

Web ČKR: návrh a realizace (případová studie)

Šárka Ocelková, ÚVT MU

Jak to všechno začalo?

Přibližně před jedním a půl rokem se na tým tvůrců www stránek Masarykovy univerzity (dále jen tvůrce) obrátila Česká konference rektorů (dále jen zákazník) s požadavkem na vytvoření vlastní www prezentace. Zákazník měl pouze orientační představu o tom, jak by prezentace měla vypadat, jasná byla pouze hlavní témata, která by prezentace měla obsahovat. Co se pod nimi bude skrývat a jakého budou rozsahu, zdaleka nebylo odhadnutelné, od samotného počátku byl však jasný požadavek na jednoduchou modifikaci, levné řešení a jednotný design. V souladu s tímto požadavkem bylo navrženo a následně realizováno řešení webu ČKR¹, které přiblížíme čtenářům v tomto článku (a jeho pokračování v příštích číslech Zpravodaje) ve formě zobecněné případové studie popisující návrh a realizaci jednoduchého informačního systému pomocí webových technologií

Stojí-li obecně tvůrce před úkolem vytvoření nového webu, vždy je dobré nejprve zanalyzovat situaci a požadavky, důležité je pak rozdělení odpovědností za jednotlivé části webu. Po té přijdou na řadu úvahy o možnostech řešení a konečná volba technologie. Následně je třeba web již „jen“ vytvořit a zajistit jeho průběžnou aktualizaci.

Základní okruhy problémů při návrhu a realizaci webu jsou tedy následující:

1. *Dělbba práce* - jak rozdělit jednotlivé činnosti při tvorbě webu a poté při jeho provozování (aktualizaci dat) mezi různé skupiny pracovníků tvůrce i zákazníka
2. *Volba metody tvorby www-stránek* - jakým způsobem převést „syrová“ data informačního systému do podoby HTML-stránek (existuje vícero možností - od ručního zapisování HTML-souborů až po jejich plně automatizované generování pomocí za tímto účelem vytvořených programů)

¹<http://crc.muni.cz/>

3. *Volba způsobu uložení dat* - zvolit jako úložiště dat informačního systému kolekci textových souborů (obsahujících například data strukturovaná ve formátu XML) nebo je lepší uložit data do databázového systému?
4. *Realizace webové prezentace* - což je hlavní část řešení, obsahující celou řadu dílčích podokruhů otázek, např.:
 - automatizované generování www-stránek;
 - návrh a realizace vnitřní architektury webu;
 - uživatelské „administrativní“ rozhraní;
 - a další.

1 Dělbba práce

Na www prezentaci lze pohlížet jako na tři samostatné části (datový obsah, technickou realizaci a grafiku), u nichž se vyplatí dobrá analýza a jež mohou z počátku řešit tři samostatné vývojové skupiny. Až posléze se tyto části spojí a vznikne úplný web. Již na počátku je však dobré si ujasnit, kdo bude za kterou z oněch částí odpovídat, určit tedy konkrétní *odpovědnou osobu*. Podívejme se nyní blíže na jednotlivé části a na to, jakým způsobem byly v případě webu ČKR rozděleny do kompetencí zúčastněných stran:

1.1 Datový (informační) obsah

Odpovědná osoba - *správce dat* nebo *správce informací* - může být odborník na informační technologie, ale často jím bývá i laik. S touto možností je dobré předem počítat, neboť úkolem správce je především trvale dohlížet na aktuálnost údajů, a proto mu musí být umožněn přístup k datům na odpovídající úrovni. Představa informačního obsahu a charakteru webu je nutným základem pro následující dvě části (technickou realizaci a grafiku), proto musí správce dat hned na počátku sdělit alespoň základní představu o tom, jaké informace se na webu budou prezentovat, jaký je předpokládaný objem a jakého budou typu (textové dokumenty, obrázky nebo přehledové informace např. zaměstnanců, publikací apod.). Od toho se dále odvíjí rozsah celého webu (desítky, stovky nebo tisíce stránek). Dobré je také vědět předem, jak častý bude požadavek na aktualizaci dat a zda bude prováděn

ručně nebo automaticky, např. přenosem z externího zdroje dat.

