

# Implementace formalizovaných lékařských doporučení v elektronickém zdravotním záznamu MUDR

Josef Špidlen, Petr Hanzlíček

EuroMISE Centrum – Kardio, Ústav informatiky AV ČR  
Pod Vodárenskou věží 2  
182 00 Praha 8  
spidlen@euromise.cz, hanzlicek@euromise.cz

**Abstrakt.** Jedním z výzkumných úkolů EuroMISE Centra – Kardio je aplikovaný výzkum v oblasti elektronického zdravotního záznamu (EHR). V rámci tohoto výzkumu zde vznikla pilotní 3-vrstvá aplikace MUDR, ve které jsou data reprezentována pomocí dvou grafových struktur. První strukturou je tzv. znalostní báze, popisující vztahy mezi druhy sbíraných údajů (sémantickými typy). Vlastní zaznamenané hodnoty tvoří druhou hierarchickou strukturu, odkazující se na zmíněné sémantické typy. Tento přístup umožňuje univerzálně a jednotně reprezentovat libovolné informace, na druhou stranu vyžaduje specifické přístupy pro jejich získávání. Aplikační vrstva EHR MUDR obsahuje systém pro podporu rozhodování lékaře, využívající formalizovaných lékařských doporučení. Knihovny, realizující kontrolu lékařských doporučení, využívají definované aplikační rozhraní MUDR API k získávání relevantních informací ze záznamu pacienta. V rámci pilotní aplikace byla implementována knihovna elektronických lékařských doporučení pro léčbu hypertenze, na které bylo MUDR API otestováno.

**Klíčová slova:** elektronický zdravotní záznam, MUDR, lékařská doporučení

## 1 Úvod

Jedním z výzkumných směrů Evropského centra pro medicínskou informatiku, statistiku a epidemiologii – Kardio (EuroMISE Centra – Kardio) je aplikovaný interdisciplinární výzkum v různých oblastech medicínské informatiky. Součástí tohoto výzkumu jsou také otázky reprezentace medicínských znalostí a vývoj univerzálního elektronického zdravotního záznamu (EHR). Navržený 3-vrstvý systém „MULTimedia Distributed Record“ (MUDR) [1] byl inspirován některými evropskými projekty z této oblasti, zejména projektem I4C/TripleC a standardy CEN TC251. V textu je popsáno, jak jsou v datové vrstvě MUDR uchovávány informace s využitím tzv. znalostní báze. Dále je zmíněno aplikační rozhraní MUDR API, které umožňuje k těmto informacím přistupovat. Toto API využívají knihovny LD k získání informací ze zdravotního záznamu pacienta, na základě kterých poskytují podporu v dalším rozhodování lékaře. V rámci pilotní aplikace byla implementována knihovna elektronických lékařských doporučení pro léčbu hypertenze, která formalizuje „1999 WHO/ISH Guidelines for the Management of Hypertension“.

## 2 Architektura MUDR

Elektronický zdravotní záznam MUDR je založen na třívrstvé architektuře s datovou, aplikační a klientskou vrstvou. Tato dekompozice umožňuje oddělit jednotlivé funkční části systému a umožňuje uživatelským rozhraním pracovat na různých platformách. Funkce datové vrstvy spočívá v ukládání informací a kontrole základní referenční integrity dat. Aplikační (funkční) vrstva zajišťuje zjednodušený pohled na databázi a zpřístupňuje data uživatelským rozhraním a různým klientským aplikacím. Tito klienti komunikují s aplikační vrstvou pomocí MUDR API definovaného ve formě platných XML dokumentů odpovídajících příslušnému XML schématu. K přenosu tohoto XML je využito HTTP serveru na aplikační vrstvě. XML je na straně klienta zabaleno do HTTP POST žádosti a tím je přeneseno na vstup CGI skriptu, který jej dále předává aplikační vrstvě.

Lékařská doporučení jsou připojována ve formě dynamických knihoven ke službě aplikační vrstvy. Jejich komunikace probíhá na úrovni win32 prostřednictvím exportovaných funkcí a sdílené paměti. Těmto funkcím je předáváno XML odpovídající MUDR API. V tomto ohledu získávají knihovny informace ze zdravotního záznamu pacienta stejným způsobem jako MUDR klienti.

