

## KOMPONENTOVÝ ROZKLAD UŽIVATELSKÉHO ROZHRANÍ WEBOVÉHO INFORMAČNÍHO SYSTÉMU

H. Netrefová

**Došlo: 29. června 2004**

### Abstract

NETREFOVÁ, H.: *Component decomposition of user interface of web information system*. Acta univ. agric. et silvic. Mendel. Brun., 2005, LIII, No. 3, pp. 281-288

At present, the usage of information systems increases in various areas of human activities, therefore it is necessary to put the emphasis on their quality and user friendliness. That is why new techniques of design, development and implementation of information systems are being sought. The author of the paper and her colleagues employ themselves in the domain of component oriented web information systems and their goal is to define the methodology for building this kind of system. The component boxing model was designed so far. It describes how to build-up the information system from the particular components at each level of the three-layer architecture in general. Within the individual levels, it is needed to carry out the detailed analysis of components and their dependencies. In the paper, the user interface is discussed. The page decomposition in particular components is analyzed in detail and the component types with relationships between them are sought. There are also terms as box, box content, box design, visual element, linked and doubly linked elements, elements linkage defined. Further way of this domain research that would lead in determination of the methodology for the information system development at the user interface and presentation level is shown as well as the integration of this particular task in the entire system development.

web information system, user interface, components, page decomposition

Informační systémy a informační technologie (IS/IT) mají v dnešní době své pevné místo v nejrůznějších odvětvích lidské činnosti. Dokonce i v oborech, kde se ještě před několika lety o počítačích ani neuvažovalo, dnes firmy přecházejí na elektronické zpracování mnohých agend a zavádějí podnikové informační systémy. Do styku s počítačově orientovanými informačními systémy přichází stále větší část populace. Je proto nutné, aby informační systémy byly kvalitní, z pohledu uživatelů intuitivní a jejich používání bylo skutečně přínosem.

V článku se zaměříme na oblast webových informačních systémů, neboť ty dnes mezi informačními systémy zaujímají výsadní postavení. Vezměme si například informační systémy na univerzitách: všechny – nejen české, ale i zahraniční univerzity provozují vlastní informační systém, který je základem pro stu-

dijní agendu a dále v sobě zpravidla integruje další subsystémy (personální subsystém, bezpapírovou kancelář, poštovní subsystém, systém pro správu dokumentů, kolejní, stravovací systémy a další). Míra integrovanosti univerzitního informačního systému je na jednotlivých univerzitách různá, v každém případě však informační systém představuje pracovní prostředek pro mnohé uživatele z řad zaměstnanců i studentů univerzity.

Návrh, vývoj a implementace komplexního informačního systému představuje velmi rozsáhlý projekt, na kterém se podílí mnoho odborníků různých profesí. Všichni musejí postupovat podle jednotné metodiky, která jasně definuje jednotlivé postupy, aby bylo dosaženo optimálního výsledku.

Současné informační systémy jsou zpravidla vyvíjeny klasickou metodou, kdy odpovědnost za zpraco-

vání vstupních dat a odeslání dat výstupních spočívá nejvíce na aplikačních programátorech, kteří ve svých aplikacích zajistí komunikaci jednak s uživatelským rozhraním a jednak s databází (samozřejmě ve spolupráci s datamanážery obsluhujícími datovou základnu systému a s designéry navrhujícími design).

Aby se vývoj informačních systémů zefektivnil, objevila se myšlenka oddělit od sebe jednotlivé komponenty systému, jasně je definovat a stanovit pravidla komunikace těchto komponent mezi sebou. Dnes je tato metodika více či méně propracována na různých úrovních složek či vrstev informačních systémů. Dosud však nebyla uplatněna pro vývoj webového informačního systému jako celku. Proto se s kolegy v rámci svých disertačních prací zabýváme návrhem komponentově generovaného webového informačního systému, kde výsledný systém bude tvořen na základě dílčích metodik nalezených a diskutovaných v jednotlivých pracích.

Informace poskytované informačním systémem jsou předávány, stejně jako veškerá komunikace uživatele se systémem, prostřednictvím uživatelského rozhraní systému. Jeho návrhu a realizaci je třeba věnovat velkou pozornost, aby splnilo nároky uživatelů a současně vhodným způsobem zapadlo do celkové koncepce a struktury informačního systému. Uživatelské rozhraní tvoří jednu kompaktní vrstvu systému a daným způsobem komunikuje se svým okolím. Vycházíme-li z hlavního cíle – definování uživatelského rozhraní, jeho komponent a pravidel – je nutné stanovit si řadu dílčích cílů. V tomto článku se zaměříme na rozklad uživatelského rozhraní informačního systému do jednotlivých komponent, budeme se snažit nalézt obecný popis jednotlivých prvků a v závěru nastíníme další směr vývoje hledané metodiky.

