

MUDRLite – elektronický zdravotní záznam šitý na míru

Josef ŠPIDLEN, Petr HANZLÍČEK, Jana ZVÁROVÁ
*EuroMISE centrum – Kardio, Ústav informatiky AV ČR
Pod Vodárenskou věží 2, 182 07 Praha 8, Česká republika*

Abstrakt. V dnešní době řada nemocnic využívá elektronické formy zdravotního záznamu integrovaného jako část nemocničního informačního systému. Nicméně tyto systémy jsou často vhodnější pro správu nemocnice než pro klinickou praxi. Zdravotní záznam není dostatečně strukturován, obsahuje mnoho volného textu a množina sbíraných a strukturovaných atributů je pevná, prakticky bez možnosti rozšíření. Lékaři sbírající údaje např. pro potřeby medicínských studií využívají k tomuto často obecné kancelářské nástroje jako například databáze MS Access či tabulky MS Excel.

EuroMISE centrum – Kardio pracuje na vývoji aplikace elektronického zdravotního záznamu (Electronic Health Record, EHR) nazývaného MUDRLite, který má potenciál vyplnit vzniklou mezeru mezi existujícími EHR aplikacemi. MUDRLite je vytvářen v rámci aplikovaného výzkumu na poli návrhu EHR, který je založen na zkušenostech získaných během spolupráce na evropském projektu TripleC. Vývoj MUDRLite je speciální větví vývoje aplikace MUDR (MULTimedia Distributed Record), který zjednodušuje jak architekturu aplikace tak princip ukládání dat.

MUDRLite sám o sobě je prázdným tělem, které musí být naplněno XML konfiguračním souborem. Tento XML soubor plně popisuje jak vzhled tak chování výsledné EHR aplikace. XML soubor obsahuje konstrukce v jazyce „MUDRLite Language“ (MLL) připomínající na první pohled principy 4GL jazyků. Toto umožňuje – využitím principů událostmi řízeného programování – definovat různé chování vyvolané různými akcemi, např. kliknutí na tlačítko může vyvolat naplnění formuláře výsledky určitého SQL dotazu. Díky tomu může být EHR MUDRLite ušit na míru prakticky každému poskytovateli zdravotní péče. Toto umožňuje jednoduché využití MUDRLite ve specifickém prostředí. Nasazení MUDRLite v první fázi testujeme na oddělení neurovaskulární chirurgie Ústřední vojenské nemocnice v Praze.

1. Úvod

Evropské centrum pro medicínskou informatiku, statistiku a epidemiologii – Kardio (EuroMISE centrum – Kardio) se zaměřuje na nové přístupy k návrhu elektronického zdravotního záznamu (Electronic Health Record, EHR) zahrnujícímu elektronická lékařská doporučení a inteligentní systémy pro podporu rozhodování a dolování dat z databází [1]. Spolupráce na projektu I4C-TripleC [2, 3, 4] čtvrtého rámcového programu Evropské komise, standardy CEN TC 251 a spolupráce s lékaři poskytly značné zkušenosti, které vyústily v seznam 15 požadavků na EHR systém [5].

Jako realizaci systému, který by splňoval tyto požadavky, EuroMISE centrum pracuje na vývoji EHR aplikace nazývané MUDR (MULTimedia Distributed Record) [6, 7, 8, 9, 10]. Dle stanovených požadavků byla definována modulární struktura systému. Tato je založena na klasické třívrstvé architektuře využívající datovou vrstvu, aplikační (funkční) vrstvu a uživatelské rozhraní. Toto dělení umožňuje oddělení fyzického úložiště dat, aplikační inteligence a klientské části.