## 3 Reprezentace dat v MUDR

### 3.1 Znalostní báze

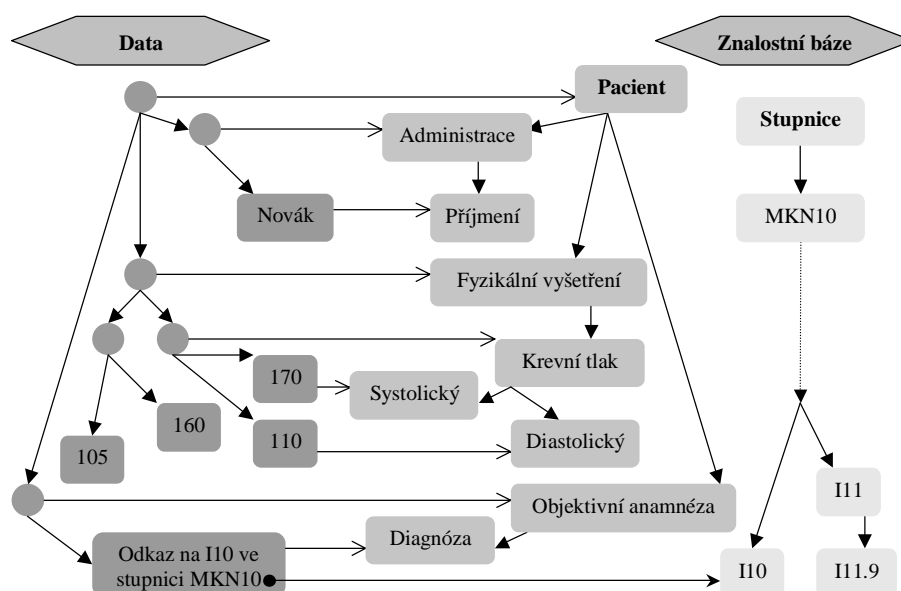
Vzhledem k požadavku na dynamicky se měnící množinu sbíraných druhů údajů nebylo možno využít jako základ pro zaznamenání hodnot klasickou relační tabulkou se sloupečky odpovídajícími jednotlivým ukládaným veličinám. Místo toho je využívána dynamická struktura – tzv. *znalostní báze*, zachycující jednotlivé druhy sbíraných údajů a vztahy mezi nimi.

Znalostní báze je *orientovaný graf*  $G = (V, E)$ . Vrcholy grafu, prvky  $u \in V$ , označujeme jako *uzly znalostní báze*. Každý uzel znalostní báze je čtveřice (*id, jméno, dtyp, platnost*), kde *id* je jednoznačný identifikátor v rámci všech uzlů, *jméno* je mnemotechnický maximálně 10-znakový řetězcový identifikátor, *dtyp* je tzv. *datový typ* a *platnost* obsahuje základní administrativní údaje například o tom, kdo a kdy vložil uzel do znalostní báze. Hrany grafu  $e \in E$  jsou také čtveřice ( $u_1, u_2, etyp, platnost$ ), kde  $u_1, u_2 \in V$  označují, odkud kam hrana vede, *etyp* je typem hrany určujícím druh vztahu mezi koncovými uzly a *platnost* opět uchovává administrativní údaje – tentokrát týkající se příslušné hrany.

Mezi typy hran existuje jeden dominantní typ – „*inferior*“ – vedoucí od rodiče k potomkovi a určující tímto *hierarchický vztah* mezi uzly znalostní báze. Vyjme-li ze znalostní báze hrany všech ostatních typů, získáme orientovaný les s několika málo stromy. Tyto stromy označujeme jako *domény*. Každá doména sdružuje uzly sloužící ke stejnému účelu. Typicky jedna doména slouží pro uchování množiny všech sbíraných údajů o pacientovi. Uzly této domény označujeme jako *sémantické typy*.

Jiné domény uchovávají například hierarchicky strukturovanou mezinárodní klasifikaci nemocí MKN10, anatomicko-terapeuticko-chemickou klasifikaci chemických názvů ATC či klasifikaci měrných jednotek SI nebo léky dostupné na českém trhu. Hrany dalších typů umožňují vložit do této struktury další informaci. Představit si můžeme například typy hran, které označují ekvivalenci dvou sémantických typů nebo indikace a kontraindikace lékových skupin.

Uzly se stejným otcem označujeme jako *bratry* a požadujeme, aby jméno každého uzlu bylo mezi bratry vždy jednoznačné. Uzly nemající otce označujeme jako *kořeny domén* a i mezi všemi těmito kořeny musí být jméno uzlu jednoznačné. Datový typ uzlu je významný především pro sémantické typy. Rozlišujeme 3 základní datové typy: číslo, boolean a textový řetězec, 4 multimediální datové typy: obraz, audio, video a binární soubor a 2 referenční datové typy: referenci na data a referenci na uzel znalostní báze. Kromě těchto existuje pomocný datový typ adresář. Datové složky (viz dále) s datovým typem adresář mají prázdnou hodnotu a slouží převážně k seskupování svých podsložek.



Obr. 1. Princip reprezentace elektronického zdravotního záznamu – příklad.

