

POČÍTAČOVÁ PODPORA DOMÁCÍ LÉČBY A TELETSTOVÁNÍ

Petr Novák, Vladimír Eck, Olga Štěpánková

Anotace

Článek se věnuje tvorbě počítačové online podpory umožňující nejen domácí léčbu (rehabilitaci), ale rovněž průběžné monitorování výsledků léčebných úloh pacienta lékařem. Zahrnuje rovněž možnosti teletestování pacientů se zaměřením například na psychologické testy, detekci Parkinsonovy nemoci, testy pro předškolní děti a řadě dalších využití.

Snaha autorů spočívá ve vytvoření tzv. „frameworku“ pro podporu těchto počítačových aplikací. Tedy možnosti snadného vytváření jak desktopových, tak webových aplikací představující zejména rehabilitační, monitorovací a testovací úlohy. Mnoho léčebných úloh využívá adaptací na měnící se stav pacienta, a tím zajišťuje stálou efektivnost léčebné metody i mezi průběžnými kontrolami u lékaře.

Každému pacientovi je poskytnut jedinečný přístupový kód, pod kterým jsou výsledky všech vypracovaných úloh a testů odesílány. Podle identifikace pacienta jsou tato data zpracovávána a předávána na pracoviště odpovědného lékaře (dozoru). Získané výsledky se nejprve automaticky hodnotí na detekci nežádoucích stavů a poté se prezentují lékaři vhodnou zejména grafickou formou. Z dat lékař dále získá informace o časovém rozložení domácí léčby (průběžná nebo jednorázová), možnost porovnání kvality tzv. „domácích výsledků léčby“ pacientů s „kontrolními výsledky“ v ordinaci lékaře a rovněž přehled celkové stability naměřených výsledků.

V současné době je vytvořený „framework“ prototypově využíván na 5 lékařských pracovištích pro zhruba 100 pacientů v domácí léčbě zrakových poruch. Dále je zahájen projekt pro rehabilitaci pohybového aparátu a připravují se úlohy pro hodnocení stavu zejména paměti, postřehu a dalších kognitivních činností zvláště u stárnoucí generace. Již první výsledky jsou pro lékařskou praxi velmi povzbudivé a přínosné. Nejen, že lze snadno detekovat, jaké dny v týdnu jsou pro pacienta při cvičení vhodnější, jak domácí trénink ovlivňuje výsledky kontroly u lékaře, ale z výsledků mnoha úloh lze navíc detekovat dokonce i dříve neodhalené poruchy nebo další schopnosti a dovednosti testované osoby.

Klíčová slova

Vzdálený dozor, domácí rehabilitace, teletestování, adaptivní úlohy, automatická detekce nežádoucích stavů

1. Úvod

V současné době stojíme před velkou příležitostí, která však může být společná pro mnoho i značně odlišných oblastí, a to přenesení mnoha, nejen lékařských činností do domácího nebo jakkoli vzdáleného prostředí. Jednou velmi

významnou motivací je samozřejmě úspora personální a finanční. Té se však tento článek přímo nevěnuje, i když z navrženého řešení toto zcela vyplývá.

Velmi se mluví o tzv. „domácí rehabilitaci“ neboli léčbě mnoha typů poruch či postižení v domácím prostředí. Toto řešení obsahuje v podstatě dvě velké výhody současně. Na jedné straně již zmíněné úspory z hlediska ošetřovatele neboli lékaře a na straně druhé umožnit pacientovi rychlejší (ne však nutně kvalitnější) léčbu a tím dřívější návrat na své pracoviště. Pro většinu pacientů je zkrácení léčebného procesu stimulační a tudíž tzv. domácí rehabilitaci zcela určitě uvítají. V tomto směru je nutno vytvořit vhodné úlohy, například formou počítačových programů, které pacient v domácím prostředí „aktivně“ využívá. Úlohy jsou samozřejmě zaměřeny na konkrétní potřeby dané poruchy. Pro skutečnou maximalizaci efektivity domácí léčby se musí úlohy automaticky adaptovat na stav pacienta a tím jej stále vhodně stimulovat. Takto lze účelně překlenout intervaly mezi návštěvou pacienta u lékaře. Tj. mezi hloubkovými kontrolami u lékaře kdy mohou být parametry domácích úloh „přenasraveny“ podle úsudku lékaře nebo „doporučeny zcela nové úlohy“. Pravidelné kontroly u lékaře jsou bezpodmínečně nutné pro zajištění skutečně správného léčebného procesu. Úlohy pro domácí prostředí jsou v podstatě doplňkové, a tudíž nemohou nahradit například přesné měřicí přístroje v ordinaci lékaře.

