN` a `NMTOKENS`.

Vysvětlení

Datové typy `ID`, `IDREF`, `IDREFS`, `ENTITY`, `ENTITIES`, `NOTATION`, `NMTOKEN` a `NMTOKENS` byly mezi datové typy XML Schema zařazeny pouze z důvodu zpětné



kompatibility s DTD. Jejich použití je podle specifikace XML Schema [9] omezeno pouze na atributy a obecně se nedoporučuje jejich použití.

5.1.3.14. Komentáře ve schématech

Pravidla

Pro schémata v rámci ISVS se pro účely zařazení komentářů do schémat DOPORUČUJE použití elementu `documentation`. XML komentáře BY NEMĚLY být používány. Pokud délka dokumentace přesáhne jeden odstavec, JE DOPORUČENO použití fragmentů XHTML v obsahu elementu `documentation`.

Vysvětlení

Element `documentation`, který je dceřiným elementem elementu `annotation`, je velmi vhodným nástrojem pro komentování schémat. Jeho výhodou je jednoznačná lokalizace komentáře, snadná parsovatelnost (je možno například vybrat všechny komentáře schématu spolu s rodičovskými elementy) a použitelnost například pro vytváření dokumentace pomocí XSLT šablon. XML komentáře (XML Comments) jsou naproti tomu pro výše zmíněné účely použitelné pouze obtížně.

Metodika vkládání fragmentů XHTML byla zvolena proto, že neformátovaný text je u delších popisů nepřehledný; navíc neumožňuje např. vkládání odkazů, tabulek apod.

5.1.3.15. Usnadnění opakovaného použití schémat

Pravidla

Pro schémata v rámci ISVS se NEDOPORUČUJE použití redefinic komponent pomocí mechanismu `xs:redefine`. Referenční schémata a schémata datových slovníků NESMĚJÍ použít redefinice komponent pomocí mechanismu `xs:redefine`. Import schémat pomocí `xs:import` NESMÍ být proveden bez deklarace importovaného jmenného prostoru pomocí `namespace` atributu.

Vysvětlení

Použití redefinice komponent může vnášet do dat neočekávané vedlejší vlivy a narušuje dědičnost mezi původní a redefinovanou strukturou.

Pokud importujeme schéma bez deklarace jmenného prostoru, maskujeme tím původ definice komponent a znesnadňujeme v případě problémů detekci chyb.

5.1.4. Design komponent schémat

5.1.4.1. Pojmenovací konvence

5.1.4.1.1. Jazyk a gramatický tvar jmen

Pravidla

Jména komponent schémat MUSEJÍ být ve spisovné češtině. Výjimku MOHOU tvořit termíny v češtině dosud nekodifikované, ale obecně srozumitelné. Všechna jména MUSEJÍ být v prvním



pádu jednotného čísla, pokud není sám koncept typu či elementu v množném čísle. Diakritika ve jménech MUSÍ být odstraněna.

Vysvětlení

Primární vlastností jmen komponent schémat musí být sémantická jednoznačnost a široká srozumitelnost. Proto není přípustný jak technický žargon, tak směšování např. anglických a českých termínů ve jménech.

Příklad

V následujícím příkladu je uveden dvojí tvar elementu, obsahujícího aktualizaci údajů faktury:

```
<FakturaUpdate />
```

```
<FakturaOprava />
```

Pouze druhý název elementu je přípustný, neboť termín update (ač obecně známý v komunitě IT) nemusí být srozumitelný běžnému uživateli.

V dalším příkladu je zřejmý koncept jednotného a množného čísla jmen:

```
<FakturyDavka>
  <Faktura>
    <FakturalID>1</FakturalID>
    .....
  </Faktura>
  <Faktura>
    <FakturalID>2</FakturalID>
    .....
  </Faktura>
</FakturyDavka>
```

Koncept elementu `<FakturyDavka>` je sada záznamů faktur, je tedy přípustné množné číslo. Jednotlivé elementy faktur pak již musejí být v jednotném čísle.

5.1.4.1.2. Case konvence

Pravidla

Vývojáři XML schémat ISVS BY MĚLI pro vytváření jmen komponent schémat používat *camel case* konvenci. Pro jména elementů MUSÍ být použit model *UpperCamelCase*, pro jména atributů MUSÍ být použit model *lowerCamelCase*.

Vysvětlení

Standardizační organizace a iniciativy (např. OASIS, ebXML, UBL, RosettaNet, BizTalk, US Federal CIO Council a další) přistupují k použití *camelCase* konvence pro zvýšení srozumitelnosti názvů XML komponent.

Příklad

V následujícím příkladu je element platby ve formátu *UpperCamelCase* a atribut měny ve formátu *lowerCamelCase*:

```
<FakturaPlatbaCastka menaKod="USD">256</FakturaPlatbaCastka>
```



5.1.4.1.3. Zkratky a akronymy

Pravidla

Vývojáři XML schémat ISVS BY NEMĚLI pro vytváření jmen komponent schémat používat akronymy. Pokud jsou akronymy použity, MUSÍ se jednat pouze o akronymy obecně srozumitelné. Akronymy MUSEJÍ být ve jménu uvedeny velkými písmeny. Použité akronymy MUSEJÍ být rozepsány do jednotlivých slov v dokumentaci k příslušné komponentě. Je-li reprezentace (Representation Class) datového prvku typu Identifikátor, MUSÍ být tato složka jména, je-li použita, zkrácena na řetězec „ID“.

Pro jména XML komponent NESMĚJÍ být používány zkratky.

Vysvětlení

Použití akronymů je vhodné pouze v okamžiku, kdy je zřejmé, že použitý akronym má obecnou platnost a bude srozumitelný pro všechny potenciální účastníky komunikace. Zkratky nesmějí být používány vůbec; důvodem je možnost vzniku neočekávaných zavádějících slovních kombinací.

Příklad

Správné názvy elementů:

<FakturaPlatbaCastka>

<OrganizaceCO>

<OrganizaceIdentifikacniCislo>

Chybné názvy elementů:

<CentralniBRC>

<FaktPlatCastka>

Jméno elementu `CentralniBRC` není v souladu s konvencemi, neboť BRC je akronym dobře známý jen v bioinformatické komunitě (Biological Resource Center), zatímco v jiném kontextu může mít zcela jiný význam (např. British Retail Consortium). Jméno elementu `FaktPlatCastka` je složeno ze zkratk.

5.1.4.1.4. Povolené znaky

Pravidla

Ve jménech XML komponent NESMĚJÍ být používány tečky (.), podtržítka (_) a pomlčky (-). Jména NESMĚJÍ obsahovat jiné znaky než písmena nebo číslice.