### 3.2 Konkrétní uchovávané hodnoty

Konkrétní zadaná data tvoří *orientovaný les*  $L = (D, E)$ . Záznamy jednoho pacienta odpovídají vždy jednomu stromu v tomto lese. Vrcholy  $d \in D$  reprezentujeme jako datové složky – čtveřice (*id*, *sémantický typ*, *hodnota*, *administrativní údaje*), kde *id* je jednoznačný identifikátor v rámci všech datových složek, *sémantický typ* je odkazem na sémantický typ této datové složky ve znalostní bázi a *hodnota* je

konkrétní instancí datového typu, který implicitně vyplývá z typu sémantického. Pod položkou *administrativní údaje* se skrývá množství informací určujících kdo a kdy, zadal, smazal či potvrdil příslušnou hodnotu, jaké je období platnosti hodnoty, spolehlivost zadaného údaje apod. Hrany  $e \in E$  jsou tentokrát netypané, vytvářejí pouze hierarchický vztah „otec – syn“ a  $E$  lze matematicky vyjádřit jako podmnožinu kartézského součinu  $D \times D$ . Navíc pro orientovaný les  $L$  musí platit:

- Každá datová složka má otce nebo její sémantický typ je kořenem domény.
- Má-li datová složka  $d$  otce  $o$ , pak sémantický typ  $d$  je syn sémantického typu  $o$ .

Zjednodušeně a laicky řečeno, podmínky zaručují, že datová část odpovídá znalostní bázi a že v ní „neplavou“ volné uzly. Malý příklad několika uložených dat, znalostní báze a příslušných odkazů je znázorněn na obrázku číslo 1.

## 4 Získávání informací z EHR MUDR

### 4.1 Aplikační rozhraní pro přístup k datům

Pro účely komunikace s aplikační vrstvou EHR MUDR bylo definováno aplikační rozhraní MUDR API. Toto rozhraní má formu platných XML dokumentů odpovídajících XML Schématu MUDRAPI.xsd [2]. Existují 2 druhy platných XML dokumentů: směrem k aplikační vrstvě je používán dokument sestávající z identifikace uživatele a posloupnosti příkazů, směrem od aplikační vrstvy je používán dokument složený z posloupnosti odpovědí. Každému příkazu je přiřazen jednoznačný identifikátor v rámci XML dokumentu. Tento identifikátor kopíruje aplikační vrstva do odpovědí, aby klient správně a jednoduše rozpoznal, k jakému příkazu se příslušná odpověď váže. Ke každému příkazu jsou definovány možné odpovědi.

Aplikační rozhraní MUDR API bylo navrženo tak, aby obsahovalo relativně malou množinu příkazů a přitom umožňovalo plnohodnotně pracovat s elektronickým záznamem včetně úprav znalostní báze, správy uživatelů a jejich pravomocí, práce s výrazy v různých světových jazycích apod. Celkem existuje něco přes 20 příkazů, z nichž pro potřeby knihoven lékařských doporučení a získávání informací ze záznamu pacienta jsou relevantní 4 uvedené v tabulce číslo 1.

**Tabulka 1.** Vybrané příkazy aplikačního rozhraní MUDR API.

Příkaz	Popis příkazu
<i>get_knowledge_domains</i>	vypíše známé znalostní domény ve formě kořenových uzlů příslušných stromů (kořenů domén)
<i>get_knowledge_node</i>	vypíše detailní informace o uzlu znalostní báze
<i>get_node_neighbours</i>	vypíše seznam sousedních uzlů k uzlu znalostní báze
<i>get_data_files</i>	vrátí určitou datovou složku z elektronického záznamu

## 4.2 Prohledání znalostní báze knihovny lékařských doporučení

Každý uzel znalostní báze je jednoznačně určen číselným identifikátorem. Tento identifikátor je ale průběžně generován a knihovny lékařských doporučení jej na začátku komunikace neznají. Vzhledem k tomu, že každý uzel je mezi svými bratry identifikován pomocí jména, je možné tento uzel globálně jednoznačně identifikovat posloupností jmen v cestě, která k němu přes všechny předky od kořene vede. Zápis této cesty v tečkové notaci nazýváme *úplné jméno uzlu* a je aplikačním rozhraním podporováno jako alternativní vyjádření každého uzlu.