Další velkou skupinou uživatelů vzdáleného teletestování (nikoli přímo monitorování) mohou být například libovolné profese, kde je nutno buď periodicky, nebo pouze namátkově kontrolovat schopnosti a dovednosti pracovníka. Jako vzor lze uvést dispečery velinů elektráren nebo řízení dopravy. Zde nemusí jít vždy přímo o přezkoušení, ale pouze o vyřešení určitého speciálního „kvízu“ na začátku pracovní směny a tím ověření aktuálních schopností jako jsou postřeh, rychlost řešení problému, přehled a další. Například porovnáním aktuálního řešení s předchozími lze snadno odhalit tzv. „právě špatnou formu“ pracovníka a případně jej ze směny odvolat a tím předejít mnohem větším nepříjemnostem. Jde tedy na první pohled o „malíček“, která je schopna upozornit na velmi „podstatný“ a přesto na první pohled zcela „skrytý“ problém.

Poslední zde zmíněnou oblastí pro použití teletestování je stárnoucí generace v domácím prostředí (samozřejmě i specializovaná zařízení). V této oblasti se velmi často setkávají všechny typy činností a to: monitorování, sledování a dnes v podstatě i tělesná a duševní cvičení. Zde pod pojem monitorování neuvážujeme přímo měření životních funkcí, ale spíše celkové aktivity osoby. Tedy sběr dat z prostředí (průběh pohybu osoby, zapnutí TV přijímače atd.) a jejich odesílání a monitorování za účelem detekce „nestandardních“ způsobů chování. Z našeho pohledu jsou však mnohem zajímavější počítačové úlohy určené zejména pro trénink neboli udržování různých kognitivních schopností člověka, které se u něho v pokročilém věku mohou postupně snižovat. Za tímto účelem lze rovněž vytvořit soubor vhodných úloh, které osoba průběžně využívá. Úlohy však nelouží pouze k tréninku těchto

schopností, kde mezi nejzákladnější patří samozřejmě paměť nebo myšlení, ale velký přínos spočívá v dlouhodobém sběru dat a tím možnosti hodnocení případného útlumu některých z těchto lidských schopností. Na základě čehož lze velmi snadno odhadnout, zda je daná osoba ještě stále schopna zcela samostatné činnosti.

I když se na první pohled jedná o velmi odlišné oblasti (domácí léčba, vzdálené testování, stárnoucí generace a ještě další zde neuvedené), tak přesto mají mnoho společného. Mezi tyto hlavní společné body lze zahrnout: vytvoření určité skupiny cílených testů nebo úloh, jejich vzdálené poskytování, sběr získaných dat / výsledků, jejich do velké míry automatické hodnocení a nakonec zobrazení odbornému personálu. Naskytá se tedy možnost vytvoření univerzálního „frameworku“ poskytujícího na jedné straně podporu pro tvorbu konkrétních testů nebo úloh (v ideálním případě i ne programátorům) a na straně druhé podporu pro použití algoritmů pro automatické zpracování velkého množství získaných dat (v ideálním případě vědeckými pracovníky). Vlastní činnost tohoto „frameworku“ by měla spočívat v jednoznačné identifikaci uživatele (v podstatě pacienta), minimalizaci úsilí při tvorbě testů a úloh, zajištění přenosu výsledků z testů a úloh do centrálního místa (samozřejmě podle oblasti použití), minimalizaci úsilí pro začlenění algoritmů pro zpracování dat a na závěr poskytnutí hodnocení pomocí různých typů reprezentací (grafické, tabulkové).

2. Základní požadavky na tvorbu framework

Požadavky na tvorbu takového frameworku lze rozdělit na 3 základní skupiny:

1. Tvorba testů a úloh (typy testů a úloh, úroveň interakce, rychlost odezvy, způsoby ovládnání, ...)
2. Organizace neboli správa (jednoznačná identifikace uživatelů, zamezení úniku osobních údajů, ...)
3. Technické (použitý programovací jazyk, webové technologie, operační systém, ...)