## MATERIÁL A METODY

Abychom mohli přistoupit k výkladu metodiky pro rozklad uživatelského rozhraní na jednotlivé komponenty, je třeba si vysvětlit základní principy nutné k pochopení popisované problematiky.

### Prezentační vrstva architektury IS

Naprostá většina současných webových (a nejen webových) IS je vyvíjena na základě třívrstvé architektury. V těchto IS lze snadno rozlišit datovou vrstvu představovanou některým relačním nebo objektovým databázovým systémem, dále aplikační vrstvu představovanou aplikacemi nebo komponentovými aplikacními servery a vrstvu prezentační, kterou před-

stavují různé technologie počínaje klasickými klient/server technologiemi až po tenké klienty v podobě webového prohlížeče. Jednotlivé vrstvy jsou na sobě vzájemně nezávislé a komunikují prostřednictvím definovaného rozhraní. Nejvyšší, prezentační, vrstva zajišťuje roli uživatelského rozhraní, zpracovává vstupní požadavky od uživatele a prezentuje výstupní informace uživateli. Obvykle existuje několik prezentačních vrstev pro různé druhy zařízení, platformy a prostředí. U webových informačních systémů může být tato vrstva tvořena aplikacemi designu pro formátování výstupu a lehkým klientem (prohlížečem internetových stránek).

### Událostní cyklus

Nejdůležitějším odlišujícím prvkem webových informačních systémů od ostatních systémů typu klient/server je práce na základě událostního cyklu. Server komunikuje s klientem (webovým prohlížečem) pomocí cyklu *otázka – odpověď*<sup>1</sup>, který je iniciován např. zadáním URL<sup>2</sup>, kliknutím uživatele na odkaz, odesláním dat ve formuláři apod. Otázkou je tedy požadavek vyslaný klientem serveru. Server pak zpracuje zasláná data, provede operace podle předaných parametrů a tvoří výstup, který zasílá zpět klientovi. Důležitou vlastností webového prostředí je též bezstavovost, kdy při komunikaci klienta se serverem jsou vždy znovu předávány veškeré parametry (Pazdziora, 2003).

V klasickém webovém informačním systému je odpověď nestrukturovaná a tvořená přímo aplikací, kde se formuje jak obsahová, tak designová stránka výstupu. Výsledný formát prezentovaných dat je tedy v rukou aplikačních programátorů.

Jak jsme se zmínili již v úvodu, výstupní data je vhodnější strukturovat. V rámci formování odpovědi se zpracovává odděleně aplikační a prezentační část cyklu *otázka – odpověď*. Aplikační programátoři se pak mohou věnovat pouze logice aplikační a formát a způsob zobrazení je řešen až ve vrstvě prezentační. Abychom oddělili obsah od způsobu prezentace dat, přecházíme ke *komponentově orientovanému systému*, odpověď zde můžeme rozdělit na tři podčásti:

- a) *Zpracování požadavků* předaných klientem. V této fázi probíhají akce, které mění stav systému (např. mění či vkládají data do databáze) na základě předaných „akčních“ parametrů. Nejsou zde generována běžná výstupní data, pouze informace o průběhu zpracování, které jsou ukládány do spe-

<sup>1</sup> v angličtině se používá výraz Request-response cycle

<sup>2</sup> Uniform Resource Locator – představuje adresu webové stránky

ciálních souborů či struktur. Mohou být vrácena chybová hlášení oznamující uživateli neúspěšný průběh operace.

- b) *Generování prostých výstupních dat* ve formátu nezávislém na způsobu prezentace.
- c) *Prezentace výstupního proudu* zformátovaného podle příslušných požadavků (design, uživatelské preference, typ klienta).

### Component boxing model

Dva výše uvedené principy (třívrstvá architektura IS a událostní cyklus) společně s teoretickými a praktickými zkušenostmi s vývojem webových informačních systémů v univerzitním prostředí byly podkladem pro návrh a následné vypracování metodiky pro komponentové generování webových informačních systémů. Metodika byla nazvána *Component boxing model (CBM)* podle dvou paradigmat, jež obsahuje (Šorm, Netrefová, Motyčka; 2004).

*Princip boxů (boxing principle)* pohlíží na stránku informačního systému jako na pravoúhlou oblast, dělenou do menších, opět pravoúhlých oblastí. Každý takovýto prvek (box) se skládá ze dvou částí – obsahu a designu. Obsahem může být buď další vnořený prvek nebo funkční komponenta (např. statický text, formulář, obrázek). Vytvořením hierarchické struktury pomocí vnořování jednotlivých boxů vzniká strom komponent. Při tvorbě odpovědi musíme postupovat od nejvnitřnějšího listu a podle daného předpisu sestavit výslednou stránku, ta je pak zaslána uživateli na výstup.

Zatímco princip boxů se zabývá vertikální linií (tzn. rozkladem otázky a složením odpovědi v rámci spolupráce všech tří vrstev architektury IS), *princip komponent (component principle)* se týká linie horizontální. Na úrovni každé vrstvy hledá a definuje konkrétní komponenty, které se v systému opakují či řeší standardizovanou úlohu, a usnadňuje tím jejich opětovné použití. Podle pravidel objektově orientovaného programování lze vytvořit strom komponent s jasně definovanými vlastnostmi a dědičnými závislostmi.

