

Článek pro konferenci OSR4 2009

Bc. Petr Mrůzek

Statková 5
Opava, 746 01
tel. +420 774325565

email: wors@email.cz

ABSTRAKT

Článek popisuje software, který je vyvíjen za účelem vytvoření inventáře objektů v nějaké organizaci. Inventář se zde bere jako ucelený souhrn všech informací, které je potřeba o objektech uchovávat. Vyvíjený software se snaží být co možná nejobecnější, aby nezáleželo na tom, o jaké objekty se bude jednat a v jaké organizaci bude hotový produkt nasazen. Primárním zaměřením je však odvětví informačních technologií.

Kategorie a předměty popisu

D.3.3 [PHP]: Webová aplikace s formuláři pro obsluhu databáze – *SQL příkazy, PHP třídy a metody, XHTML formuláře, tvorba inventáře.*

Všeobecné pojmy

Dokumentace, Návrh, Spolehlivost, Bezpečnost.

Klíčová slova

Inventarizace, SRSW4IT, opensource.

1. ÚVOD

Software byl inspirován potřebami Slezské univerzity v Opavě (SU). Ukázalo se, že je nutné začít uchovávat detailnější informace o vlastnictví a provozu zařízení používaných na různých fakultách. Proto vznikl podnět pro vypracování projektu v rámci bakalářské diplomové práce.

Cílem bylo vytvořit aplikaci s otevřeným zdrojovým kódem, která by splnila požadavky SU a kterou by zároveň bylo možné později rozšířit o další funkce. Dalšími požadavky byly nezávislost na platformě a databázovém systému a také dostatečně obecný datový model, aby se docílilo jakési univerzálnosti použití.

Jak jsme si s tím poradili se dočtete níže.

2. PROGRAMOVACÍ JAZYK

Software je psán ve skriptovacím jazyce PHP v objektové verzi hlavně z důvodu vysoké použitelnosti i na freehostingových serverech. Dalším důvodem je požadovaná nezávislost na platformě. Komunikaci s databází zajišťuje třída PHP Data Object (PHP-PDO), která je od verze 5.2 klasickou součástí

jazyka. Díky této třídě je možné používat shodné funkce pro připojení k různým databázovým systémům.

3. DATOVÝ MODEL

Datový model je základem celého softwaru a je-li navržen nevhodně, přináší to nemalé problémy s vývojem. Proto jsme této části práce věnovali velké množství času a úsilí.

Model byl v průběhu práce měněn a doplňován vždy, když bylo nutné vytvořit nové tabulky či sloupce. V současnosti je tvořen 18-ti relacemi, které jsou vzájemně propojeny pomocí cizích klíčů. Většina z těchto relací slouží hlavně pro ukládání vnitřních informací aplikace, případně pro automatické zaznamenávání událostí.

3.1 Pomocné objekty

Přestože hlavním účelem aplikace je ukládání informací o používaných zařízeních, je potřeba vložit tzv. pomocné objekty, které pomáhají při jejich identifikaci nebo lokalizaci. Mezi tyto objekty patří například *Organizační jednotka*, *Budova* a *Místnost*.

Organizační jednotkou je myšlen jakýkoli útvar v organizaci. Na Slezské univerzitě může jít o fakulty, ústavy, katedry apod. Hierarchie mezi nimi je znázorněna podobně jako v LDAP. Jejich název je tvořen zkratkou jednotky, po něm následuje tečka a název nadřazené jednotky. Např. *ui.fpf.su* označuje ústav informatiky na FPF SLU a zároveň patří do *fpf.su* a *su*.

Dalším typem pomocných objektů jsou *Typy událostí* a *Události*. Díky těmto objektům je možné nadefinovat, při jakých událostech má aplikace uložit záznam s informacemi. Například v okamžiku přihlášení se do tabulky *Události* vloží záznam s typem „User login“ a několika upřesňujícími daty.

3.2 Uživatelé, skupiny, oprávnění

Všichni uživatelé jsou ukládáni v systému na jednom místě a tím je tabulka *Uživatelé*. Každý z nich může náležet do jedné z *Uživatelských skupin*. Již po instalaci jsou vytvořeny 2 skupiny: *guest* a *admin*, přičemž do skupiny *guest* jsou automaticky zařazováni nepřihlášení uživatelé.

K jakékoli skupině může být přiřazeno oprávnění pro čtení či úpravu již existujícího zařízení. Můžeme tedy zařadit, aby například „PC pro výuku“ bylo možné zobrazit i nepřihlášenému uživateli a to tak, že skupině *guest* přidáme oprávnění pro čtení informací o tomto počítači.

Výjimkou jsou skupiny, které jsou uvedeny jako *Vedoucí* některé z organizačních jednotek (OU). Ty mají vždy administrátorská práva ke všem zařízením pod touto OU.

3.3 Zařízení

Zařízením se rozumí základní datová jednotka, kterou budeme ukládat. Stejně tak se může jednat o část již existujícího zařízení. Veškeré obecné parametry se ukládají v tabulce *Zařízení*. Jedná se

zejména o propojení na jiné tabulky (*Název, Místnost, Typ zařízení, Organizační jednotka, Rodičovské zařízení, Vlastník*, atd.). Dále zde systém automaticky ukládá datum vytvoření i datum poslední úpravy včetně uživatele, který změnu provedl.

Jediné tři nepovinné atributy začínají písmeny *Img*. Jsou zde nachystány pro případ vytvoření přídatného modulu, pomocí něhož se zařízení barevně rozliší a rozmístí na 2D či 3D mapě.