První bod je nejobsáhlejší, a bude tudíž podrobněji popsán v následující kapitole. Další dva body jsou mnohem snadnější. Jednoznačná identifikace uživatele neboli pacienta a přitom zamezení zveřejnění jakýchkoliv osobních údajů je zcela primární požadavek v této oblasti. Jako nejjednodušší řešení pro jednoznačnou identifikaci klienta by bylo možno využít například tzv. „hash“ z rodného čísla (případně i jména). Jde o postup jak z jednoho unikátního řetězce vytvořit jiný unikátní řetězec při použití pouze jednosměrného převodu (převod opačným směrem je v podstatě nemožný). I když je tento postup velmi bezpečný, tak však neřeší jeden z našich velkých problémů a to současně jednoznačné přiřazení pacientů ke konkrétnímu lékaři. Tedy snadné získání informace od jakého pacienta se výsledky zasílají k jakému lékaři (komu jsou přístupné). Z tohoto důvodu bylo přistoupeno k identifikaci nejen pacientů, ale i lékařů podle číselných identifikačních kódů. Tento číselný kód se skládá ze tří částí a to: místa působení lékaře, identifikace lékaře v tomto místě

a současně z pořadového čísla pacienta u tohoto lékaře. Takto lze velmi snadno identifikovat nejen pacienta, ale současně i k jakému lékaři přísluší a kde se jeho lékař nachází. Tedy kam mají směřovat pacientova data. Pro správné vložení je tento identifikační číselný kód chráněn heslem. Po dobu přenosu a zpracování jsou pacientova data opatřena pouze tímto identifikačním kódem a teprve až po vyhodnocení jsou vložena do databáze u lékaře ke jménu pacienta. Převodní tabulku pacienta na identifikační kód vlastní pouze lékař. Toto nezpůsobuje žádná omezení ani například při jejich vědeckém zpracování a detekci nežádoucího stavu u některého z pacientů. O takto detekovaném stavu je informován lékař a jeho databáze automaticky poskytne doplnění identifikace pacienta o jeho skutečné jméno.

Technické řešení je vždy určitý kompromis mezi dostupnými technologiemi, přesněji řečeno jejich složitostí a znalostmi tvůrců vytvářeného systému. Rovněž v našem projektu bylo nutno při výběru programové / technologické platformy zvážit mnoho požadavků jako jsou například:

- Možnost tvorby v podstatě libovolně interaktivních úloh s téměř libovolnou multimediální podporou a grafikou.
- Využívat především programovací jazyky pro tvorbu komplexních aplikací a nikoli různé skripty s omezenými schopnostmi.
- Využívat technologie dostupné současně jak pro tvorbu desktopových tak i webových aplikací s velkou přenositelností vytvořeného kódu.
- Umožnit včlenění různých speciálních výpočetních algoritmů přímo do úlohy uživatele (například algoritmy na adaptivnost úlohy).
- Možnost vytvoření aplikací i s velmi omezeným nebo dokonce bez dlouhodobého přístupu na internet.
- Stejně část úlohy přesunout na klienta a spojení se serverem využívat (pokud možno) pouze pro spuštění úlohy a poté odesílání řešení od uživatele.
- Možnost využití libovolných externích zařízení od běžných WEB kamer až po specializované ovladače.

Bereme-li v úvahu zde zmíněné technické požadavky, tak jsou pro naše řešení v podstatě nevhodné jakékoli statické webové stránky (HTML) nebo jakkoli ze serveru generované stránky (ASP, PHP). Pro tvorbu úloh využívající jak připojenou WEB kameru nebo různé speciální USB zařízení a tedy obsahující i celkem složité algoritmy (přesnost kresby, chvění ruky, ...) jsou rovněž nevhodné technologie jako JavaScript nebo Flash. Pokud neustoupíme ze zde již podrobněji hodnocených požadavků a současně uvažujeme o začlenění některých speciálních výpočtů (funkcionální jazyky, náročné matematické algoritmy) přímo do webové aplikace, nebo dokonce požadujeme vytvoření v podstatě téměř jedné aplikace (jednoho programového kódu) jak pro desktop (stolní počítače i bez přístupu na internet) tak i web (přístup z libovolného místa), tak se okruh použitelných technologií velice rychle zužuje. V podstatě jediná dostupná technologie splňující většinu zmíněných požadavků je Microsoft .NET [1] nebo některé jeho vytvářené „klony“ jako například Mono

(určené pro Linux). Jelikož každé pro má i své proti, tak „určitou“ nevýhodou této technologie je její (zatím stále) velká vázanost na operační systém Microsoft Windows. Tato vázanost není však v našem případě nikterak limitující, neboť se nejedná o aplikace pro širokou veřejnost, ale pro cílenou skupinu uživatelů. Současně však přenos této technologie na jiné operační systémy je aktivně podporována i ze strany velkých firem.