Při budování IS získáváme kombinací obou výše uvedených pravidel systém, který je znovupoužitelný (díky komponentovému pojetí), personalizovatelný (díky boxům) a snadno rozšiřitelný (kombinací obou vlastností).

### Uživatelské rozhraní

Dosud jsme hovořili o obecných vlastnostech in-

formačního systému a o metodice pro tvorbu nové generace webových informačních systémů. Nyní se podívejme podrobněji na problematiku uživatelského rozhraní.

Uživatelské rozhraní odráží z velké části efektivitu informačního systému, neboť představuje komunikační prostředí uživatelů se systémem. Čím je uživatelské rozhraní kvalitnější, tím se zvyšuje spokojenost uživatelů a obliba práce se systémem. Je třeba si uvědomit, že cílem koncového uživatele není počítač či jeho software, ale vhodné použití počítače pro řešení problému reálného světa. Musí se starat o to, jak správně interpretovat data poskytovaná počítačem, jak tato data využít jako podkladu pro svá rozhodování apod. (Voříšek, 1996).

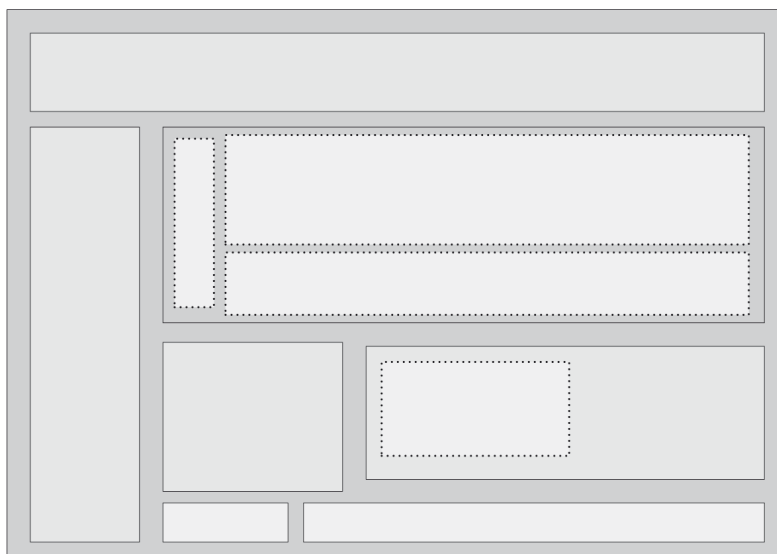
O designu webových stránek a jejich správném navrhování byla napsána řada publikací – např. (Satrpa, 1997), (Lynch, Horton; 2004) – a při návrhu designu informačního systému je třeba určitá pravidla dodržovat. Materiál zpracovávající designová doporučení je možné nalézt také v (Netrefová, 2003). V rámci naší práce je pro nás důležitá komponentová skladba stránky, abychom byli schopni definovat její obecné rozložení. Výhody pro uživatele plynoucí z komponentově pojatého designu stránky jsou diskutovány např. v (Netrefová, Šorm; 2003).

### VÝSLEDKY

#### Rozklad stránky

Princip boxů v component boxing modelu hovoří o rozkládání a opětovném skládání stránky z komponent. Prvním předpokladem úspěšného užití CBM při implementaci informačního systému je nalezení vhodného způsobu rozvrhování stránky do boxů. Hledání rozkladu stránky a jeho zdůvodnění může být podepřeno i myšlenkami Sémantického webu (Berners-Lee, Hendler, Lassila; 2001).

V našich úvahách se omezme na jednu stránku informačního systému, vzájemné provázání stránek pomocí hypertextových odkazů není v tomto okamžiku podstatné. Uživatelské rozhraní webového informačního systému je pak tvořeno pravoúhlou stránkou s informacemi, kterou lze dále rozložit na sadu významově podobných pravoúhlých oblastí (např. záhlaví, zápatí, navigační menu, odstavce textu, části formulářů), ty mohou obsahovat další vnořené oblasti. Rozklad končí, nalezneme-li nejmenší jednotku, která se již dále nedělí. Příklad jednoduchého dělení stránky je znázorněn na obrázku č. 1.



1: Jednoduché rozdělení stránky

Nyní definujme rozkladové prvky:

*Box* je pravoúhlá oblast stránky, skládající se ze dvou částí – obsahu a designu.

$$B(O, D)$$

*Obsah boxu* je představován dalším boxem nebo vizuálním prvkem.

$$O(B \parallel VP)$$

*Vizuální prvek* – již dále nedělitelný obsah boxu.