Příklad

Chybné názvy elementů:

</Faktura_Platba.Castka>

</Faktura2003PlatbaCastka>

Jméno elementu `Faktura_Platba.Castka` je vytvořeno pomocí nepovolených znaků. Jméno elementu `Faktura2003PlatbaCastka` obsahuje číslice, indikující (například) účetní rok; elegantnější řešení je ovšem převést tuto logiku ze značkování do dat (např. do samostatného elementu).



5.1.4.1.5. *Jmenné konvence dle ČSN ISO/IEC 11179*

Pravidla

Jména XML elementů a atributů pro datové prvky, popsané pomocí ISVS schémat, BY MĚLY vyhovovat požadavkům normy ČSN ISO/IEC 11179, Část 5, pro jmenné a identifikační principy datových prvků.

Vysvětlení

ČSN ISO/IEC 11179-5 [2] zavádí konstruování jmen datových prvků ze tří hlavních komponent:

- Třída objektu (Object Class)
- Vlastnost (Property)
- Reprezentace (Representation Class)

Některé iniciativy (např. ebXML, UN/CEFACT) publikovaly omezenou množinu jmen pro Representation Class, které by měly být používány ve jménech datových prvků. Norma ČSN ISO/IEC 11179 však deklarativně seznam pro Representation Class neuvádí. Jména pro Object Class a Property nejsou omezena.

Příklad

Uveďme si příklad pro konstrukci datového prvku, popisujícího např. celkovou hmotnost nákladu:

Třída objektu: Naklad

Vlastnost: HmotnostCelkova

Reprezentace: Mereni

Jméno pro datový prvek pak v objektovém zápisu bude `Naklad.HmotnostCelkova.Mereni`, v *camel case* formátu pro XML komponenty pak `NakladHmotnostCelkovaMereni`.

Jména je možno zkracovat, pokud vypuštěním některé ze složek jména nedojde ke změně významu jména. Např. zkrácení uvedeného příkladu na `NakladHmotnostCelkova` nebo např. `OsobaRodneCisloID` na `OsobaRodneCislo` nebude mít vliv na srozumitelnost jména.

5.1.4.1.6. *Jména lokálních elementů a atributů*

Pravidla

Jména XML elementů a atributů, deklarovaných v XML schématech ISVS lokálně, MOHOU mít vynechán popis třídy objektu, pokud je jejich rodičovský element deklarací této třídy a pokud to není na úkor srozumitelnosti.

Vysvětlení

Lze-li aplikovat jméno rodičovského elementu jako identifikátor třídy objektu, je zbytečné opakovat tuto třídu objektu ve všech dceřiných elementech či atributech daného elementu. Lokálně deklarované elementy a atributy tedy mohou mít jména konstruována jednodušeji s přihlédnutím k rodičovskému elementu.



Příklad

Uveďme si příklad pro design elementu, popisujícího např. osobu. Raději než „upovídaný“ a doslovný design:

```
<Osoba>  
  <OsobaID>1234</OsobaID>  
  <OsobaJmeno>Pavel</OsobaJmeno>  
  <OsobaPrijmeni>Benda</OsobaPrijmeni>  
  <OsobaPohlavi>muž</OsobaPohlavi>  
</Osoba>
```

bude vhodnější použít následující syntaxi:

```
<Osoba>  
  <OsobaID>1234</OsobaID>  
  <Jmeno>Pavel</Jmeno>  
  <Prijmeni>Benda</Prijmeni>  
  <Pohlavi>muž</Pohlavi>  
</Osoba>
```

Povšimněte si, že element `OsobaID` zůstal nezměněn – je vhodnější ponechat element ID v původním tvaru, který lépe vystihuje určení elementu.

Podobně jako u elementů, lokálně deklarovaný atribut může mít název jednoduchý:

```
<FakturaDatumSplatnosti format="dd.mm.rrrr">15.01.2003</FakturaDatumSplatnosti>
```

Pokud by byl tentýž atribut deklarován globálně, měl by mít jméno vyhovující pravidlu:

```
<FakturaDatumSplatnosti datumFormat="dd.mm.rrrr">15.01.2003</FakturaDatumSplatnosti>
```

5.1.4.1.7. Jména elementů zpráv

Pravidla

Globální element, deklarující kořenový element instance XML dokumentu zprávy, MUSÍ být pojmenován v souladu s kontextem použití zprávy.

Kořenový element instance XML dokumentu zprávy NESMÍ mít jméno `<Envelope>`.

Vysvětlení

Ze jména kořenového elementu zprávy by mělo být na první pohled patrné, k jakému účelu je zpráva určena. Nepřípustná jsou proto jména elementů s neutrálním obsahem bez informace o obsahu zprávy.

Jméno elementu `<Envelope>` je vyhrazeno pro obálku protokolu SOAP [26]. Pro lepší orientaci a zamezení záměn je použití tohoto jména mimo kontext SOAP zakázáno.

Příklad

Korektně pojmenovaný element zprávy, například zprávy dotazu do registru ekonomických subjektů, bude mít tvar:

```
<DotazRegistrARES>
```

Chybný by v tomto kontextu byl například element:

```
<KorenovyElement>
```



5.1.4.1.8. *Jména jednoduchých a složených typů XML*

Pravidla

Navržená jména jednoduchých a složených typů XML ISVS schémat MUSEJÍ vyhovovat požadavkům kapitoly 5.1.4.1 tohoto dokumentu. Jména jednoduchých typů a složených typů s jednoduchým obsahem BY MĚLA navíc mít ke jménu připojeno příponu `Type`. Jména složených typů se složeným obsahem BY MĚLA mít ke jménu připojeno příponu `Structure`. Jména jednoduchých i složených typů NESMÍ s výjimkou typové přípony obsahovat termíny `Type` a `Structure`.

Příklady

Příklady správně vytvořených jmen komponent:

`NakladHmotnostCelkovaType`

`DodavkaStructure`

Nesprávně vytvořené jméno komponenty:

`NakladTypeType`

5.1.4.2. Použití základních datových typů ISVS

Pravidla

Pro nově vytvářená XML schémata ISVS JE DOPORUČENO, aby všechny datové typy, používané v rámci ISVS, vycházely z množiny základních datových typů ISVS. Pro existující schémata je DOPORUČENO uplatnit toto pravidlo při přechodu na novější verze schémat.

Vysvětlení

Základní datové typy ISVS jsou koncipovány jako nejmenší možný společný jmenovatele všech XML schémat ISVS. Při důsledném uplatnění základních datových typů dojde k žádoucímu zvýšení srozumitelnosti schémat a ke zvýšení interoperability.