Výběrem technologie Microsoft .NET a jejích součástí jako jsou WPF nebo SilverLight lze vytvářet aplikace současně jak pro desktop, tak i pro web s minimální úpravou. Další nesporná výhoda spočívá v možnosti přímé integrace do webové aplikace i vědeckých výpočtů například pomocí jazyků jako jsou F#.NET nebo Prolog.NET. Tyto aplikace lze samozřejmě vytvářet v jakémkoli programovacím jazyce dostupném pro .NET, ale v našem případě byl jako hlavní programovací jazyk zvolen C#. Jde o skutečný vysokourovňový programovací jazyk poskytující množství schopností (generické typy, LINQ, paralelizace, ...) a to i ve webové aplikaci.

3. Požadavky na tvorbu testů nebo úloh a přenos dat pro framework

Na první pohled nemusí být patrná souvislost mezi požadavky na tvorbu různých typů úloh a použitou technologií pro tvorbu frameworku. Opak je však často pravdou. Ve většině případů požadavek na technologické řešení vychází teprve z požadavků na vlastní aplikace, tedy podle rozmanitosti testů a úloh. V našem případě však můžeme tyto požadavky v podstatě otočit a to z jednoho zásadního důvodu. Jako webová technologie byla zvolena platforma disponující skutečně velkými schopnostmi (srovnatelnými s desktopovými aplikacemi) a není tedy nutno při tvorbě vlastních úloh nikterak brát zásadní zřetel na její omezení. Zcela opačný případ by nastal při výběru platformy jako například PHP, kde bychom museli velmi zřetelně uvažovat její omezení.

Zde již tedy nevybíráme technologii podle požadavku na testy a úlohy, ale naopak snažíme se využít všech schopností zvolené technologie pro tvorbu skutečně potřebných a vhodných úloh. Webový framework poskytující podporu pro snadnou tvorbu úloh bude tedy vytvořen ve zvolené technologii (Microsoft .NET SilverLight) a tudíž bude schopen poskytnout téměř srovnatelné možnosti s desktopovou aplikací, a to od jednoduchých dialogů, přes složité výpočty až po komplexní 3D scény.

Požadavky na vytvářený framework z hlediska úloh lze dále rozdělit do dvou oblastí a to přenos dat do / z úlohy a vlastní tvorba úlohy. Nejprve tedy přenos dat. Ve většině případů se bude jednat o výsledky úloh a testů a ty musí být uloženy v některém standardním formátu z důvodu jejich možného přenosu přes různé komunikační kanály a zpracování libovolným cílovým systémem a algoritmem. Přenášená data sice budou ve formátu XML, ale ten stanovuje pouze jejich obecnou strukturu a nikoli jejich význam nebo dokonce popis. Nejen pro vlastní přenos, ale i různá (většinou dočasná) ukládání, bude využito samo-popisného formátu z tzv. „Univerzálního úložiště“ vytvořeného přímo za účelem přenosu a ukládání nejen libovolného typu dat, ale současně

informací o nich. Popis Univerzálního úložiště a jeho význam je uveden v [2,3]. V podstatě každá uložená položka nese informaci nejen o své číselné hodnotě (čas, skóre, ...) nebo o svém významu z celkového pohledu (konfigurace, výsledek, ...), ale současně o svém zařazení v souboru naměřených dat (první kolo, druhé kolo, ...).

Vytvářený framework musí poskytnout podporu pro tvorbu mnoha typů úloh od běžných dotazníků, přes jednoduché doplňovací testy až po komplexní pohyblivé scény pro trénink reakce a postřehu uživatele. Pro jednoduchost jsou typy úloh a z nich plynoucí požadavky rozděleny do následujících témat a celků [7].