*Designem boxu* rozumíme atributy nesoucí informace o způsobu vykreslení boxu – barvy, typy písma, typy grafických prvků, ale i rozměry a umístění boxu. Design představuje způsob prezentace obsahu. V podrobnějším dělení lze i design popsat dvojicí hodnot – grafické prvky, popisné prvky.

$$D(G, P)$$

Tyto definice popisují vytvoření stromu jednotlivých komponent stránky. Dále bude nutné určit *mechanismus závislosti* – tedy definovat pravidla, podle kterých bude probíhat rozklad stránky do komponent a skládání jednotlivých komponent při sestavování

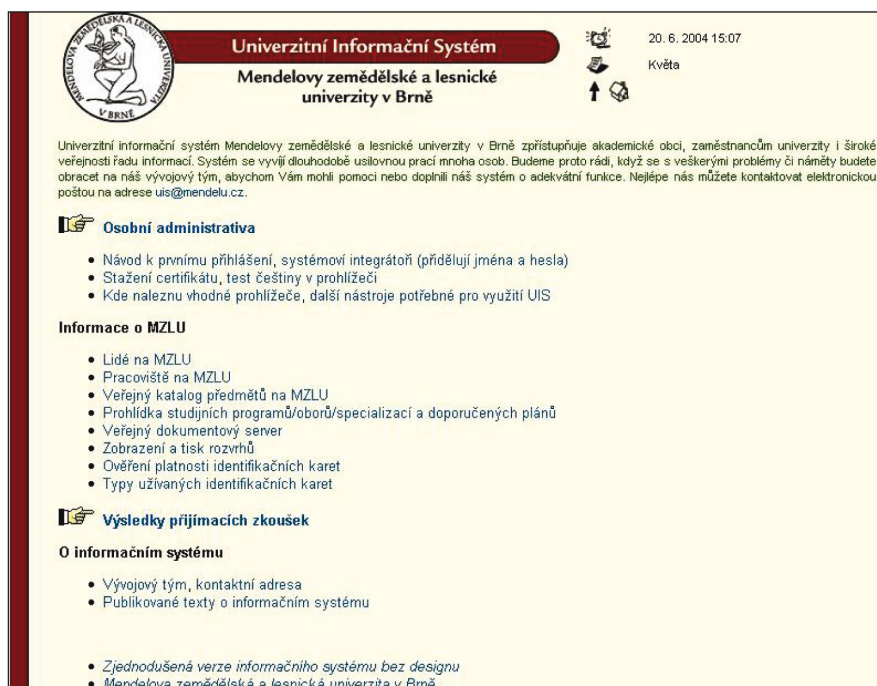
odpovědi. V rovině definice typů komponent však tento mechanismus zatím není podstatný.

### Nalezení komponent stránky

Při hledání jednotlivých komponent stránky si lze položit následující otázky, jejichž zodpovězení pomůže najít hledané objekty a závislosti mezi nimi:

- Jak rozdělit stránku do komponent?
- Objevují se na stránce typické útvary?
- Existují typické podstromy stromů skládaných z komponent?
- Je možné zobecnit rozdělení každé stránky?

Zkusme se podívat na několik různých stránek informačního systému. Vzhledem k tomu, že typy prezentovaných dat se liší podle tematického zaměření systému, začneme u nám blízkého prostředí informačního systému univerzity. Na obrázku 2 a 3 jsou ukázky stránek Univerzitního informačního systému Mendelovy zemědělské a lesnické univerzity v Brně.



**Univerzitní Informační Systém**  
Mendelovy zemědělské a lesnické univerzity v Brně

20. 6. 2004 15:07  
Květa

Univerzitní informační systém Mendelovy zemědělské a lesnické univerzity v Brně zpřístupňuje akademické obci, zaměstnancům univerzity i široké veřejnosti řadu informací. Systém se vyvíjí dlouhodobě uslovnou prací mnoha osob. Budeme proto rádi, když se s veškerými problémy či náměty budete obracet na náš vývojový tým, abychom Vám mohli pomoci nebo doplnili náš systém o adekvátní funkce. Nejlépe nás můžete kontaktovat elektronickou poštou na adrese [uis@mendelu.cz](mailto:uis@mendelu.cz).

**Osobní administrativa**

- Návod k prvnímu přihlášení, systémoví integrátoři (přidělují jména a hesla)
- Stažení certifikátu, test češtiny v prohlížeči
- Kde naleznu vhodné prohlížeče, další nástroje potřebné pro využití UIS

**Informace o MZLU**

- Lidé na MZLU
- Pracoviště na MZLU
- Veřejný katalog předmětů na MZLU
- Prohlídka studijních programů/oborů/speciálizací a doporučených plánů
- Veřejný dokumentový server
- Zobrazení a tisk rozvrhů
- Ověření platnosti identifikačních karet
- Typy užívaných identifikačních karet