Množina sbíraných atributů se liší v různých odděleních, organizacích a zároveň během času. MUDR využívá dynamicky rozšiřitelnou a modifikovatelnou strukturu položek založenou na principu tzv. znalostní báze a datových složek. Tento přístup umožňuje reorganizaci rozšíření hierarchie sbíraných atributů bez změny databázového schématu, což dělá systém absolutně flexibilním, nicméně přináší též komplikace. Oříškem se stává vytvoření univerzálního uživatelského rozhraní, které by bylo zároveň dostatečně přátelské a pohodlné na ovládání. Nasazení EHR MUDR do specifického prostředí vyžaduje poměrně značné úsilí, znalostní báze konkrétní oblasti musí být namodelována a všechny MUDR komponenty musí být korektně nainstalovány a nakonfigurovány.

V současné době většina nemocnic využívá různých forem elektronického zdravotního záznamu integrovaného do nemocničního nebo klinického informačního systému. Bohužel tyto systémy jsou často vhodnější k řízení nemocnice než k poskytování léčebné péče. Takovýto zdravotní záznam nebývá dostatečně strukturován, obsahuje množství volného textu a množina sbíraných atributů bývá pevná, prakticky bez možnosti uživatelského rozšíření. Lékaři sbírající informace za účelem medicínských studií často využívají různých pomocných nástrojů jako například databáze MS Access či tabulky MS Excel k ukládání medicínských dat. Použití EHR MUDR v takovýchto případech je sice možné, ale může být až příliš složité a zbytečné. Navíc, výsledek nemusí být tak uživatelsky příjemný jako speciální aplikace vyvinutá na míru.

2. MUDRLite

Využití EHR MUDRLite je pro tento účel daleko jednodušším řešením. Aplikace MUDRLite je stejně jako MUDR vyvíjena v rámci aplikovaného výzkumu návrhu EHR. MUDRLite tvoří samostatnou větev vývoje MUDR, která zjednodušuje jak architekturu aplikace záznamu, tak princip ukládání dat.

2.1. Architektura MUDRLite

Architektura MUDRLite je založena na dvouvrstvé architektuře. První vrstvou je relační databázový stroj, momentálně podporované jsou MS SQL Server verze 7 a 2000. Druhou vrstvou je již přímo uživatelské prostředí na platformě Windows.

Databázové schéma odpovídá konkrétním požadavkům dané instance a tedy na rozdíl od pevného databázového schématu aplikace MUDR je rozdílné v různých prostředích. Univerzalita aplikace MUDRLite je založena na zcela jiném přístupu. Databázové schéma MUDRLite může být navrženo s použitím standardních modelovacích technik jako např. E-R Modelování [11]. Uživatelské rozhraní MUDRLite si dokáže poradit s rozličnými databázovými schématy. Tato vlastnost zjednodušuje způsob importování starých dat uložených v rozličných databázích či souborech.

2.2. Uživatelské rozhraní MUDRLite

Celkový vzhled i chování aplikace MUDRLite je kompletně popsáno konfiguračním souborem ve formátu XML. Koncovému uživateli se aplikace jeví jako množina formulářů definovaných v konfiguračním souboru takto:

```
<form name="new_hosp_form" label="New Hospitalization" author="JS"
date="27.1.2004" language="en" sizeX="420" sizeY="410"> ... </form>
```

Atributy popisují interní jméno formuláře, titulek, který je zobrazován uživateli, kým a kdy byl formulář navržen, v jakém je jazyce a vizuální velikost formuláře. Ovládací prvky

jsou na formuláře umístovány pomocí pod-elementů jako `<button>`, `<combobox>`, `<groupbox>`, `<textbox>`, `<datagrid>`, `<checkbox>` atd.