Dotazníky. Podpora pro sestavení základního typu dotazníku formou: otázka (textové zadání), jedna z možných odpovědí (výběr odpovědi), několik možných odpovědí, vstup textu (textová odpověď). Výstupem dotazníku nebude pouze stav vyplněného formuláře, ale i celkový průběh při vlastním vyplňování formuláře: časové značky u všech akcí uživatele (posuv formuláře, zobrazení číslo otázky, potvrzení odpovědi, časy vepsaných znaků textové odpovědi atd.). Tyto „doplňkové“, zejména časové údaje poskytnou mnoho informací o chování uživatele (posloupnosti a způsob vyplňování), ze kterých lze usuzovat nejen na složitost některých otázek, ale i ohotu nebo dokonce váhavost uživatele při odpovědi. Tyto informace nejsou ze současných testů zatím dostupné.

Doplňování. Ve většině případů se jedná o maticově založené hrací pole. Jedno pole tvoří předložené vzory, jako jsou obrázky nebo písmena a druhé pole pouze části obrázků tyto vzory postrádající. Úkolem uživatele je doplnit podle vzorů chybějící části v druhém poli. Případně jde o různé významové skládačky z předem daného souboru políček. Jako příklad lze uvést mozaiky nebo velmi známé puzzle. Úlohy jsou tedy založeny na mřížce neboli matici, ve které se buď pouze mění stav jednotlivých políček, nebo se políčka i přesouvají. Hodnocení těchto úloh může být dvou základních typů: pouhé vyčíslení počtu chyb nebo komplexnější posouzení vypovídající nejen o tom, jaké vzory jsou pro uživatele problematické, i v jaké posloupnosti byly právě zobrazeny nebo jakou posloupnost volil uživatel při plnění úlohy. První hodnocení je velkou snadné a spočívá v prostém sečtení chyb uživatele. Druhé hodnocení již není tak snadné a většinou vyžaduje algoritmy v jiných než běžných programovacích jazycích (příkladem mohou být různé funkcionální jazyky).

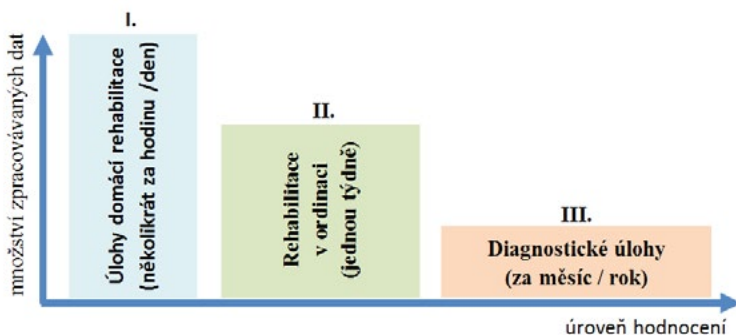
Logické hry. Ty jsou rovněž velmi často založeny na maticové hrací ploše. Jako vzorové příklady lze uvést šachy, dámu, patnáčku, zhasni, pipe-line, sudoku nebo velmi oblíbené bludiště. Na rozdíl od doplňovacích úloh však vyžadují tyto hry i vyšší interakci s uživatelem a rovněž mnohem bohatší vnitřní logiku a samozřejmě i grafické podání. Vnitřní logika je dána buď externím programovým kódem anebo nějakou formou skriptu.

Obkreslování. Tyto úlohy jsou zcela odlišného typu než doposud uvedené. Hrací plocha se v podstatě nemění a představuje jedno velké kreslicí plátno, na něž je nutno umístit nejen libovolný obrázek, ale současně i různé přídavné

informace, jako je vytvářená kresba pacientem. Zajištění vlastního testu není nikterak složité, ale nejsložitější část tvoří jeho hodnocení, kdy jsou využity i výpočetně velmi náročné algoritmy. Například detekce přesnosti kresby uživatele podle vzoru zahrnující hodnocení rychlosti a plynulosti pohybu pera, počet a velikost odchylek atd. Z tohoto důvodu je vhodné, aby navrhovaný framework poskytoval nejen dostatečný výpočetní výkon, ale zejména možnost tyto výpočty „naprogramovat“ neboli vložit.

Postřehové testy. Tyto úlohy představují rovněž velmi odlišnou skupinu úloh, protože jsou založeny převážně na rychlosti zobrazení, tedy na rychlosti změny zobrazované scény. Průběh změny (pozice, tvaru a barvy zobrazovaných předmětů) velmi často není náhodný, ale řídí se pomocí daných zákonitostí. Testy musí být mezi sebou srovnatelné, a proto jsou založeny na předem dané (uložené) posloupnosti akcí, například ve tvaru: čas zobrazení, pozice, tvar obrazce, barva obrazce, časová délka zobrazení atd. Framework tedy musí poskytovat celkem rychlou změnu zobrazované scény a rovněž odezvu na vstupy uživatele.