V případě ČKR se role správce dat ujal sám zákazník. Od počátku bylo jasné, že na webu ČKR budou vystavovány více méně jen textové dokumenty a přehledy (statut, zápisy ze zasedání, usnesení, adresář, seznam členů, historie apod.), a tak se zákazník jakožto autor textů ujal i další role – definice vlastního vzhledu jednotlivých dokumentů prezentovaných na www stránkách.

1.2 Technická realizace

Osoba odpovědná za technickou realizaci – *správce systému* – bývá téměř výhradně odborník na informační technologie. Jeho úkolem je na základě informací od správce dat navrhnout příslušnou technologii, celý web zrealizovat, nadále udržovat a odpovídat za jeho trvalý a bezchybný chod. Důležitá pro volbu technologie je rovněž informace, zda prezentovaná data budou veřejná nebo přístupná autentizovaně.

Technickým realizátorem webu ČKR byl a stále je již na počátku zmíněný tvůrce. Rozhodující vliv na volbu technologie měl požadavek na aktualizaci obsahu (prakticky ihned po změně) a skutečnost, že správcem dat je laik. Jinak řečeno, bude-li zákazník potřebovat jakoukoliv změnu ve vlastním obsahu www stránek (od obyčejného překlepu ve slově až po vystavení dalšího nového dokumentu), nebylo by únosné, aby pokaždé musel o doplnění/aktualizaci stránek žádat tvůrce. Musí tedy existovat jednoduchý způsob, jak si zákazník bude moci do obsahu stránek zasahovat sám, ale tak, aby nedopatřením neohrozil celou prezentaci. K jakému řešení tvůrce nakonec dospěli, je obsahem dalších kapitol tohoto článku.

1.3 Vzhled stránek (grafika)

Odpovědnou osobou by měl být v ideálním případě zkušený *grafik*, který nejen navrhne vizuální podobu stránek, ale také ji převede do rozumně použitelné (elektronické) podoby. O takové tvůrčí osoby bývá bohužel nouze. Grafickou podobu stránek ale rozhodně není dobré podcenit. Grafika dává tvář celé www prezentaci a právě ta určuje, jak bude prezentace přehledná a jak s ní budou uživatelé rádi pracovat.

V oblasti grafiky lze v případě ČKR mluvit o štěstí, neboť mezi tvůrci se podařilo najít velmi schopného grafika a zároveň informatika, takže veškeré grafické návrhy mohly být od počátku tvořeny hned v elektronické podobě. Byla také učiněna dohoda, že zákazník musí grafickou podobu schválit, a jakoukoliv následnou změnu již schválené grafiky ze strany tvůrce musí zákazník opět vždy schválit.

2 Metody tvorby www stránek

Nejprve bylo třeba utvořit si představu o konečném rozsahu www prezentace. Po několika konzultacích se zákazníkem vyplynulo, že se bude jednat o web středního rozsahu – několik desítek www stránek – přičemž (vyjma úvodní stránky) budou všechny ostatní stránky ve stejném grafickém provedení. Z toho je také patrný postoj k možným řešením. Z nejobecnějšího hlediska existují v podstatě dvě možnosti, jak vytvářet www stránky:

2.1 „Ruční“ psaní stránek

V dnešní době možností nejrůznějších technologií by toto řešení bylo krajně neefektivní. Každá www stránka prezentace je napsána přímo v jazyce HTML a je statické povahy. Je-li navíc u prezentace požadován jednotný vzhled všech stránek, pak je zásadní nevýhodou tohoto řešení opisování téhož formátovacího kódu do každé stránky. Při požadavku na sebemenší drobnou změnu vzhledu je nutné projít všechny stránky a všude provést příslušné úpravy. Tuto situaci lze částečně řešit použitím SHTML² a tedy vyčleněním společného formátovacího kódu do samostatného souboru, nelze tak již ale plnohodnotně a obecně řešit vnitřní formátovací prvky, jako např. vzhled tabulek, odkazů apod.