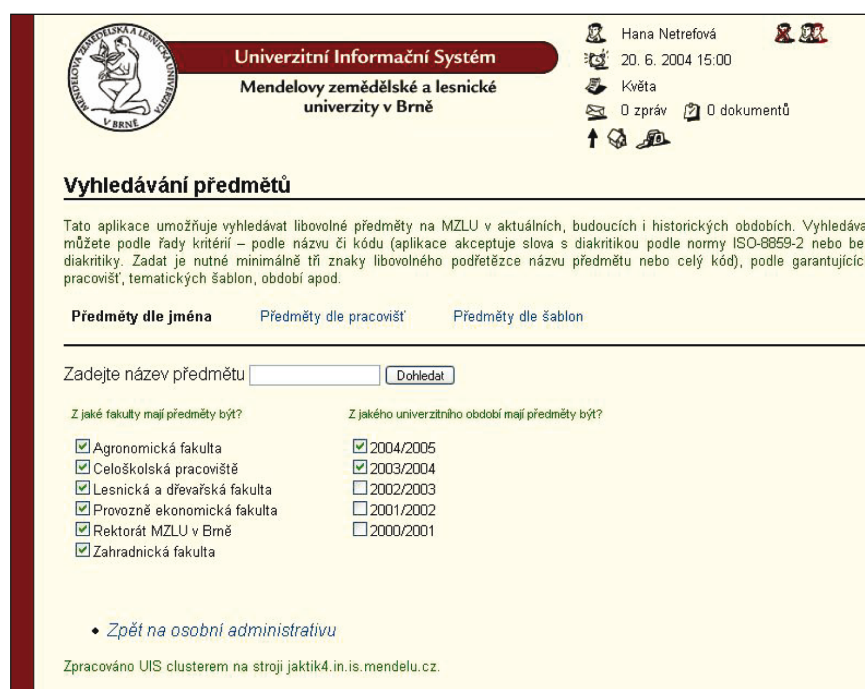
**Výsledky přijímacích zkoušek**

**O informačním systému**

- Vývojový tým, kontaktní adresa
- Publikované texty o informačním systému

- Zjednodušená verze informačního systému bez designu
- Mendelova zemědělská a lesnická univerzita v Brně

2: Příklad stránky informačního systému – úvodní strana



**Univerzitní Informační Systém**  
Mendelovy zemědělské a lesnické univerzity v Brně

Hana Netřefová  
20. 6. 2004 15:00  
Květa  
0 zpráv 0 dokumentů

**Vyhledávání předmětů**

Tato aplikace umožňuje vyhledávat libovolné předměty na MZLU v aktuálních, budoucích i historických obdobích. Vyhledávat můžete podle řady kritérií – podle názvu či kódu (aplikace akceptuje slova s diakritikou podle normy ISO-8859-2 nebo bez diakritiky). Zadat je nutné minimálně tři znaky libovolného podřetězce názvu předmětu nebo celý kód, podle garantujících pracovišť, tematických šablon, období apod.

**Předměty dle jména**   **Předměty dle pracovišť**   **Předměty dle šablon**

Zadejte název předmětu

Z jaké fakulty mají předměty být?   Z jakého univerzitního období mají předměty být?

|                                                                  |                                               |
|------------------------------------------------------------------|-----------------------------------------------|
| <input checked="" type="checkbox"/> Agronomická fakulta          | <input checked="" type="checkbox"/> 2004/2005 |
| <input checked="" type="checkbox"/> Celoskolská pracoviště       | <input checked="" type="checkbox"/> 2003/2004 |
| <input checked="" type="checkbox"/> Lesnická a dřevařská fakulta | <input type="checkbox"/> 2002/2003            |
| <input checked="" type="checkbox"/> Provozně ekonomická fakulta  | <input type="checkbox"/> 2001/2002            |
| <input checked="" type="checkbox"/> Rektorát MZLU v Brně         | <input type="checkbox"/> 2000/2001            |
| <input checked="" type="checkbox"/> Zahradnická fakulta          |                                               |

- [Zpět na osobní administrativu](#)

Zpracováno UIS clusterem na stroji jaktik4.in.is.mendelu.cz.

3: Příklad stránky informačního systému – vyhledávání předmětů v katalogu

Za prvek nejvyšší úrovně, tedy kořen stromu, můžeme považovat celou *stránku*. Na stránce je jednoznačně určené záhlaví, zápatí a tělo stránky.

*Záhlaví* je tvořeno logem Univerzitního informačního systému a skupinou ikon s texty. Ta se dále dělí na informační a navigační prvky, které jsou zobrazo-

vány podle kontextu, v němž se nacházíme. Nakonec můžeme každý prvek rozložit ještě na ikonu a text. Podle předchozích definic představují ikony a texty příslušné vizuální prvky, všechny ostatní nadřazené komponenty jsou představiteli boxů.

*Zápatí*, na uvedených příkladech nejjednodušší, se



dále skládá z jednoho nebo více zpětných odkazů a *případně* dodatečné informace. Jednotlivé objekty současně reprezentují vizuální prvky zápatí.