5.1.4.3. Opakované použití ISVS typů a elementů

Pravidla

Vývojáři XML schémat v rámci ISVS MUSEJÍ ve svých schématech a instancích XML dokumentů používat existující návrhy datových typů a elementů referenčních schémat ISVS. Stejně tak MUSEJÍ používat existující datové slovníky, pokud jsou využitelné pro jejich účely. Vyvíjet vlastní datové typy, elementy a datové slovníky MOHOU pouze v případě, že v rámci ISVS neexistuje vhodné využitelné řešení.

Komponenty XML schémat v rámci ISVS by měly být opakovaně používány především mechanismem odkazů na globálně deklarované elementy.

Vysvětlení

Pro dosažení interoperability je žádoucí v maximální míře opakovaně používat navržené komponenty XML schémat ISVS prostřednictvím mechanismu importu či zahrnutí schémat. Jakákoli diverzifikace řešení vnáší nežádoucí nutnost přídatného mapování mezi datovými slovníky.



Použitím mechanismu odkazů na globálně deklarované elementy je zajištěna identita jmen elementů v rámci ISVS. To napomůže čitelnosti a sémantické jednoznačnosti dat, přenášených v rámci ISVS. Pokud jsou však komponenty využívány např. mimo původní problémovou doménu datového slovníku, je možné opakovaně využít datové typy (složené i jednoduché) v jiné oblasti a přejmenovat je podle vlastního uvážení; samozřejmě v souladu se jmennými konvencemi.

5.1.4.4. Podpora dědičnosti typů

Pravidla

Opakovaně používané datové typy MOHOU být podrobeny restrikci nebo rozšíření. Tyto úpravy MUSEJÍ být provedeny s ohledem na zachování maximální zpětné kompatibility s výchozím typem.

Vysvětlení

Datové typy mohou být „redefinovány“, pokud zcela nevyhovují požadovaným účelům konkrétního řešení. Každá úprava by však měla v optimálním případě proběhnout tak, aby množina platných hodnot pro nový datový typ byla podmnožinou platných hodnot výchozího typu. Není vhodné tedy např. upravit jednoduchý datový typ pomocí `xs:extension` nebo složený typ restrikcí některých elementů.

5.1.4.5. Smíšený obsah elementů

Pravidla

Schémata ISVS pro datově orientované XML dokumenty NESMĚJÍ umožňovat `mixedContent` model obsahu elementů. Schémata pro textově orientované XML dokumenty MOHOU mít deklarován `mixedContent` model obsahu elementů.

Vysvětlení

Smíšený obsah elementů je naprosto přirozeným jevem pro textově orientované XML dokumenty. Naproti tomu pro datově orientované XML dokumenty je velmi vhodné omezit výskyt hodnot na koncové listy stromové struktury XML dokumentu.

5.1.4.6. Reprezentace dat pomocí kódů nebo textu

Pravidla

Schémata ISVS pro datově orientované XML dokumenty a dokumenty dočasné BY MĚLA definovat datový obsah s důrazem na kódy. Schémata pro textově orientované XML dokumenty a dokumenty, které budou trvale uchovány, BY MĚLA mít definován datový obsah s orientací na textové hodnoty.

Vysvětlení

Pro dočasné a datově orientované XML dokumenty (typicky zprávy při komunikaci pomocí XML rozhraní) je vhodné používat pro definici datového obsahu kódy, především z důvodu menší pravděpodobnosti chyb z přepisů a vhodnosti kódů jako položky klíčů. Naproti tomu dokumenty, u nichž je předpoklad dlouhodobého uchování, by měla být spíše orientována na



textové hodnoty. Důvodem je možnost ztráty vazby na související číselník, což je v horizontu desítek let naprosto reálný předpoklad (například pro archivní dokumenty).

5.1.5. Dokumentace schémat a instancí XML dokumentů.

Pravidla

Všechna schémata ISVS MUSEJÍ být dokumentována v souladu s touto kapitolou 5.1.5 této dokumentace. NEDOPORUČUJE se dokumentovat instance XML dokumentů.

Pro schémata v rámci ISVS se pro účely minimální dokumentace schémat DOPORUČUJE použití elementu `documentation`. XML komentáře BY NEMĚLY být používány.

Referenční schémata a schéma základních datových typů MUSEJÍ kromě české verze dokumentace obsahovat i anglickou verzi. Schémata datových slovníků a schémata zpráv BY MĚLA kromě české verze dokumentace obsahovat i anglickou verzi. Minimální rozsah anglické dokumentace BY MĚL zahrnovat element `documentation` a elementy Dublin Core, uvedené v oddíle 5.1.5.1 jako dvojjazyčné.

Vysvětlení

Schémata jako metadata o instancích XML dokumentů a jako trvalé zdroje informací by měla být opatřena alespoň minimální dokumentací. Naproti tomu instance XML dokumentů, které zvláště pro datově orientované zprávy XML rozhraní jsou entity s velmi krátkým životním cyklem, dokumentovány být nemusí, neboť k nim pravděpodobně nebude přistupovat nikdo bez znalosti souvislostí. Pouze trvalé (textově orientované) XML dokumenty mohou být dokumentovány.

Element `documentation`, který je dceřiným elementem elementu `annotation`, je velmi vhodným nástrojem pro základní dokumentaci schémat. Jeho výhodou je jednoznačná lokalizace dokumentace, snadná parsovatelnost a použitelnost například pro vytváření dokumentace pomocí XSLT šablon. XML komentáře (XML Comments) jsou naproti tomu pro výše zmíněné účely použitelné pouze obtížně.

5.1.5.1. Dokumentace pomocí standardu Dublin Core

Pravidla

Referenční schémata datových typů ISVS a datové slovníky MUSEJÍ obsahovat dokumentaci schémat pomocí Dublin Core elementů. Schémata zpráv BY MĚLY obsahovat dokumentaci schémat pomocí Dublin Core elementů. Dokumentace schémat MUSÍ být vložena do elementu `appinfo`, který je dceřiným elementem elementu `annotation`. Element `annotation` MUSÍ být vložen jako dceřiný element elementu `xs:schema`.

Referenční schémata datových typů ISVS MUSEJÍ obsahovat dokumentaci jednotlivých komponent schémat pomocí Dublin Core elementů. Datové slovníky ISVS MOHOU obsahovat dokumentaci jednotlivých komponent schémat pomocí Dublin Core elementů. Dokumentace komponent MUSÍ být vložena do elementu `appinfo`, který je dceřiným elementem elementu `annotation`.

Element `annotation` MUSÍ být pro deklaraci elementu vložen jako dceřiný element elementu `xs:element`, pro deklaraci jednoduchého datového typu jako dceřiný element elementu



`xs:simpleType` a pro deklaraci složeného datového typu jako dceřiný element elementu `xs:complexType`.