Tato varianta je nevýhodná i pro zákazníka. Bude-li sám provádět obsahové změny ve stránkách, nejen že by musel znát jazyk HTML, ale

²SHTML (Server Side Includes HTML) je používáno jako přípona souboru s HTML značkami, který je obohacený o odkazy na jiné HTML soubory, jež se mají zahrnout do výsledné odpovědi www serveru.

1. Prof. Ing. Jan Novák, CSc., rektor Univerzita Karlova v Praze předseda České konference rektorů Václavské nám. 1 100 00 Praha 1	tel.: 123 456 789 987 654 321 fax: 123 456 789 e-mail: jan.novak@abc.cuni.cz crc.predseda@muni.cz
2. Prof. MUDr. PhDr. Jana Nováková, CSc., rektorka Univerzita Palackého v Olomouci Křížkovského 7 700 07 Olomouc	tel.: 666 555 444 333 222 111 fax: 999 888 777 987 987 987 e-mail: novakova@xyz.upol.cz
3. atd.	

Obrázek 1: Ukázka adresáře

hrozí zde reálné nebezpečí neúmyslného překlepu a tím i (byť dočasné) grafické znehodnocení stránky. Proto také byla tato možnost téměř okamžitě vyloučena.

2.2 Generování stránek

V tomto řešení stojí za každou stránkou webu skript/program, který ji na vyžádání vygeneruje. Můžeme rozlišit dvě varianty generování stránek: on-line a off-line generování. Společnou výhodou obou možností je skutečnost, že veškeré formátovací prvky stránky (jednotný vzhled, navigační prvky, ...) mohou být uloženy na jediném místě, v programu je rovněž možné ohlídat většinu chyb a překlepů. Výhodné také je, jsou-li data oddělena od formátovacího kódu stránky. Tvorba skriptu přirozeně vyžaduje znalost programování, proto se generování stránek „vyplatí“, má-li web alespoň nějakou tu desítku stránek, což plánovaný web ČR splňoval.

On-line generování. Každá stránka je vygenerována ihned na vyžádání uživatele. To přináší obrovskou výhodu možnosti variability a parametrizace stránek, jediný skript tak může na základě obměny parametru generovat tisíce stránek stejného typu a vzhledu, jen s jiným požadovaným obsahem.

Off-line generování. Stránka je v tomto případě vygenerována a následně uložena do statického HTML souboru, díky tomu je zde podstatně menší možnost variability a parametrizace. Zpravidla se negeneruje jen jedna stránka, ale celá kolekce stránek, proto musí

u tohoto řešení někde existovat (např. v konfiguračním souboru nebo v databázi) seznam všech skriptů, které se mají požadovat, seznam všech možných parametrů k těmto skriptům a seznam cílů, kam mají být výsledné vygenerované stránky uloženy (podrobněji popsáno také ve [2]). Tento mechanismus pak může být automaticky načasován, a celý web se tak dávkově najednou zaktualizuje. Vzhledem ke skutečnosti, že se generuje a ukládá tolik stránek, kolik je použitých hodnot parametrů, není tento způsob vhodný pro příliš rozsáhlé weby (cca do několika málo set stránek).

Jelikož se při přegenerování může vyskytnout nějaká chyba (ať již ve skriptu generujícím stránku nebo nějaká jiná), je třeba zamezit situaci, kdy se vygeneruje jen část stránek a web je pak neúplný a chybný. Pro tento účel se tvůrcům již osvědčilo řešení tzv. *transakčního přegenerování*, zjednodušeně řečeno, celý web se vygeneruje do jiného adresáře, než je aktuální adresář webu, a až v případě korektního vygenerování všech stránek se pak tyto adresáře „vymění“.