Ovládací prvek je umístěn na formulář touto syntaxí:

```
<label name="but_del_hosp" label="Delete..." posX="200" posY="315"
  sizeX="80" sizeY="23" tabIndex="2" color="indigo" />
```

Atributy `name`, `label`, `posX`, `posY`, `sizeX`, `sizeY`, `tabIndex` a `color` jsou jednotně používány ve všech elementech popisujících rozličné ovládací prvky na formulářích. Vyjadřují interní jméno prvku, titulek či popisek prezentovaný uživateli, pozici a velikost prvku, tabulátorový index a barvu prvku. K dispozici máme přes 160 různých jmen barev, které jsou přesně definovány v knihovně GDI+ .NET Framework [12]. Tyto barvy zahrnují kromě klasických barev také barvy systémové (barva okna, okenního rámu, atd.). Pouhým přeložením atributů `label` získáme velmi rychle jazykově lokalizovanou verzi aplikace MUDRLite. Existují jednoduché elementy jako například `<groupbox>` nebo `<label>`, které používají pouze výše uvedené atributy. Často využívaným atributem je atribut `readonly` určující, zda je možné příslušnou hodnotu na formuláři editovat. Ostatní elementy využívají více atributů, které dále podrobněji popíšeme.

Nejtypičtějším elementem je editovací políčko definované pomocí elementu `<textbox>`. Přidané atributy `acceptsReturn` a `acceptsTab` v tomto případě určují, zda je možné do políčka vkládat zalomení řádku či tabulátory. Pomocí atributu `multiline` návrhář formuláře určuje, zda políčko může obsahovat více řádek textu. Posuvníky přidáme k editovacímu políčku pomocí atributu `scrollbars`, který může nabývat čtyř různých hodnot: `none`, `horizontal`, `vertical` a `both`.

Dalším v databázových aplikacích hojně využívaným ovládacím prvkem je datová mřížka. V konfiguraci MUDRLite element `<datagrid>` používá atribut `colwidth` k nastavení preferované šířky sloupce. Ostatní vizuální aspekty datové mřížky se nastavují použitím stejných technik, kterými se nastavuje i obsah tohoto ovládacího prvku. Tyto budou popsány dále.

Výčtové typy bývají často spojeny s ovládacími prvky typu „combo-box“ či „list-box“. Element `<combobox>` využívá numerický `maxDropDownItems` atribut ke specifikaci maximálního počtu položek zobrazitelných najednou. Atribut `sorted` určuje, zda se položky mají automaticky abecedně třídit. Existují dva odlišné styly „combo-boxu“: listový styl, kde uživatel může pouze vybrat hodnotu z předem dané pevné množiny, a editovací styl, kde je možné do množiny přidat vlastní hodnotu. Tento styl je nastavován `dropDownStyle` atributem. Dále máme k dispozici dvě možnosti jak naplnit „combo-box“ hodnotami. První z nich je opakované využití pod-elementu `<item>`, který definuje každou položku pomocí `value` a `display` atributů. Toto znamená, že hodnoty ukládané v databázi nemusejí být shodné s těmi, které se zobrazují uživateli. Tato funkčnost pomáhá udržet třetí normální formu datového schématu (3NF) [13] a zároveň uchovat aplikaci uživatelsky přívětivou. Druhou možností je využít SQL dotazu do databáze. Takovýto dotaz je pak zpracován při nahrávání formuláře. Dotaz by měl vracet dva sloupce: datovou a zobrazovanou položku, nebo pouze jeden sloupec v případě, kdy tyto položky jsou shodné. Technicky je tento způsob realizován pod-elementem `<items>` s atributem `command`, který příslušný dotaz specifikuje.