Pro dokumentaci pomocí Dublin Core MUSEJÍ být přidány do příslušných schémat jmenné prostory Dublin Core:

- Pro základní elementy Dublin Core <http://purl.org/dc/elements/1.1/> (v tomto dokumentu je mu přiřazen prefix `dc:`)
- Pro doplňkové elementy Dublin Core <http://purl.org/dc/terms/> (v tomto dokumentu je mu přiřazen prefix `dcterms:`)

Pro ty elementy v následující tabulce, které mají ve sloupci Překlad uvedeno „ano“, MUSÍ být, jsou-li tyto elementy použity, uvedeny jak verze česká, tak verze anglická.

Pro ty elementy v následující tabulce, které mají ve sloupci Překlad uvedeno „volitelně“, MŮŽE být, jsou-li tyto elementy použity, uvedeny jak verze česká, tak verze anglická.

Dokumentace schémat ISVS pomocí Dublin Core elementů BY MĚLA obsahovat údaje podle následujícího seznamu:

Položka	Dublin Core element(y)	Překlad
URL fyzické lokalizace schématu či umístění schématu	<code>dc:source</code>	ne
Cílový jmenný prostor schématu	<code>dc:identifier</code>	ne
Popis schématu	<code>dc:description</code>	ano
Importovaná nebo zahrnutá schémata	<code>dcterms:requires</code>	ne
Datum vytvoření schématu	<code>dcterms:created</code>	ne
Datum platnosti schématu	<code>dcterms:valid</code>	ne
Autor (autoři) schématu	<code>dc:creator</code>	ne
Osoba či organizace, zodpovědná za publikaci schématu	<code>dc:publisher</code>	ne
Hlavní jazyk schématu a jeho dokumentace	<code>dc:language</code>	ne

Dokumentace schémat ISVS pomocí Dublin Core elementů může dále volitelně obsahovat údaje podle následujícího seznamu:

Položka	Dublin Core element(y)	Překlad
Jméno schématu	<code>dc:title</code>	ano
Klíčová slova, identifikující oblast využití schématu	<code>dc:subject</code>	ano
Autorská práva, copyright, omezení distribuce a použití	<code>dc:rights</code>	volitelně



Spoluautor (spoluautoři)	dc:contributor	ne
Typ zdroje (hodnota musí obsahovat text „XML Schema“)	dc:type	ne
Odkaz na dokument, uvádějící licenční práva ke schématu	dc:license	ne
Odkaz na vlastníka licenčních práv ke schématu	dc:rightsHolder	ne
Popis změny práv ke schématu nebo správcovství schématu od okamžiku jeho uveřejnění	dc:provenance	ne

Dokumentace komponent schémat ISVS pomocí Dublin Core elementů BY MĚLA obsahovat údaje podle následujícího seznamu:

Položka	Dublin Core element(y)	Překlad
URI (URL) identifikaci deklarované komponenty, existuje-li	dc:identifier	ne
Jméno komponenty	dc:title	ano
Popis komponenty	dc:description	volitelně
Číselník, definující hodnoty datového prvku, existuje-li	dcterms:requires	ne
Datum vytvoření komponenty	dcterms:created	ne
Datum platnosti komponenty	dcterms:valid	ne

Dokumentace komponent schémat ISVS pomocí Dublin Core elementů MŮŽE dále volitelně obsahovat údaje podle následujícího seznamu:

Položka	Dublin Core element(y)	Překlad
Klíčová slova, identifikující oblast využití komponenty	dc:subject	ano
Autorská práva, copyright, omezení distribuce a použití	dc:rights	volitelně
Autor (autoři) komponenty	dc:creator	ne
Spoluautor (spoluautoři)	dc:contributor	ne
Osoba či organizace, zodpovědná za publikaci komponenty	dc:publisher	ne
Popis změny práv ke komponentě nebo správcovství komponenty od okamžiku jejího uveřejnění	dc:provenance	ne



Vysvětlení

Standard Dublin Core [18] je jeden ze široce přijímaných návrhů systému dokumentace zdrojů na Webu. Používá transparentní jednoduchou sémantiku, schopnou pokrýt většinu potřeb pro metadata schémat ISVS.

Uchovávaná metadata jsou z výše uvedených důvodů využitelná i v případě změn standardu uložení metadat v budoucnosti; úsilí vložené do dokumentace pomocí Dublin Core standardu je tedy dlouhodobou investicí.

Dokument s doporučeným postupem Best Practices pro použití entit Dublin Core pro dokumentaci XML dokumentů je přístupný na stránce DCMI [19].

Příklady

Příklad dokumentace schématu je uveden v následujícím výpisu:

```
<xs:schema targetNamespace="urn:cz:isvs:micr:schemas:CommonTypes:v1"
xmlns:cct="urn:cz:isvs:micr:schemas:CoreComponentTypes:v1" xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="urn:cz:isvs:micr:schemas:CommonTypes:v1"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.02a">
  <xs:annotation>
    <xs:documentation xml:lang="cs">Návrh XML schématu s definicemi obecných jednoduchých datových prvků
ISVS (Informačních systémů veřejné správy) verze 1.0.2. Toto schéma obsahuje definice jednoduchých datových
prvků, používaných pro ostatní schémata ISVS včetně ostatních schémat jednoduchých datových prvků a v rámci
celého systému ISVS ČR© ASD Software s.r.o., 2000 - 2002</xs:documentation>
    <xs:appinfo>
      <rdf:RDF>
        <rdf:Description rdf:about="http://localhost/schemas/isvs_common/1.02a" xml:lang="cs">
          <dc:creator>Pavel Benda (benda@asd-software.cz)</dc:creator>
          <dc:contributor>Martin Šilar (silar@asd-software.cz)</dc:contributor>
          <dc:description>Návrh XML schématu s definicemi obecných jednoduchých datových prvků ISVS
(Informačních systémů veřejné správy) verze 1.0.2. Toto schéma obsahuje definice jednoduchých datových prvků,
používaných pro ostatní schémata ISVS včetně ostatních schémat jednoduchých datových prvků a v rámci celého
systému ISVS ČR</dc:description>
          <dc:identifier>http://localhost/schemas/isvs_common/1.02a</dc:identifier>
          <dc:publisher>Ministerstvo informatiky ČR</dc:publisher>
          <dc:rights>© ASD Software s.r.o., 2000 - 2002</dc:rights>
          <dc:source>http://localhost/schemas/isvs_common/1.02a/isvs_common.xsd</dc:source>
          <dc:type>XML Schema</dc:type>
          <dcterms:created>2002-12-09</dcterms:created>
          <dcterms:valid>2002-12-09</dcterms:valid>
          <dc:subject>Jednoduché datové prvky; ISVS; JEDNODUCHÝ DATOVÝ PRVEK OBECNÝ;
Informační systémy veřejné správy</dc:subject>
          <dc:format>text/xml</dc:format>
          <dcterms:requires>http://localhost/schemas/uvis_simpletypes
Template_SimpleTypes.xsd</dcterms:requires>
        </rdf:Description>
      </rdf:RDF>
    </xs:appinfo>
  </xs:annotation>
  ....
```