Z uvedeného vyplývá, že výsledky úloh budou zpracovávány v podstatě ve dvou kolech. První kolo tvoří jejich základní zpracování přímo v úloze za účelem aktuálního hodnocení a případné okamžité změny obtížnosti. Druhé kolo zpracování představuje složitější proces zahrnující použití mnoha sofistikovaných algoritmů například z oboru umělé inteligence, které budou aplikovány v podstatě „offline“ až na soubor dat uložených na serveru nebo v databázi lékaře. Například pro domácí rehabilitaci množství zpracovávaných dat rovněž zcela závisí na úrovni hodnocení (Obr. 1). Vlastní úloha představuje nejnižší úroveň hodnocení I, a tudíž zpracovává velké množství dat a rovněž velmi často (například hodnocení shody kresby se vzorem po každém obrázku). Druhá úroveň hodnocení II již využívá pouze soubor koncových výsledků z předchozí nejnižší úrovně I (například shoda kresby v procentech), ty hodnotí vždy po delším časovém období (týdny až měsíce) a tím poskytuje informace o celkovém zlepšování, stagnaci nebo zhoršování poruchy



Obrázek 1 — Množství zpracovávaných dat v závislosti na úrovni I, II, a III jejich hodnocení

pacienta. Třetí a nejvyšší úroveň hodnocení III představuje v podstatě lékaře. Ten nikterak nehodnotí procentuální shodu kresby, ale již pouze vývoj poruchy za mnohem delší časové období (většinou měsíce) a to ještě velmi často pomocí svých měřících přístrojů. Na základě svého rozhodnutí poté stanovuje další léčebný postup.

4. Realizace navrženého framework

Při vývoji frameworku byly převzaty podklady z mnoha již existujících úloh a testů. Ty byly vytvořeny jako desktopové aplikace v právě již zmíněné technologii Microsoft .NET, a proto posloužily jako vhodný výchozí materiál. Představovaly totiž vzorové zástupce dříve zmíněných skupin úloh. Částečným průnikem a současně i částečným sjednocením jejich programových částí vznikl základ zde popisovaného frameworku.

Vytvořený framework se skládá ze dvou částí. Jednu tvoří v podstatě prázdná aplikace, do které jsou vkládány později vytvořené testy a úlohy. Druhou část tvoří knihovna poskytující podporu nejen pro tvorbu těchto testů a úloh, ale i pro přenos jejich výsledků. Rozdělení na tyto dvě samostatné části je velmi praktické a to z následujícího důvodu. Samostatně vytvořenou knihovnou obsahující všechny podpůrné činnosti lze doplnit v podstatě libovolnou již existující samostatnou aplikaci a tím ji o framework obohatit neboli ji tímto do celkového systému snadno začlenit. Tato schopnost sice nebude využívána velmi často, ale poskytne možnost začlenění již vytvořené a rovněž komplexní aplikace například pro přesné testování do systému se sběrem dat a jejich hodnocením. Přepis již existující aplikace do později vytvořeného frameworku by mohl být časově velmi náročný nebo skoro nemožný (vzhledem k poskytovaným schopnostem frameworku). Samozřejmě nejvíce bude využívána „prázdná aplikace“ doplněná knihovnou pro podporu tvorby vlastních testů a úloh. Vlastní úlohu lze vytvořit jednou ze dvou možností a to buď jako soubor programového kódu vloženého do projektu prázdné aplikace (pro jednodušší úlohy) nebo jako samostatnou knihovnu k tomuto projektu přiřčenou (pro složitější úlohy).