Připravováno je několik dalších ovládacích prvků jako jsou stromové pohledy, tabulátorové záložky, obrázky, posuvníky, indikátory postupu a stavu, tipy a další, které ale zatím nejsou plně implementovány. Malý příklad uživatelsky definovaného formuláře je zobrazen na obr. 1. Tento je definován takto:

```

<form name="edit_patient_xx" label="Alena Nováková" author="JS"
  language="en" date="18.2.2004" sizeX="250" sizeY="260">
  <label label="Jméno:" name="lbl_patient_" posX="20" posY="38"
    sizeX="80" sizeY="25" color="indigo"/>
  <label label="Adresa:" name="lbl_address" posX="20" posY="68"
    sizeX="80" sizeY="25" color="indigo"/>
  <label label="Pohlaví:" name="lbl_sex___" posX="20" posY="138"
    sizeX="80" sizeY="25" color="indigo"/>
  <textbox name="t_nm" posX="100" posY="35" sizeX="130" sizeY="15"/>
  <textbox name="t_ad" posX="100" posY="65" sizeX="130" sizeY="60"
    acceptsReturn="true" multiline="true"/>
  <combobox name="c_ge" posX="100" posY="135" sizeX="130" sizeY="21"
    sorted="true" maxDropDownItems="2" dropDownStyle="list">
    <item value="m" display="muž"/><item value="w" display="žena"/>
  </combobox>
  <groupbox label="Detaily" name="g_dtt" posX="10" posY="10"
    sizeX="230" sizeY="190" color="red"/>
  <button label="Uložit" name="but_save" posX="20" posY="220"
    sizeX="80" sizeY="23" tabIndex="4" color="blue"/>
  <button label="Storno" name="but_exit" posX="150" posY="220"
    sizeX="80" sizeY="23" tabIndex="5" color="blue"/>
</form>

```

Obr. 1. Příklad jednoduchého uživatelsky definovaného formuláře.

Výše uvedená definice vytváří formulář zobrazený na obr. 1. Nicméně zde zatím zůstávají nevyřešeny dva problémy. Se stlačením tlačítek není spojena žádná akce. Dalším problémem je, že ačkoli by se formulář měl zobrazit s předvolenými detaily aktuálně zvoleného pacienta, po startu se vždy otevře prázdný. Oba tyto problémy je možné vyřešit pomocí jazyka „MUDRLite Language“ (MLL).

2.3. Jazyk MLL

Jazyk MLL je událostmi řízený jazyk sloužící k popsání vzhledu i chování MUDRLite aplikace. Jazyk obsahuje jednoduché konstrukce připomínající principy 4GL jazyků, což umožňuje jednoduchou práci s databází.

Konstrukce jazyka MLL jsou vloženy v XML konfiguračním souboru pomocí elementu `<action>`. Tento element může být vložen jako pod-element elementu formuláře

nebo libovolného ovládacího prvku. Tímto je určeno, zda se akce vztahuje k celému formuláři nebo jen k ovládacímu prvku na formuláři. Každá akce je vyvolána určitou událostí. Typickými událostmi jsou otevření či zavření formuláře, kliknutí na tlačítko apod. Události, které akci spouštějí, jsou připojeny k akcím atributem `invoke`.

Existují různé typy akcí. Obsah akce je vždy popsán v obsahu elementu `<action>`. Jednoduché akce pracují s ovládacími prvky na formuláři. Je možné např. zobrazit či schovat ovládací prvek pomocí pod-elementů `<hide>` resp. `<show>`. Cílový ovládací prvek je specifikován atributem `result`, ve kterém odkazujeme ovládací prvky pomocí jejich jmen. Typicky můžeme odkázat i na více ovládacích prvků současně tím, že jejich jména oddělíme čárkou. V případě, že potřebujeme odkázat ovládací prvek umístěný na jiném formuláři, použijeme plné jméno ovládacího prvku vytvořené tečkovou notací – spojením jména formuláře a ovládacího prvku tečkou. Rezervováno je jméno `parent` využívané k odkazu na rodičovský formulář, tj. formulář, který byl aktivní bezprostředně před otevřením aktuálního formuláře. Díky tomu není možné žádný formulář pojmenovat jménem „parent“. Plné jméno může být i o něco komplikovanější v případě, že nechceme odkazovat celý ovládací prvek nýbrž pouze jeho část. Příkladem je odkazování sloupce datové mřížky. V tomto případě je tečka použita i ke spojení jména datové mřížky se jménem jeho sloupce. Plné jméno pak vypadá např. `parent.patients_grid.patient_id`.