Příklad dokumentace komponenty uvádí pak výpis:



```
<xs:complexType name="MenaType">
  <xs:annotation>
    <xs:documentation>Datový prvek pro slovní označení měn a fondů.</xs:documentation>
    <xs:appinfo>
      <rdf:RDF>
        <rdf:Description
rdf:about="http://localhost/XML_lfcs/uvis_tree/schemas/uvis_datatypes/v_1.0.2/#MENA">
          <dc:creator>Pavel Benda (benda@asd-software.cz)</dc:creator>
          <dc:description>Datový prvek pro slovní označení měn a fondů.</dc:description>
          <dc:title>NÁZEV MĚNY</dc:title>
          <dcterms:created>2002-12-09</dcterms:created>
          <dcterms:valid>2002-12-09</dcterms:valid>
        </rdf:Description>
      </rdf:RDF>
    </xs:appinfo>
  </xs:annotation>
  <xs:simpleContent>
    <xs:restriction base="cct:JmenoType">
      <xs:maxLength value="240"/>
      <xs:minLength value="2"/>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>
.....
```

5.1.5.2. Verzování schémat

Pravidla

Schématá ISVS MUSEJÍ obsahovat popis verze schématu pomocí atributu `version` elementu `xs:schema`. Každé nové verzi schématu, která je kompatibilní s předchozí verzí, MUSÍ být přidělena nová hodnota atributu `version`. Každé nové verzi schématu, která je nekompatibilní s předchozí verzí, MUSÍ být přidělen nový jmenný prostor.

Instance XML dokumentů zpráv MUSEJÍ obsahovat indikaci odpovídající verze schématu pomocí atributu `version` v kořenovém elementu instance dokumentu.

Vysvětlení

Použití atributu `version` je standardním mechanismem indikace verze schématu. Při zpětně kompatibilní změně schématu (např. po přidání nepovinných elementů) nebude potřeba modifikovat staré aplikace, případně udržovat dvě verze rozhraní pro staré a nové schéma.

Pokud však změny ve schématu vedou k nekompatibilitě mezi verzemi, je nutno změnu verze schématu doprovázet změnou jmenného prostoru.

Příklad

V následujícím výpisu je verze schématu uvedena jednak atributem `version`, jednak cílovým jmenným prostorem:

```
<xs:schema targetNamespace="urn:cz:isvs:micr:schemas:CommonTypes:v1"
xmlns:cct="urn:cz:isvs:micr:schemas:CoreComponentTypes:v1" xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="urn:cz:isvs:micr:schemas:CommonTypes:v1"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.02a">
```



5.1.5.3. Data a metadata závislá na platformě a aplikaci

Pravidla

Schémata ISVS NESMĚJÍ deklarovat a obsahovat data a metadata, která jsou specifická pro komunikační infrastrukturu, platformu, operační systém či koncovou aplikaci.

Vysvětlení

Zanášením dat a metadat, specifických pro komunikační infrastrukturu, platformu, operační systém či koncovou aplikaci (např. API volání, SQL příkazy apod.) se narušuje základní výhoda přenosu dat pomocí XML, a tím je právě nezávislost na zmíněných skutečnostech. Dokument se tím stává jednostranně orientovaným a předjímá způsob zpracování.

Dalším argumentem proti zavádění dat a metadat, závislých na platformě a aplikaci, je zvýšení bezpečnostního rizika. Zavedeme-li například jako obsah dat SQL příkaz, je možno modifikací SQL způsobit rozsáhlé škody (neoprávněné získání dat, zlovolné zrušení či modifikace dat).

5.1.5.4. Elektronický podpis schémat

Pravidla

Schémata ISVS MOHOU být digitálně podepsána. Digitální podpis, je-li přítomen, MUSÍ být ve formátu XML Signature. Element XML Signature MUSÍ být vložen do elementu `appinfo`, který je dceřiným elementem elementu `annotation`. Element `annotation` MUSÍ být vložen jako dceřiný element elementu `xs:schema`. JE DOPORUČENO, aby v elementu XML Signature byl nesen kompletní certifikát, svázaný s použitým privátním klíčem, a to včetně veřejného klíče.

Vysvětlení

Digitální podpis schémat ISVS slouží k jednak k ověření identity autora schématu (autentizaci), jednak k potvrzení integrity schématu. Je-li digitální podpis úspěšně ověřen, má uživatel schématu jistotu, že schéma nebylo někdy v minulosti náhodně či úmyslně změněno.

Digitální podpis ve formátu XML Signature je vložen do podepisovaného dokumentu; jedná se tedy o obalený podpis (enveloped signature) ve smyslu dle standardu XML Signature [14].

5.1.5.5. Použití nedokumentovaných schémat

Pravidla

Schémata ISVS se MOHOU v případě potřeby vyskytovat ve dvou formách, a to v plně dokumentované formě a ve formě bez dokumentace. V případě odlišností má přednost dokumentovaná verze schématu.

Vysvětlení

Plně dokumentovaná schémata mohou být poměrně rozsáhlá; porovnáme-li pro schéma ISVS, dokumentované podle doporučení tohoto materiálu, na jedné straně rozsah metadat o schématu (včetně digitálního podpisu) a na straně druhé rozsah informací o vlastnostech popisovaných komponent, jednoznačně několikanásobně převažují metadata o schématu.

Schémata jsou využívána pro validaci instancí XML dokumentů a z tohoto důvodu k nim bude poměrně často přistupováno po komunikační síti. Zmenšení schémat odstraněním dokumentace



a digitálního podpisu vytvoří formu schématu, využitelnou pro validace a nevyžadující tak intenzivní síťový provoz.

5.1.6. Doplnková validace

Pravidla

Schémata ISVS MOHOU deklarovat přídavnou validaci instancí XML dokumentů. DOPORUČENÝ způsob doplňkové validace je validace pomocí jazyka Schematron. DOPORUČENÝ způsob přiřazení schématu Schematronu k instanci XML dokumentu je vložení kódu schématu Schematronu do elementu `appinfo` příslušného XML schématu.

Vysvětlení

Ačkoli schémata jsou velmi silným nástrojem pro validaci XML dokumentů, nelze jimi postihnout některá často požadovaná omezení. Nelze například ověřovat závislost hodnoty jednoho elementu na hodnotě jiného elementu, tedy například výrok „Jestliže `element1` nabývá určité hodnoty, musí `element2` nabývat hodnoty z definovaného oboru hodnot“. Pro tyto účely je vhodné validaci schématem doplnit o validační postup, schopný postihnout podobná omezení.