Ze dvou uvedených variant byla vybrána ta „bezpečná“ z pohledu uživatele, tj. zákazníkovi byl zabráněn přístup do části určující vzhled a formátování stránek, tím je eliminováno možné grafické znehodnocení v případě překlepu.

3 Způsoby uložení dat

Data jsou základem každého webu a jako taková musí být vždy někde uložena. Možností jejich

Usnesení 50. zasedání České konference rektorů
Praha, 19. 10. 2000

Česká konference rektorů (ČKR) přijala na svém 50. zasedání následující usnesení:

1. ČKR zhodnotila
2. a) předkládaný návrh rozpočtu vysokých škol
b) podíl navrhovaného objemu finančních prostředků pro vysoké školy
- 3.

V Praze dne 19. října 2000

Za Českou konferenci rektorů
Prof. Ing. Jan Novák, CSc.
předseda

Obrázek 2: Ukázka usnesení

uložení není mnoho, obecně nejpoužívanějšími úložišti jsou buď soubor (jak strukturovaný, tak nestrukturovaný) nebo databáze (vždy strukturovaná). Každé řešení má své výhody a nevýhody, na něž se nyní podíváme podrobněji:

- *Nestrukturovaný soubor.* Data jsou uložena v souboru společně s kódem, který formátuje stránku (např. HTML nebo SHTML), v tomto případě není dodržována žádná jednotná datová struktura. K vytvoření obsahu takového souboru postačí pouhá znalost jazyka HTML. Sloučení dat s kódem nese samozřejmě velké riziko chyb, které mohou stránku (buď dočasně) znehodnotit.

Tento způsob je vhodný pro malé (rozsahem stránek i objemem dat) weby, kde by byla investice do větších řešení zbytečná a neekonomická. Jelikož u webu ČKR se předpokládaly desítky až několik málo stovek stránek, prakticky ihned byla tato možnost vyloučena.

- *Strukturovaný soubor.* Data jsou oddělena od formátovacího kódu stránky a uložena v předem definované struktuře do samostatného souboru. V dnešní době se k zápisu strukturovaných dat nabízí formát XML. K tomu je ovšem potřeba naprogramovat další software, v tomto případě XSL transformaci, jež bude data převádět do požadovaného výsledného HTML. Velkou výhodou tohoto řešení je snadné zpřístupnění dat jejich správcům (např. zpřístupnění části disku s datovými zdroji) a snadná možnost editace v běžném textovém editoru nebo specializovaném

XML editoru, obzvláště je-li správcem dat laik. Další výhodou je přehledný zápis, velmi vhodný pro strukturované texty (např. zápisy, výroční zprávy, články ve sbornících apod.). Toto řešení se stává nevýhodným v případě velkých objemů dat (tisíců, milionů záznamů), kdy je již editace souboru nepřehledná a manipulace s tak objemným souborem může být pro běžné textové editory komplikovaná a pomalá.

- *Strukturovaná (relační) databáze.* Data jsou uložena nejčastěji v relační databázi. Z povahy databáze je zřejmé, že data jsou zde uložena vždy strukturovaně a jsou oddělena od formátovacího kódu stránky. K vytvoření www stránky je opět potřeba naprogramovat software – skript generující stránku. Velkou výhodou tohoto řešení je dynamičnost a možnost parametrizace stránek, snadná je hromadná manipulace s daty. Komplikovanější bývá zpřístupňování dat v databázi správcům dat. Pokud jím není zrovna odborník znalý jazyka SQL, je nutné mít naprogramovaný software pro editaci dat. Použití databáze není příliš vhodné pro udržování a vkládání strukturovaných textů (zápisy, zprávy, ...).