Další typ akcí je vázán na celé formuláře. Formulář může být uzavřen elementem akce `<exit_form>`. Nový formulář otevřeme pomocí elementu `<new_form>` s atributem `name` specifikujícím jménem, který formulář má být otevřen. Například formulář editující administrativní údaje o pacientovi zobrazíme následujícím kódem:

```
<action invoke="click">
  <new_form name="edit_patient"/>
</action>
```

Další akce používají jazyk MLL ke komunikaci s databází k získání/uložení hodnot z/do ovládacích prvků. V obsahu těchto akcí je používána rozšířená forma SQL specifikována v atributu `command`. Jména ovládacích prvků jsou v těchto příkazech ohraničena párem dvojteček. Výsledné atributy jsou specifikovány pomocí jejich jmen v atributu `result`. I zde je možné adresovat více ovládacích prvků tím, že jejich jména nebo plná jména v tečkové notaci oddělíme pomocí čárek. Akce typu „select“ může být provedena například pomocí následujícího kódu:

```
<action invoke="load">
  <select
    command="select full_name as 'Jméno', address as 'Adresa',
              sex_id as 'Pohlaví', patient_id as 'ID pacienta'
            from patient
            where patient_id = ':parent.patients_grid.patient ID:'"
    result ="e_nm, e_ad, c_g, l_id"/>
</action>
```

Tento příklad ukazuje jak naplnit formulář editující administrativní údaje o pacientovi detaily aktuálně zvoleného pacienta v datové mřížce rodičovského formuláře. Zároveň je zde demonstrován jednoduchý trik. Identifikátor pacienta je uložen v neviditelném popisku, který má zde roli interní pomocné proměnné. V případě, že uživatel klikne na tlačítko „Uložit“ je tato proměnná využita ke specifikaci pacienta, který má být aktualizován „MLL update“ akcí. Zpravidla počet hodnot vrácených SQL příkazem odpovídá počtu definovaných výstupních ovládacích prvků v atributu `result`. Výjimkou je když výstupní ovládací prvek představuje datovou mřížku. V tomto případě se mřížka automaticky přizpůsobí výstupu SQL příkazu. Názvy sloupců mřížky lze specifikovat pomocí „as“ SQL

konstrukce. Vkládání, aktualizace a mazání provádíme analogicky příkazy <insert>, <update> a <delete>.

V první verzi MUDRLite existuje ještě doplňková akce. Určitý text může být počítačem automaticky přečten pomocí akce <speak> specifikující čtený text v atributu text. Podporována je však zatím pouze anglická výslovnost.

Různé akce lze také řetězit za sebe. Obr. 2 ukazuje komplexnější příklad formuláře navrženého pro oddělení neurovaskulární chirurgie Ústřední vojenské nemocnice v Praze.

Pacienti

Jméno:

	Jméno	Rodné číslo	Adresa	ID pacienta
	Kalinova Alena	<anonymized>	Teplicka 7, Krupka	9797
	Kotoučová Alena	<anonymized>	Polská 58, P-2	9842
	Mocová Alena	<anonymized>	Kněžmost 225, MB	9952
	Mrackova Alena	<anonymized>	Tovarní 260, Dubí	9650
▶	Nováková Alena	<anonymized>	Jiráskova 4211, Libeň	10063
	Řířpová Alena	<anonymized>	Rytířova 6, P-4	10031
	Sedláčková Alena	<anonymized>	Kosmonautů 1550, Turnov	9888
	Svatková Magdale	<anonymized>	Brigadniku 12, KV	9663
	Vlčková Alena	<anonymized>	Hekrova 816, P-4	10128

Hospitalizace

	Číslo	V roce	Věk	Kód	Riziková	Out měsíc	Out rok	Out finá
	6	1996	62	2	2	2	1	1
	7	2001	67	2	3	2	2	1
▶	8	1998	64	3	2	1	2	2

Detaily:

Vstupní nález:

Rizikové faktory:

Závěr:

Aneurisma AVM Karotidy

Obr. 2. Formulář MUDRLite – Oddělení neurovaskulární chirurgie ÚVN v Praze.