Jazyk Schematron je XML technologie, velmi vhodná pro tyto účely. V současné době je Schematron, spolu s dalšími jazyky popisu struktur XML dokumentů, zařazen do návrhu standardu DSDL (Document Schema Definition Language). Standard DSDL by měl komplexně postihnout problematiku validace XML dokumentů.

Vhodným místem pro umístění schémat Schematronu je element `appinfo` elementu `documentation` XML schémat. Je možné vkládat jednak kompletní schéma Schematronu do dokumentace celého XML schématu, jednak je možno (je-li to vhodné) vkládat části kódu Schematronu do dokumentace jednotlivých datových typů.

Příklad

V následujícím výpisu je do schématu obecných datových typů ISVS vloženo vzorové schéma Schematronu:

```
<xs:schema targetNamespace="urn:cz:isvs:micr:schemas:CommonTypes:v1"
xmlns:cct="urn:cz:isvs:micr:schemas:CoreComponentTypes:v1" xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="urn:cz:isvs:micr:schemas:CommonTypes:v1"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.02a">
  <xs:annotation>
    <xs:documentation xml:lang="cs">Návrh XML schématu s definicemi obecných jednoduchých datových prvků
ISVS (Informačních systémů veřejné správy) verze 1.0.2. Toto schéma obsahuje definice jednoduchých datových
prvků, používaných pro ostatní schémata ISVS včetně ostatních schémat jednoduchých datových prvků a v rámci
celého systému ISVS ČR© ASD Software s.r.o., 2000 – 2002</xs:documentation>
  <xs:appinfo>
    <sch:schema xmlns:sch="http://www.ascc.net/xml/schematron">
      <sch:pattern name="Vzorová šablona Schematron">
        <sch:rule context="Test">
          <sch:assert test="Test">Element Test chybí.</sch:assert>
          <sch:report test="Test">Element Test je přítomen.</sch:report>
          <sch:assert test="@name">Element Test nemá atribut "name".</sch:assert>
          <sch:report test="@name">Element Test obsahuje atribut "name".</sch:report>
        </sch:rule>
      </sch:pattern>
    </sch:schema>
  </xs:appinfo>
</xs:schema>
```



```
</sch:schema>  
</xs:appinfo>  
<xs:appinfo>  
<rdf:RDF>
```

.....

5.1.7. Schémata a zprávy

Pravidla

JE DOPORUČENO, aby každé schéma zprávy, popisující přenášenou XML zprávu ISVS, obsahovalo deklaraci pouze jednoho typu (jedné třídy) instancí XML dokumentů.

Vývojáři MOHOU vyčlenit deklarace zvolené množiny datových typů schémat zpráv do separátního schématu. Vyčleněné schéma BUDOU importovat koncová schémata zpráv.

Vysvětlení

Je sice možné vytvářet schémata, která budou popisovat více tříd XML dokumentů; z důvodu přehlednosti, robustnosti, opakované použitelnosti a snadné údržby je však vhodné zvolit variantu *jedna zpráva – jedno schéma*.

Pokud bude mít skupina zpráv velmi podobná schémata, je možné společně použitelné komponenty těchto schémat vyčlenit do schématu společných komponent a datových typů. Toto „schéma typů“ mohou importovat nebo zahrnout všechna příslušná schémata zpráv. Zmíněný přístup je vhodný zvláště s ohledem na předpokládaný vývoj, kdy mohou přibývat typy zpráv a kdy je vhodné mít deklarace datových typů odděleně od deklarací zpráv.

Extrémním (ale velmi výhodným a s úspěchem používaným) případem je vyčlenění *všech* jednoduchých datových typů a *sdílených* složených typů do schématu typů.

5.2. Vývoj a správa schémat

Nedílnou součástí standardizace v oblasti přenosu dat je také stanovení organizační struktury, zahrnující subjekty, zúčastněné v procesu vytváření schémat. Transparentní organizační struktura s jasně vymezenými rolmi a odpovědnostmi jednotlivých subjektů je základním předpokladem úspěchu.

5.2.1. Správce schématu, správce dat

Pravidla

Každé schéma ISVS MUSÍ mít ustanoveno správce schématu a správce dat. Obsahuje-li schéma dokumentaci Dublin Core, správce schématu MUSÍ být uveden v elementu `creator` a správce dat MUSÍ být uveden v elementu `publisher`.

Vysvětlení

Organizační schéma zahrnuje několik typických rolí. Jsou to především správce schématu, který je zodpovědný v konečném důsledku za vývoj schématu a za faktickou správnost schématu, a správce dat, který je zodpovědný za publikaci a dostupnost schématu.



5.2.2. Publikace schémat

Pravidla

Každé schéma ISVS MUSÍ být publikováno prostřednictvím Metainformačního systému ISVS. Také dokumentace ke schématu a další informace, vztažené ke schématu, BY MĚLY být pro účely referenčního rozhraní ISVS uloženy v Metainformačním systému ISVS.

Vysvětlení

Předpokládá se, že schémata ISVS budou publikována prostřednictvím Metainformačního systému ISVS. Pokud budou schémata ISVS fyzicky umístěna v rámci MS ISVS, bude správce dat MS ISVS také správcem dat ve smyslu této metodiky schémat ISVS. Pokud bude rozhodnuto, že konkrétní schémata budou umístěna fyzicky mimo MS ISVS, musí být pro dané schéma ustanoven také konkrétní správce dat.

Pro konkrétní schémata mohou existovat i další dokumenty (dokumentace, RDDDL soubory, transformační šablony pro publikační účely apod.). Také tyto zdroje (či odkazy na ně) by se měly nacházet v MS ISVS. Předpokládá se, že v rámci MS ISVS bude zpřístupněna služba, která na základě dotazu pomocí URN jmenného prostoru schématu zpřístupní veškeré dostupné informace o schématu.

5.2.3. Proces tvorby schémat, vazby na procesy přenosu dat

Pravidla

Tato kapitola má pouze informativní charakter.

Vysvětlení

Vytváření schémat je úkolem jak pro experty z oblasti XML, tak pro širší odbornou veřejnost. Při projektování konkrétního datového slovníku je vhodné dojít ke shodě se všemi potencionálními uživateli. Proto je nutno k této skutečnosti přihlížet a při tvorbě datového slovníku k vývoji přizvat všechny potenciální účastníky komunikace. Všichni účastníci by měli dostat možnost vznést své požadavky na obsah a omezení, spojené s vytvářeným slovníkem.