3.4 Atributy, typy a skupiny zařízení

Všechny obecné informace máme tedy uloženy, teď se zaměříme na vlastnosti, kterými se zařízení liší, případně které mají společné.

Vzhledem k tomu, že jsme chtěli mít aplikaci co nejobecnější, není pevně definován žádný atribut nebo šablona vlastností. V systému lze totiž vytvořit *Typ atributu*, který má popsat jednu určitou vlastnost zařízení. Tyto atributy je pak možné spojovat do *Typu zařízení*. Uživatel si tedy může vytvořit šablonu přesně podle svých představ. Další výhodou je absence prázdných řádků, které se v tabulkách často objevují, mají-li všechna zařízení jednotnou šablonou.

Vlastnosti sloučené do typu zařízení jsou ukládány do zvláštní tabulky *Data atributů*. Každý atribut navíc obsahuje informaci, je-li možno spojit jej s jiným zařízením (tzv. *Typ propojení*). Díky tomu jsme mohli vytvořit unární relaci a jednotlivé řádky tabulky mezi sebou propojovat. Tak můžeme například propojit switch se všemi k němu připojenými stanicemi.

Zařízení je možno rozdělovat do skupin podle různých parametrů. Např. zařízení *Dataprojektor* může patřit do skupiny *Vizuální technika* a zároveň do skupiny *Zařízení ústavu informatiky*. Každá skupina má pouze název a textový popis.

4. ETAPY VÝVOJE

Po analýze a vytvoření datového modelu jsme přistoupili k programování jednotlivých souborů tříd a jejich metod. Tyto soubory obsahují skripty, které se používají při komunikaci s databází a pro manipulaci s daty objektů. V případě pozdější tvorby přídatného modulu lze samozřejmě vytvořit nové třídy a metody.

Ve chvíli, kdy byly všechny metody otestovány, jsme začali pracovat na tvorbě uživatelského rozhraní v podobě webové aplikace. Ta je tvořena jedním PHP souborem (*index.php*), do něž se podle situace vkládají data získaná obslužnými skripty, ať už jde o přihlašovací skript, nebo skript zpracovávající vyplněný webový formulář.

V průběhu vývoje jsme samozřejmě občas změnili názor na vhodné řešení problému a podle toho jsme byli nuceni upravit také aplikaci. Například v začátku projektu jsme komunikovali s databází přes ovladače ODBC, ale z důvodu rozdílných ovladačů na různých platformách jsme byli nuceni přejít na již zmiňované PHP-PDO. Také jsme změnili způsob přihlašování uživatelů. Klasické přihlášení přes webový formulář jsme zaměnili za HTTP autentizaci. Důvodem byl možný import kontaktů ze serveru LDAP, aby uživatel, již přihlášený k jiné aplikaci SU nemusel své údaje zadávat znovu.

5. FUNKCE APLIKACE

5.1 Co již software umí

V tuto chvíli je možné vkládat jak pomocné objekty, tak zařízení, je ale nutné postupovat postupně. Ukládání i úprava je řešena webovými formuláři, pro každou tabulku existuje jeden.

Na stránkách vývoje byl zveřejněn doporučený postup vytváření struktury inventáře¹.

Jak již bylo řečeno, můžeme vytvořit hierarchii organizačních jednotek. Systém vždy testuje, jestli nadřazená jednotka (uvedená za tečkou) existuje.

Uživatelé mohou být děleni do skupin a těm lze přiřazovat jedno ze tří oprávnění k zařízení. Jsou to tyto:

- "r" - čtení
- "w" - čtení a úprava neklíčových atributů
- "a" - úplné oprávnění administrátora

Přihlašování uživatelů je řešeno pomocí HTTP autentizace. Zároveň je vyřešeno i odhlášení bez nutnosti zavírat prohlížeč.

Vyhledáváním můžeme zobrazit zařízení, které odpovídají zadaným parametrům. Chceme-li zobrazit všechna zařízení, které můžeme číst, zadáme do vyhledávače znak '%'. U každého výsledku je zobrazeno několik upřesňujících informací a odkaz pro zobrazení veškerých uložených hodnot (Více). Zatímco vyhledávání zobrazí pouze zařízení, pomocné objekty můžeme najít pomocí rolovacích nabídek.

5.2 Na čem se pracuje

Co se týče modifikace objektů, řešíme právě jeden z posledních problémů. Jedná se o možnost úpravy *Typu zařízení* a k němu přiřazených atributů.

Další v pořadí jsou skripty pro příkazový řádek (PHP-CLI), které by měly být používány k hromadnému naplnění systému daty. Administrátor si pak bude moci připravit např. XML soubor podle DTD šablony a pomocí CLI skriptu data vloží. Skripty by měly používat metody, které jsou již napsány, jen bez grafického rozhraní. K tomu se váže i potřeba importu a exportu dat. Tyto dvě řešení se nejspíš budou moci spojit v jedno. Předpokládaný formát vstupních i výstupních souborů je standard XML.

5.3 Vize do budoucna

Výhledově bychom rádi propojili aplikaci SRSW4IT se systémem pro obnovu disků PC stanic, jehož vývoj také probíhá pod záštitou Slezské univerzity. Systém obnovy by mohl čerpat data z naší aplikace a také do ní ukládat.

Možným rozšířením by mohl také být modul pro grafické znázornění rozmístění jednotlivých zařízení. Dále jsme uvažovali o exportu DNS.

Na stránkách vývoje je možné napsat nápad na funkci či vylepšení softwaru. Vaše nápady vřele uvítáme a budeme-li toho schopni, dopracujeme je do aplikace.

¹ <http://srsw4it.bp.opf.slu.cz/wiki/structure>