3. Nasazení EHR MUDRLite

Nasazení EHR MUDR do specifického prostředí vyžaduje několik přípravných fází. Nejprve je nutné, aby příslušní lékaři specifikovali, jaké informace bude potřeba ukládat. Toto musí být provedeno přesně. Je nutné vytvořit model se všemi sbíranými atributy, jejich typy, vzájemnými vztahy, jednotkami, přesností apod. Společně s datovými inženýry je dále vytvořen tzv. „Entity-Relationship Model“ (E-R Model [11]).

Další dvě fáze následují paralelně. Jednou z nich je migrace dat ze starého používaného systému. V dnešní době se setkáváme zřídka se situací, kdy neexistují žádná historická data. Migrace je prováděna pomocí různých technologií, zpravidla s využitím nástrojů „SQL Server Data Transformation Services“.

Výsledkem druhé fáze je definice MUDRLite formulářů a chování MUDRLite aplikace zapsané v konfiguračním XML souboru v jazyce MLL. Za tímto účelem vytvoření tohoto souboru je možné využít různých XML editorů. Ke zjednodušení celého procesu je také připravována speciální aplikace – „MUDRLite Forms Designer“. Je-li konfigurační soubor hotov, je třeba MUDRLite nějakou dobu testovat a následně vyladit přesně pro potřeby lékařů na základě jejich komentářů.

4. Výsledky

Testování aplikace MUDRLite potvrdilo, že tento EHR je dostatečně flexibilní a umožňuje dynamické změny v databázovém schématu vyžadující žádné nebo pouze malé změny v konfiguračním XML souboru. Tento soubor může být vytvářen za pomoci rozličných XML editorů, nicméně v současné době připravujeme aplikaci „MUDRLite Forms Designer“, která tvorbu ještě značně zjednoduší. Dvouvrstvá architektura záznamu odděluje uživatelské rozhraní od úložiště dat. Tato funkčnost umožňuje vzdálený přístup k záznamu. Abychom získali ještě flexibilnější vzdálený přístup plánujeme vyvinout Pocket PC verzi MUDRLite uživatelského rozhraní, která by byla spustitelná na rozličných přenosných zařízeních.

Verifikaci funkčnosti a jednoduchosti MLL jazyka ukazuje, že tento jazyk se zdá být dostatečný pro mnoho aplikací, což je především díky síle jazyka SQL integrovaného v MLL. Nicméně naší snahou je sílu MLL jazyka ještě zvýšit, což plánujeme realizovat přidáním aritmetických výrazů a elementů logických podmínek přímo do MLL. Stále jsme si však vědomi, že je třeba přitom zachovat jazyk MLL co možná nejjednodušším.

5. Závěr

Naším zájmem obecně je zvýšení kvality EHR systémů, zjednodušení sdílení a migrace medicínských dat mezi rozličnými EHR systémy a pomoc při postupném překonávání klasického zdravotního záznamu založenému na zprávách psaných volným textem. Toto je cesta, jakou může být zvýšena kvalita léčebné péče, což bude mít přínos v první řadě pro pacienta. Většina poskytovatelů zdravotní péče využívají různých forem EHR. Bohužel tyto systémy většinou neposkytují dostatečnou možnost strukturalizace záznamu. Lékaři sbírající specifické informace (například pro potřeby medicínských studií) jsou často nuceni využívat nestandardních způsobů ukládání informací například v kancelářském software.