Je zřejmé, že neexistuje vyčerpávající výčet schémat, které by měly být součástí rozhraní IS. Takový seznam bude podléhat neustálému vývoji a pokoušet se vytvořit datové slovníky XML rozhraní ISVS jednorázově a v jednom projektu je nereálné. Proto je nutno pro vytváření XML rozhraní IS přijmout model *postupného distribuovaného vývoje*. Postupný znamená, že není třeba zprovoznit všechny komponenty jednorázově; distribuovaný pak značí relativní samostatnost pracovních skupin, zodpovědných za vývoj jednotlivých schémat.

Vytvářené datové slovníky budou sloužit hlavně k definici rozhraní informačních systémů. Na základě dohody je nutno předem definovat všechny známé termíny, asociace a omezení a zahrnout je do dokumentace rozhraní spolu s jejich důvody, zdroji a způsoby použití. Není možné omezit dokumentaci k rozhraní na pouhá schémata, byť popisná síla schémat je značná. Je nutné použít také některý způsob dokumentace procesů. Velmi vhodná je dokumentace procesů pomocí některého procesního modelovacího jazyka. Vytvořený procesní model by měl zahrnovat jak části s XML technologiemi, tak ty non-XML části rozhraní, které jsou nutné pro deklaraci



požadované funkce rozhraní. Velmi mnoho systémů používá XML jako část své technologie, avšak analýza musí být provedena vždy kompletně.

Klíčovou komponentou celého systému však zůstává vrstva schémat jednoduchých a složených datových typů ISVS (viz [obr.1](#)). Bez standardních datových typů nelze vybudovat spolehlivý systém výměny informací ve formátu XML zpráv mezi účastníky komunikace v rámci ISVS.

5.2.4. Životní cyklus vývoje schémat

Pravidla

Životní cyklus schématu se skládá z následujících fází:

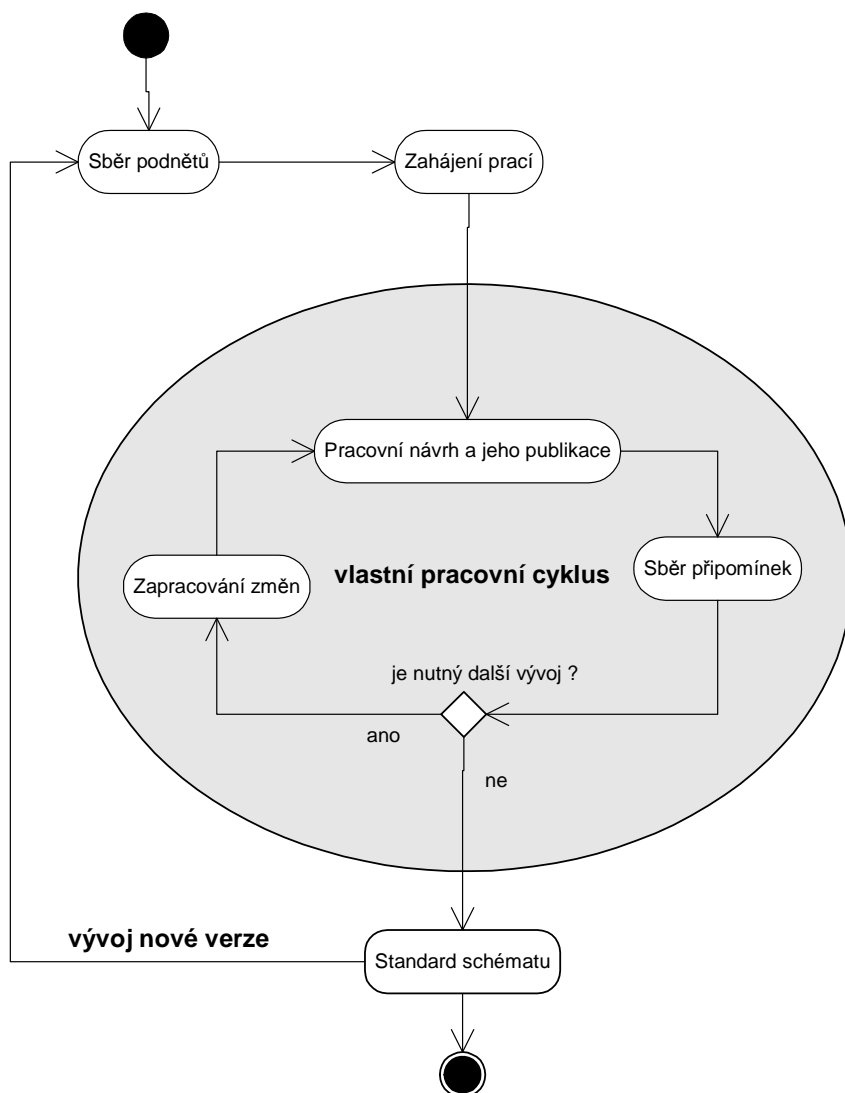
- **Sběr podnětů (*Request for Proposals*).** Tento proces je *veřejný*. Pokud vyvstane potřeba nového schématu, každý subjekt (např. organizace, instituce) může definovat své požadavky na vytvoření nového schématu.
- **Zahájení prací, Návrh schématu.** Proces je *neveřejný*. Došlé náměty jsou posouzeny a pouze schválený námět je „materializován“ do návrhu schématu.
- **Publikace návrhu, Sběr připomínek (*Request for Comments*).** Námět je publikován a je k dispozici ke komentování. Je také definován termín pro ukončení sběru komentářů. Proces je *veřejný*.
- **Rozhodnutí.** Proces je *neveřejný*. Došlé komentáře jsou využity nebo nevyžity. Pokud z komentářů vyplývá nutnost změny návrhu, je vytvořen a publikován nový návrh schématu. V opačném případě je publikován konečný standard příslušné verze schématu.
- **Požadavek změn.** Proces je *veřejný*. Každý subjekt (např. organizace, instituce) má právo definovat své náměty na změny standardu schématu. Požadavky změn jsou průběžně posuzovány, v případě shledání potřeby změny je zahájen proces změny schématu.

Vysvětlení

Při vývoji schémat je nutno věnovat pozornost také časové ose:

- Tvorba schémat není jednorázový, ale přinejmenším střednědobý proces.
- Náměty schémat by měly být podrobeny široké diskusi.
- Sada schémat daného rozhraní není konstantní. Je možno očekávat vývoj jak v kvalitě, tak v počtu implementovaných schémat.

Vývoj schémat probíhá v cyklu, naznačeném na obr. 3:



OBR. 3 – Model vývojového cyklu schématu

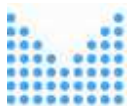
5.2.4.1. Podpora starších verzí schémat

Pravidla

Pro schémata ISVS MUSÍ být zajištěna dlouhodobá podpora předešlých verzí. Bez rozhodnutí oprávněných osob NESMÍ být zastavena podpora jednotlivých verzí schémat ISVS.