*Tělo stránky* je komponentou nejsložitější. Obsahuje v podstatě všechny významové informace zobrazené na stránce. Podle povahy informačního systému nebo podle zobrazovaných informací lze tělo stránky dělit dále do menších významových (např. menu, oblast pro zadávání vstupních *dat*, zobrazování výsledků) i vizuálních celků (formulář, tabulka, odstavec). V *libovolné* úrovni hierarchického stromu komponent se pak dostáváme k vizuálním prvkům (nadpis, nápočedný text, položka portálového menu, položka formuláře, odesílací tlačítko<sup>3</sup> ve formuláři apod.).

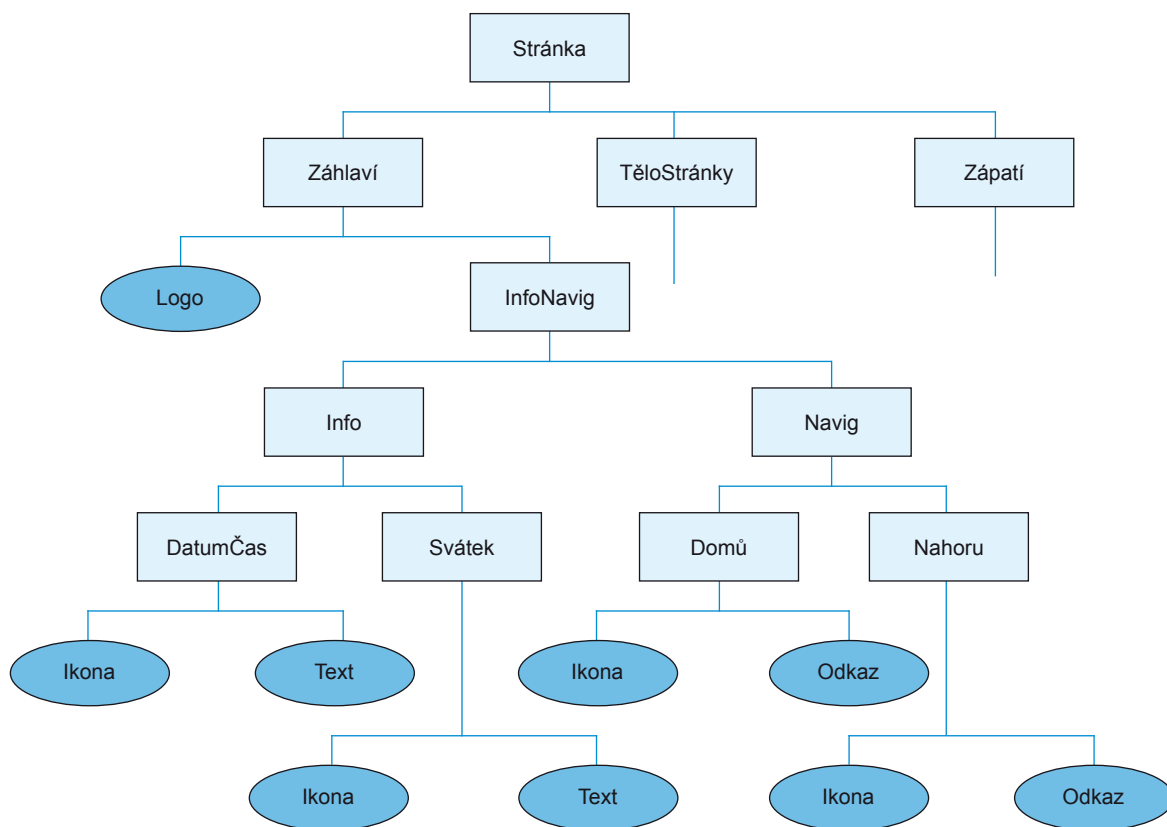
Při komponentovém rozkladu stránky tedy vidíme, že kombinujeme dva přístupy: logický a vizuální. Z logického pohledu dělíme stránku na celky, které jsou odvozené od typu informačního systému (stránky) a prezentovaných informací. Z pohledu vizuálního je stránka tvořena prvky určujícími, jak se bude

daná informace prezentovat. Tyto prvky mohou být odvozeny např. ze značek jazyka HTML.

Z uvedeného příkladu je zřejmé, že se komponenty mohou dělit na povinné, volitelné a vázané. Mezi *povinné* komponenty zařazujeme stránku, záhlaví, zápatí a tělo stránky. Ostatní komponenty jsou *volitelné*, jejich případný povinný výskyt však může být vynucen povahou předkládané informace na stránce. *Vázané* komponenty jsou takové, které jsou sice volitelné, ale nemohou stát samostatně (např. textové pole pro vyhledávání a tlačítko „Dohledat“ by jeden bez druhého neměly smysl).