3.1 Databáze versus XML

Jak již bylo nastíněno v předchozí kapitole, uložení dat ve strukturovaném souboru nebo v databázi odděluje technickou realizaci (za kterou odpovídá řešitel) od vlastních dat (za které odpovídá zákazník). Na rozhodnutí, zda jako úložiště

Přehled usnesení	
...	
50. zasedání ČKR	(Praha, 19. října 2000)
49. zasedání ČKR	(Lednice na Moravě, 21.-22. září 2000)
48. zasedání ČKR	(Opava, 25.-26. května 2000)
47. zasedání ČKR	(Brno, 24.-25. února 2000)
...	

Obrázek 3: Ukázka přehledu usnesení

dat použít relační databázi nebo XML soubor, má vliv podrobnější analýza předpokládané podoby www stránek. Ukažme si proto dva typické příklady dat z ČKR: adresář a usnesení.

Adresář je strukturovaný seznam členů ČKR, jejich funkcí, kontaktů a vysoké školy na níž působí, včetně uvedení adresy. Ukázka adresáře je na obrázku 1.

Data obsažená v adresáři se však budou objevovat i na řadě dalších stránek, např. seznamy členů předsednictva, pléna a komor, historie členů – vše jsou ve skutečnosti jen určité podmnožiny adresáře.

Usnesení jsou textové dokumenty z jednotlivých zasedání ČKR, které mají všechny přibližně stejnou strukturu: záhlaví, jednotlivé body usnesení (mohou být i víceúrovňové) a zápatí. Usnesení ukazuje obrázek 2.

Informace ze záhlaví (zasedání, místo a datum) slouží současně jako podklady také k další stránce – přehled usnesení, ukázka je na obrázku 3.

Jak naznačují příklady, různé www stránky mohou obsahovat tatáž data. Odtud plyne, že na datové obsahy stránek nelze pohlížet jako na samostatné množiny dat, ale je nutné zohlednit v datové analýze právě již výše zmíněnou skutečnost vícenásobného výskytu týchž dat. Základní položky, které je nutné evidovat pro naše dva příklady, jsou:

Adresář. Budou potřeba evidence členů, kontaktů, funkcí a vysokých škol. Ke každému členu bude možné evidovat jméno, příjmení, tituly, kontakty, funkce a vysokou školu. Z obrázku 1 je patrné, že k jednomu členu může existovat více kontaktů, tj. telefonů, faxů nebo e-mailových adres. U vysokých škol bude kromě názvu uvedena také adresa, případně

URL www prezentace. V případě funkcí stačí pouze jejich název.

Usnesení. U každého usnesení bude potřeba evidovat pořadí (které je dáno pořadím zasedání, na němž bylo usnesení přijato), místo a datum konání, podpis a vlastní text usnesení, který může a nemusí být dále strukturován.

Nyní si ukažme, jak by vypadala struktura dat v případě uložení do relační databáze a do XML souboru.

3.2 Struktura dat v relační databázi

Adresář. Pro každou výše uvedenou evidenci (v databázové terminologii entitu) je rozumné uvažovat samostatnou tabulku, tj. tabulku pro členy, školy a funkce. Každá entita musí mít svůj jednoznačný identifikátor (primární klíč) id. Je-li pak potřeba u členu uvést vysokou školu, je v entitě člen uvedeno pouze `id_školy` (klíč do tabulky škola). Zcela obdobně je to s uvedením funkce. Zdánlivý problém může nastat s evidencí kontaktů. Jelikož je vztah mezi členem a kontaktem obecně 1:N, měla by ve správném návrhu databáze existovat další samostatná tabulka s kontakty, kde by např. nějaký atribut „typ“ říkal, zda se jedná o telefon, fax nebo e-mail a každý kontakt by byl vždy vázán na konkrétního členu uvedením `id_člena`. Tuto situaci je možné bez újmy na obecnosti zjednodušit tak, že u každého členu budou uvedeny atributy telefon, fax a e-mail, v případě vícenásobného výskytu budou jednotlivé kontakty odděleny např. čárkou. Strukturu databáze ukazuje následující schéma:

```
člen (id,
      titul_před_jménem, jméno, příjmení,
      titul_za_jménem, id_funkce_v_ČKR,
      komora, id_školy, funkce_ve_škole,
```

```

telefon, fax, e-mail)
škola (id, název, url_www_stránek,
      adresa_ulice, adresa_číslo,
      adresa_PSC, adresa_město)
funkce (id, název)