Vysvětlení

Údržba verzí schémat dokumentů může být zajištěna modelem, kdy každé nové oficiální verzi, nekompatibilní s předchozí verzí, bude přidělen nový cílový jmenný prostor (viz též kapitolu 0 tohoto dokumentu). Důvodem pro toto řešení je možnost používání starších verzí schémat.



5.2.4.2. Metodická podpora životního cyklu schémat

Pravidla

Návrhy schémat ISVS BY MĚLY být podávány prostřednictvím Informačního systému o datových prvcích (ISDP). ISDP BUDE metodicky řídit vývojový cyklus schémat ISVS.

Vysvětlení

ISDP je systém, uchovávající informace o standardizovaných jednoduchých i složených datových prvcích ISVS. Existence ISDP je dána zákonem č. 365/2000 Sb., o informačních systémech veřejné správy a o změně některých zákonů. [1]. ISDP by měl kromě prosté evidence uvedených datových prvků poskytovat i možnost detailního sledování vývojového cyklu datových prvků, tedy i schémat.



6. Průvodce vývojem schématu

Tato kapitola je popisem možného postupu při vývoji nového schématu. Týká se především schémat zpráv (viz 4.2.). Nemá charakter závazného doporučení a je pouze příkladem, který může být použit kompletně, částečně nebo může být zcela odmítnut. Je určena k dalšímu vývoji; jakékoli doplnění, připomínky či návrhy změn jsou vítány.

Kapitola vychází z materiálu *Draft Federal XML Developer's Guide* [17]

6.1. Jednotlivé kroky vývoje schémat zpráv

Jednotlivé kroky vývoje schématu zprávy mohou být postiženy sekvencí kroků, uvedených dále. Důrazně je doporučeno, aby fáze syntakticky neutrálního modelování (kroky 1-9) nebyla směřována s fází vlastního vytváření schématu (kroky 10-11):

1. Analýza procesů výměny dat. Je vhodné prozkoumat, která data jsou vyměňována mezi službou a jinými aplikacemi, uživateli a také mezi službou a vlastní databází. Analýza by měla zahrnout i identifikaci komunikujících stran.
2. Zjištěné informace by měly sloužit k vytvoření UML modelu procesu výměny dat (Use-Case model), identifikujícího účastníky komunikace (osoby, organizace, informační systémy) a jejich role v komunikaci.
3. Na základě modelů, získaných v předchozím kroku (2), a informací z kroku (1) je jasné, jaké zprávy jsou přenášeny v rámci komunikace. Pro tyto zprávy by měly být vytvořeny UML modely přenášených dat. UML modely jsou preferovány jednak pro jejich univerzálnost a syntaktickou neutralitu, jednak pro možnost přímého generování XML schémat z UML modelů pomocí některých softwarových nástrojů.
4. Pro všechny datové prvky (jednoduché i složené) v navržených UML modelech je třeba posoudit, zda již neexistují v některých předchozích řešeních (a to jak v řešeních ISVS, tak v některých specifických případech i v jiných řešeních). K podpoře vyhledávání existujících komponent by měl sloužit Metainformační systém ISVS.
5. Existuje-li použitelná definice datového prvku v některých předchozích řešeních, měl by být datový prvek založen na této definici. Postup, navrhovaný dále v bodech 6, 7 a 9, bude redukován na kroky, povolené v oddílech 5.1.4.2 a 5.1.4.3 této metodiky.
6. Všechny navržené datové prvky v UML modelech by měly být pojmenovány v souladu s oddílem 5.1.4.1 tohoto dokumentu.
7. Pro všechny navržené datové prvky v UML modelech je třeba posoudit, zda nebudou obsahovat metadata. Pokud existuje potřeba metadat, je nutno rozhodnout, zda se jedná o metadata, která se budou vyskytovat v instanci XML dokumentu, nebo zda se budou metadata vyskytovat pouze ve schématu. V případě, že se jedná o metadata v instanci dokumentu, je třeba je navrhnout (v kroku 10 a 11) jako XML atributy.
8. Pro všechny navržené datové prvky v UML modelech je třeba posoudit, zda se v kontextu zprávy nevyskytují opakovaně či zda nejsou členy nějaké obecné skupiny. Sdílené prvky



je třeba navrhovat jako globální datové typy, příslušnost do obecné skupiny prvků je možno postihnout (v kroku 10 a 11) modelem abstraktních elementů XML Schema.

9. Existují-li pro navržené datové prvky (složené i jednoduché) nějaká obecně přijímaná pojmenování, je vhodné je připojit k metadatům daného prvku.
10. Je-li dostupný nástroj pro generování schémat, je možno generovat schéma. Vygenerované schéma je však nutno podrobit pečlivé analýze a upravit jej tak, aby bylo v souladu s výše uvedenými body 5, 7, 8 a 9 a s oddíly 5.1.3 a 0 této metodiky.
11. Neexistuje-li dostupný nástroj pro generování schémat, je nutno vytvořit schéma ručně. Postup by měl vycházet z oddílů 5.1.3 a 0 této metodiky.
12. Závěrečné úpravy schématu by měly směřovat k přizpůsobení designu případným následujícím požadavkům, např.:
 - Rozšiřitelnost dokumentů ve smyslu datového obsahu – dokument může obsahovat i doplňková data, nepopsaná schématem.
 - Přizpůsobení schématu textovému či datovému charakteru výsledných instancí XML dokumentů.
 - Časté změny schématu – některé typy schémat mohou podléhat periodickým změnám a design je nutno přizpůsobit této skutečnosti.
13. Validitu hotových schémat je vhodné ověřit více parsery; přinejmenším je vhodné testovat schémata na těch parserech, které přicházejí do úvahy pro ten typ komunikace, jejíž datový model schéma popisuje.

Konkrétní opatření k docílení uvedených požadavků se často liší případ od případu a vysvětlení přesahuje rámec této metodiky. Pro tyto účely doporučujeme publikace [15]



7. Změny

7.1. Změny oproti předchozí verzi

Formální úpravy v souvislosti s přechodem kompetenzí z Ministerstva informatiky na Ministerstvo vnitra.

7.2. Změnové řízení

Status	Datum	Popis	Garant	Schválil
Verze 1.00	2.5.2006	První verze metodického pokynu	odbor koncepce a koordinace ISVS	ředitelka odboru
Verze 1.01	20.3.2009	Schválená verze metodického pokynu určená pro zveřejnění	odbor koncepce a koordinace ISVS	ředitelka odboru