My zde prezentujeme univerzální řešení nazvané MUDRLite představující jednoduchý způsob vytvoření EHR šitého na míru. V první instanci nasazujeme toto řešení do oddělení neurovaskulární chirurgie Ústřední vojenské nemocnice v Praze. Zároveň jsme

začali spolupráci s katedrou stomatologie na 1. lékařské fakultě Univerzity Karlovy v Praze, která by měla následně vyústit v rozšíření aplikace MUDRLite do českých stomatologických ambulancí. Věříme, že tímto se aplikace MUDRLite dostane do rutinního praktického používání, což doufáme bude přínosem pro zdravotní péči v České republice.

Poděkování

Práce je částečně podporována projektem číslo LN00B107 Ministerstva školství, mládeže a tělovýchovy České republiky.

Literatura

- [1] Rauch J.: Mining for Statistical Association Rules. In: Fong J., Ng M.K. (eds.): The Fifth Pacific/Asia Conference on Knowledge Discovery and Data Mining, University of Hong Kong (2001) 149-158
- [2] Iakovidis I.: Towards Personal Health Record: Current Situation, Obstacles and Trends in Implementation of Electronic Healthcare Record in Europe. *Int J Med Inform* 52, Nos. 1-3 (1998) 105-115
- [3] Pierik F. H., Ginneken A. M., Timmers T., Stam H., Weber R. F.: Restructuring Routinely Collected Patient Data: ORCA Applied to Andrology. In: Bemmell J. H., McCray A. T. (eds.): Yearbook of Medical Informatics 98, Health Informatics and the Internet, Schattauer, Stuttgart (1998) 257-263
- [4] Ginneken A. M. The Computerized Patient Record: Balancing Efort and Benefit, *Int J Med Inform* 65, (2002) 97-119
- [5] Hanzlicek P.: Development of Universal Electronic Health Record in Cardiology. In: Surján G., Engelbrecht R., McNair P. (eds.): Health Data in the Information Society. Amsterdam, IOS Press (2002) 356-360
- [6] Zvarova J., Hanzlicek P., Spidlen J.: Electronic Health Record in Cardiology: Pilot Application in the Czech Republic. In: MIST2002 Proceedings, Taiwan, John Wiley & Sons Pte Ltd. (2002) 10-13
- [7] Spidlen J., Adaskova J., Heroutova H., Zvarova J., Mazura I.: Statistical Processing of Genetic Information in the MUDR Health Record. In: Vignerova J., Riedlova J., Blaha P. (eds.): Anthropology and Society, Charles University, Prague (2003) 190
- [8] Hanzlicek P., Spidlen J., Heroutova H.: User interface of MUDR Electronic Health Record, *Medical Informatics Europe Proceedings CD* (2003)
- [9] Hanzlicek P., Spidlen J., Zvarova J.: The Internet in Connecting Electronic Health Record Mobile Clients, Technology and Health Care, Vol. 10, Num 6, IOS Press (2002) 502-503
- [10] Spidlen J., Hanzlicek P.: Implementace formalizovaných lékařských doporučení v elektronickém zdravotním záznamu MUDR. In: Svatek V. (ed.): Znalosti 2003 proceedings of the 2nd annual conference, VŠB-TU Ostrava (2003) 386-391
- [11] Reingruber M. C., William W. G.: The Data Modeling Handbook : A Best-Practice Approach to Building Quality Data Models, John Wiley & Sons, Inc. (1994), ISBN: 0-471-05290-6
- [12] Microsoft Corporation: Microsoft® .NET Framework 1.1 Class Library Reference Volume 7: System.Windows.Forms, System.Drawing, and System.ComponentModel, Microsoft Press® (2003), ISBN: 0-7356-1818-6
- [13] Piattini M., Diaz O.: Advanced Database Technology and Design, Artech House, Inc. (2000), ISBN: 0-89006-395-8

Adresa pro korespondenci

Josef Špidlen, EuroMISE centrum – Kardio, Ústav Informatiky AV ČR,
Pod Vodárenskou věží 2, 182 07 Praha 8, Česká republika
Email: spidlen@euromise.cz