Chce-li kdokoli prohledat množinu všech sémantických typů, začne typicky příkazem `get_knowledge_domains` a dále aplikuje prohledávání grafu znalostní báze do hloubky pomocí příkazů `get_knowledge_node` a `get_node_neighbours`. Počet nutných iterací s aplikační vrstvou v tomto případě odpovídá hloubce grafu znalostní báze, což může být pro potřeby knihoven LD příliš mnoho. Tyto knihovny využívají explicitní znalosti o struktuře znalostní báze danou buď konfigurací nebo částečně zjištěnou při nahrávání knihovny. Např. knihovna hypertenzních LD předpokládá, že systolický krevní tlak odpovídá uzlu `PATIENT.PHYS_EXAM.BP.SYSTOLIC`. Při své inicializaci navíc knihovna ověří, zda takovýto uzel existuje a případně zkusí mezi `BP` a `SYSTOLIC` vložit meziuzel `RA` a `LA` pro případ, že je znalostní bázi rozlišován sémantický typ krevního tlaku na pravé a levé ruce. Pro potřeby hypertenzní knihovny LD je analyzováno přes 50 sémantických typů ovlivňujících rozhodování.

## 4.3 Vyhledání informací v záznamu pacienta

K získávání informací v EHR MUDR slouží příkaz `get_data_files` MUDR API. K vyhledání hodnot lze tento příkaz použít třemi různými způsoby.

- *Dotaz na data pomocí jejich id.* Vrácena je kompletní informace o datové složce. V případě multimediálních dat lze dokonce specifikovat některé vlastnosti hodnoty datové složky a systém provede příslušné transformace jako například převod do určitého formátu obrazu či nastavení úrovně komprese.
- *Dotaz na data pomocí jejich společného otce.* Vráceny jsou všechny datové složky, která mají otce specifikovaného dotazem. U těchto multimediálních dat je vynechána konkrétní hodnota datové složky.
- *Vyhledáváním dat pomocí zadané klíčové hodnoty, uzlu znalostní báze a upřesňujících parametrů.* Parametry lze zadat například minimální či maximální číselnou hodnotu, vzorkem textového popisu apod.

Knihovny LD využívají převážně druhou možnost – vyhledávání datových složek pomocí společného otce. Při žádosti o posouzení stavu pacienta dostane knihovna LD odkaz na kořenovou datovou složku příslušného pacienta. Díky tomuto odkazu může knihovna LD do šířky projít a získat celý zdravotní záznam pacienta. Implicitně nejsou vráceny obsáhlé multimediální hodnoty, které pravděpodobně nebudou třeba, čímž je zmenšeno množství informací vyměňovaných mezi aplikační vrstvou a knihovnou LD. Knihovna navíc při procházení stromu datových složek vynechává větve, které obsahují pro konkrétní účely irelevantní informace.

#### 4.4 Zpracování informací knihovnou hypertenzních LD

Po načtení všech informací ze záznamu pacienta odpovídajících relevantním sémantickým typům začne knihovna s jejich systematickým tříděním. U každé datové složky je brána v úvahu informace o spolehlivosti, stavu, datu zadání a období platnosti z administrativních údajů. Každý sémantický typ má navíc svá specifika při zpracovávání. Např. informace, že pacient někdy kouřil, je navždy brána jako jeho rizikový faktor. U krevních tlaků bere knihovna v úvahu maximálně pět nejnovějších naměřených hodnot s maximálním stářím 1 rok a z nich se snaží získat relevantní hodnotu a odhadnou tendenci změny tlaku v poslední době pomocí lineární regrese. Na rozdíl od formalizovaných LD [3] je zde využíván časový faktor změny údajů.

## 5 Závěr

Knihovna hypertenzních LD ověřila funkčnost získávání informací pomocí MUDR API. V dalším vývoji je zvažováno, jak upravit API tak, aby došlo ke snížení počtu přenášených zpráv bez zvýšení objemu zbytečně přenášených dat. Jisté řešení se nabízí s využitím predikce informačních potřeb každé knihovny LD.

## Reference

1. Hanzlíček P. Development of Universal Electronic Health Record in Cardiology. MIE 2002. *Proceedings of MIE 2002*. IOS Press 2002, Amsterdam. ISBN 1-58603-279-8, ISSN 0926-9630
2. Špidlen J., EuroMISE Centrum – Kardio. <http://www.euromise.cz/MUDRAPI.xsd>
3. Zvárová J., Peleška J., Hanzlíček P., Zvára K. Enhanced care of hypertensive patients using Internet. *Technology and Health Care Vol.9 No.6 2001*, IOS Press 2001, Amsterdam, str. 487-488. ISSN 0928-7329

### Annotation.

*Implementation of the formalized medical guidelines in the electronic health record MUDR*

Important research task of the EuroMISE Center – Cardio is applied research in the field of electronic health record (EHR). In this frame a pilot 3-layer universal EHR application “MULTimedia Distributed Record” (MUDR) was developed. Article describes the architecture of the developed system, the graph representation of patient data and the access of the integrated medical guideline libraries to the MUDR health record using the application interface MUDR API. Finally, the extraction of the relevant information about the patients health using the implemented hypertension library is mentioned.