### Popis rozkladu stránky

Při rozkladu stránky postupujeme tak, že procházíme jednotlivé prvky stromu od kořene až k listům, přičemž pro potřeby rozkladu si všímáme pouze složky obsahové. Design komponenty se uplatňuje při operaci opačné, tedy při skládání stránky do výsledného výstupu, proto jej ponechme stranou. Rozklad do stromu komponent je patrný na obrázku č. 4.



4: Rozklad stránky do komponent

<sup>3</sup> často se používá výraz submit button z angličtiny

Kořenovým objektem je objekt *Stránka*, který musí být typu *box*. Jeho tři povinné složky jsou *Záhlaví*, *Zápatí* a *TěloStránky*.

*Záhlaví* je tvořeno položkami *Logo* a *InfoNavig*. *Logo* představuje již významový prvek. Druhým potomkem uzlu *Záhlaví* je prvek *InfoNavig*. Ten se dělí na položky *Info* a *Navig*. Položka *Info* má další dvě složky, kterými jsou *DatumČas* a *Svátek*. A konečně prvek *DatumČas* obsahuje dva vizuální prvky *Ikona* a *Text*, které představují listy stromu v této větvi. Pro nalezení dalších listů procházíme celý strom podle pravidla o uspořádaných n-árních stromech, které zaručuje nalezení všech listů stromu.

Takto vypadá popis rozkladu velmi jednoduchého záhlaví, které odpovídá záhlaví na obrázku č. 2. Co se ovšem stane, bude-li záhlaví obsahovat nějaké další, případně jiné prvky? Srovnáme s obrázkem č. 3, kde je komponenta *InfoNavig* v záhlaví obohacena o další položky. Nestane se nic zvláštního, komponenta *InfoNavig* bude obsahovat další vnořené prvky, např. *JménoUživatele* a ten bude opět zahrnovat dva vizuální prvky, *Ikona* a *Text*. Můžeme proto konstatovat, že strom je dynamický a umožňuje přidávání a odeírání volitelných komponent. V našem modelovém příkladu ponechme stranou možnost, že prostor na stránce určený pro záhlaví může být příliš malý a všechny požadované prvky se do něj „nevejdu“.

Pravidlo rozkladu je aplikováno na každý box. Proto lze říci, že strom zajistí, že všechny listy jsou tvořeny vizuálními komponentami.

### Vázané prvky

Jak jsme se zmínili dříve, některé objekty na stránce spolu mohou souviset a zobrazení jednoho objektu bez druhého postrádá smysl. Dobře viditelné je to na příkladu webového formuláře, kdy jednotlivé položky formuláře musejí být logicky zakončeny odesílacím tlačítkem, nebo naopak toto tlačítko postrádá smysl, když nemá co odesílat.

Vázané prvky můžeme rozdělit na obousměrně a jednosměrně vázané:

- *Obousměrně vázané* jsou takové prvky, které nemohou stát jeden bez druhého, a to v obou směrech. Příkladem je výše uvedený případ s webovým formulářem.
- *Jednosměrně vázané* prvky jsou pak takové prvky, kde jeden prvek potřebuje ke své existenci prvek druhý, ale opačně to neplatí. Například nadpis kapitoly a odstavec textu. Nadpis kapitoly nemůže (nebo by alespoň neměl) stát samostatně, text bez nadpisu je však přípustný.

Vázanost jednotlivých prvků lze definovat jako pevnou a volnou:

- *Pevná vázanost* je taková, kdy existuje vazba mezi dvěma přesně danými prvky. Např. ikona v záhlaví představující datum a čas je pevně vázána s textem popisujícím aktuální datum a čas (i když obráceně to platit nemusí).

$$P \Leftrightarrow V$$

- *Volná vázanost* představuje častější případ, kdy se obecně libovolný prvek z jedné skupiny váže k libovolnému prvku ze skupiny jiné. Zůstaneme-li u příkladu s formulářem, potvrzovacím tlačítkem může být doplněna libovolná formulářová položka.  
( $P1 \parallel P2 \parallel P3 \dots Pn \Leftrightarrow (V1 \parallel V2 \parallel V3 \dots Vn)$ ).

### DISKUSE A ZÁVĚR

Po definování rozkladu stránky do jednotlivých komponent můžeme přistoupit k dalšímu kroku směřujícímu ke komponentově generovanému uživatelskému rozhraní – k definování pravidel rozkladu. Jako vodičko pro stanovení pravidel dělení stránky si můžeme položit následující otázky:

- Je možné nalézt pravidla, podle kterých se stránka dělí?
- Je možné tato pravidla zobecnit?
- Existují pravidla pro rozklad komponent do opakujících se podstromů?
- Záleží při aplikování pravidel rozkladu na jejich pořadí?
- Jak souvisí pravidla rozkladu s pravidly skládání?

Zodpovězením těchto otázek a řady dalších, které se vynoří při vlastním hledání, nalezneme pravidla rozkladu.

Pak budeme schopni definovat ucelenou metodiku pro komponentový rozklad stránky (uživatelského rozhraní) z pohledu definice komponent, stromu komponent a pravidel pro rozklad aplikovaných nad těmito prvky.