```

Usnesení. Na první pohled se může zdát, že usnesení je jediná entita, jejíž identifikátor id může být v podstatě pořadí zasedání, na kterém bylo usnesení přijato. Další položky jsou místo a datum konání (vztahuje se na zasedání) a konečně v jednom dalším atributu by mohl být uložen celý text usnesení již naformátovaný HTML značkami. Pro zákazníka by ale toto řešení nebylo příjemné, neboť by se musel naučit jazyk HTML. Protože mají jednotlivá usnesení poměrně jednoduchou strukturu (jednotlivé body a podbody), lze tyto části uložit do samostatné tabulky. Musí být ale kromě identifikátoru usnesení, k němuž se část vztahuje, nutně uvedeno také pořadí části, její úroveň a typ číslování (číslem, písmenem nebo odrážkou). Jak by mohla struktura vypadat ukazuje následující schéma:

```

usnesení (id (=pořadí_zasedání),
          místo_konání, datum_konání)
usnesení_část (id_části, id_usnesení,
              pořadí_části, úroveň,
              typ_číslování,
              text_části)

```

3.3 Struktura dat v XML souboru

Adresář. Obdobou databáze mohou být jeden nebo více XML souborů. V případě adresáře lze každou uvažovanou tabulku z předchozí kapitoly reprezentovat jedním souborem: člen, škola a funkce. Každé databázové entitě pak odpovídá hlavní element souboru: <ČLEN>, <ŠKOLA> a <FUNKCE>. Atributy tabulek odpovídají atributům hlavního elementu. Oproti databázi lze však pomocí XML velmi snadno řešit problém vícenásobných kontaktů - součástí každého elementu <ČLEN> budou vnořené značky <TELEFON>, <FAX> a <E-MAIL>, které budou u každého člena tolikrát, kolik se u něho eviduje telefonů, faxů nebo e-mailů. Struktura hlavních XML elementů je ukázána v následujícím schématu:

```

<ČLEN id=„...“ titul_před_jménem=„...“
      jméno=„...“ příjmení=„...“

```

```

      titul_za_jménem=„...“
      id_funkce_v_ČKR=„...“ komora=„...“
      id_školy=„...“
      funkce_ve_školě=„...“>
        <TELEFON> ... </TELEFON>
        <TELEFON> ... </TELEFON>
        <FAX> ... </FAX>
        <FAX> ... </FAX>
        <E-MAIL> ... </E-MAIL>
        <E-MAIL> ... </E-MAIL>
        ...
    </ČLEN>

```

```

    <ŠKOLA id=„...“ název=„...“
      url_www_stránek=„...“
      adresa_ulice=„...“ adresa_číslo=„...“
      adresa_PSC=„...“ adresa_město=„...“/>

```

```

    <FUNKCE id=„...“ název=„...“/>

```

Usnesení. Databázovou tabulku je opět možné jednoduše reprezentovat souborem usnesení s hlavním elementem <USNESENÍ>. Díky strukturovatelnosti XML je zde ovšem mnohem lepší možnost uložení vlastního textu usnesení, členěného na jednotlivé body a podbody. Příslušným vnořováním značek <BOD> (eventuálně <TEXT> pro vlastní text bodu) lze elegantně docílit toho, co v databázi jen velmi obtížně další tabulkou. Strukturu usnesení v XML ukazuje následující schéma:

```

<USNESENÍ pořadí=„...“
  místo_konání=„...“
  datum_konání=„...“>
  <BOD typ_číslování=„...“>
    <TEXT> ... </TEXT>
    <BOD typ_číslování=„...“>
      <TEXT> ... </TEXT>
    </BOD>
    ...
  </BOD>
</USNESENÍ>

```

Pokračování příště ... □