V textu jsme definovali box jako prvek, skládající se ze dvou částí: obsahu a designu. Doposud jsme se zabývali pouze stránkou obsahovou. Jedním z dalších kroků tedy bude stanovení designových prvků, jejich úlohy a významu. Hrají designové prvky roli již při rozkladu stránky? Nebo se uplatňují až ve fázi skládání odpovědi klientovi? Odpověď jistě není jednoznačná, musíme tedy i pro designovou složku boxů nalézt způsob a pravidla začlenění do rozkladu (a skládání) stránky a do celého component boxing modelu.

Na úrovni prezentační vrstvy component boxing modelu pak budeme schopni realizovat řadu služeb, mezi něž patří řízení struktury portálu, správa zdrojů, správa šablon a obsahu boxů, navigace v systému, vícejazyčnost a personalizace. Spojením s ostatními vrstvami component boxing modelu vznikne ucelená metodika pro návrh, vývoj a implementaci komponentově generovaných webových informačních sys-

témů, které ve srovnání s klasickými informačními systémy budou snadno rozšiřovatelné a pro uživatele více intuitivní. Tím bude naplněno jedno z poslání informačních systémů dnešní doby.

#### SOUHRN

Informační systémy pronikají do stále více oblastí lidské činnosti, proto je třeba klást stále větší důraz na jejich kvalitu a uživatelskou přívětivost. S tím souvisí hledání nových cest, jak informační systémy lépe navrhovat, vyvíjet a implementovat. Autorka článku se spolu s kolegy již delší dobu zabývá tematikou komponentově orientovaných webových informačních systémů s cílem definovat metodiku pro tvorbu tohoto typu informačních systémů. Doposud byl navržen tzv. Component boxing model, který říká, jak obecně skládat informační systém z komponent na úrovni jednotlivých vrstev třívrstvé architektury systému. Na úrovni jednotlivých vrstev je dále třeba provést detailní analýzu prvků a závislostí mezi nimi. Tento článek se zabývá uživatelským rozhraním informačního systému. Podrobně rozebírá problematiku rozkladu stránky do jednotlivých komponent a hledá typy komponent včetně vazeb mezi nimi. Definuje pojmy jako box, obsah boxu, design boxu, vizuální prvek, jednosměrně a obousměrně vázané prvky, vázanost prvků. V závěru článku je nastíněn směr dalšího zkoumání problematiky, který by měl vyústit do stanovení metodiky vývoje informačního systému na úrovni uživatelského rozhraní a prezentační vrstvy, a dále začlenění tohoto dílčího problému do vývoje celého systému.

webový informační systém, uživatelské rozhraní, komponenty, rozklad stránky

#### LITERATURA

- BERNERS-LEE, T., HENDLER, J., LASSILA, O.: The Semantic Web. *SCIENTIFIC AMERICAN.COM*, 5/2001. <http://www.scientificamerican.com> [cit. 20. 6. 2004].
- LYNCH, H. J., HORTON, S.: *Web Style Guide: Basic Design Principles for Creating Web Sites*. <http://www.webstyleguide.com> [cit. 20. 6. 2004].
- NETREFOVÁ, H.: *Nové postupy uplatňované při návrhu uživatelsky přívětivých informačních systémů*. Rozpracovaná disertační práce. Brno: MZLU, 2003. 74 s.
- NETREFOVÁ, H., ŠORM, M.: User-friendly Web Information Systems. In *Proceedings of the 9th International Conference of European University Information Systems*. Amsterdam: UvA, 2003. s. 518–522. ISBN 90-9017079-0.
- PAZDZIORA, J.: *Converting Web Applications to Data Components: Design Methodology*. <http://www.fi.muni.cz/~adelton/papers/compsac2001.ps> [cit. 9. 5. 2003].
- SATRAPA, P.: *Web Design*. Liberec: Neokortex, 1997. 414 s. ISBN 80-902230-1-X.
- ŠORM, M.: *Využití principu skládání komponent při návrhu webového informačního systému*. Rozpracovaná disertační práce. Brno: MZLU, 2004. 73 s.
- ŠORM, M., NETREFOVÁ, H.: The Theory of Web Information Systems. In *Proceedings of the 9th International Conference of European University Information Systems*. Amsterdam: UvA, 2003. s. 319–322. ISBN 90-9017079-0.
- ŠORM, M., NETREFOVÁ, H., MOTYČKA, A.: Component boxing model as technology for building portals. In *Proceedings of the 10th International Conference of European University Information Systems*. Ljubljana, 2004. In press.
- VOŘÍŠEK, J.: *Informační technologie a systémová integrace*. Praha: VŠE, 1996. 198 s. ISBN 80-7079-895-5.

#### Adresa

Ing. Hana Netrefová, Ústav informatiky, Mendelova zemědělská a lesnická univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika, e-mail [hanac@pef.mendelu.cz](mailto:hanac@pef.mendelu.cz)