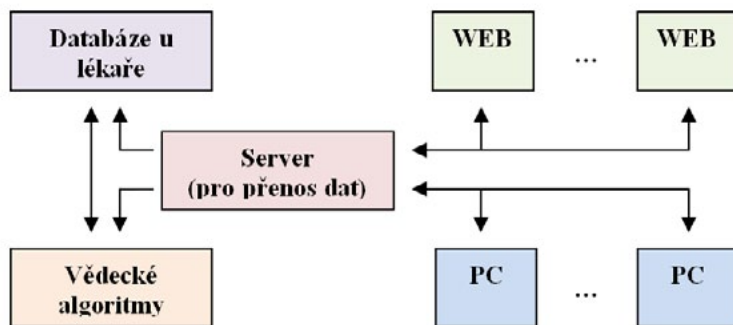
Pro skutečně široké použití je framework vytvořen současně ve dvou verzích a to jak pro desktopové, tak i webové použití. Vzhledem ke zvolené technologii bylo tohoto dosaženo celkem snadno a to v podstatě pouze podmíněnou kompilací některých programových částí (zejména počáteční konfigurace a ukládání nastavení) v rámci celého projektu a současně vytvořením velmi malé části, která je cílově závislá (synchronní/asynchronní zpracování). V některých směrech je samozřejmě desktopová verze frameworku snazší a výkonnější (přístup k databázi, využití i více jádrových procesorů).

Pokud tedy nejsou využity skutečně speciální schopnosti desktopové verze frameworku (nativní knihovny, paralelní zpracování dat), tak lze test nebo úlohu vytvořit pouze jednou a použít ji jak s desktopovou, tak i webovou verzí vytvořeného frameworku bez sebemenších úprav. V mnoha případech jsou testy a úlohy vytvářeny jako tzv. „plné“ a tzv. „omezené“. Jedná se v podstatě

o jeden a tentýž test nebo úlohu, která své schopnosti mění podle toho, v jaké verzi frameworku je umístěna. Desktopové aplikace většinou poskytují větší množství nastavení a tudíž i výstupů (pod dozorem lékaře) než webové aplikace (používané pouze pacientem).

Další nesporná výhoda ve vytvoření dvou verzí frameworku a tedy i testů a úloh (desktopové a webové) spočívá v jejich univerzálnějším využití. Při dosahu kvalitního internetového spojení je samozřejmě lepší využít webové verze (stále aktualizovaná verze úloh), ale například při rehabilitaci na chatě nebo bez připojení na internet lze využít desktopové verze úloh a výsledky odeslat teprve až při prvním připojení se na internet. Vzhledem ke vhodné zvolené technologii, navržené struktuře frameworku a jednotné tvorbě testů a úloh je této významné schopnosti dosaženo s minimálním úsilím, hlavně ale bez podpory tvůrců vlastních testů a úloh.

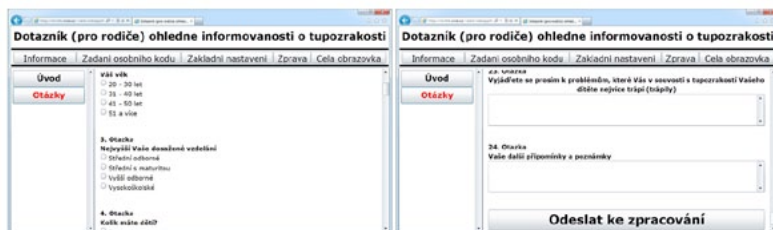
Nedílnou součástí celého systému je přenos dat (Obr. 2). Ta jsou z testů a úloh pomocí frameworku přenášena do dočasného úložiště na libovolném serveru (PHP script, webová služba, ...) v již zmíněném XML formátu. Zde mohou být vyzvednuty libovolnou jinou aplikací za účelem jejich zpracování (data jsou v podstatě anonymní) nebo vyzvednuty a přímo vloženy do databáze na pracovišti lékaře (dozorového personálu) pod skutečné jméno uživatele (pacienta). Jejich zpracování je tedy rozděleno na dvě části. Na jedné straně jsou převážně výzkumné aktivity využívající v podstatě pouze anonymní data doplněná identifikačním číslem pacienta (vyzvedávána nejčastěji ze serverového úložiště) a na straně druhé jsou převážně grafická znázornění pro potřeby lékaře již se skutečnými jmény pacientů (vyzvedávána z databáze na pracovišti lékaře). Z databáze lékaře lze samozřejmě anonymně data kdykoli později také vyzvednout (budou obsahovat pouze číselnou identifikaci pacienta) nebo naopak pouze podle identifikace uživatele nově vložit (výsledky výzkumných aktivit). Pro zamezení úniku osobních údajů nejsou data a informace o pacientech (rodná čísla, jména) uchovávány na externích serverech (například univerzitních), ale pouze v databázi na pracovišti lékaře.



Obrázek 1 — Princip spolupráce jednotlivých částí systému a cesty se směry přenosu dat

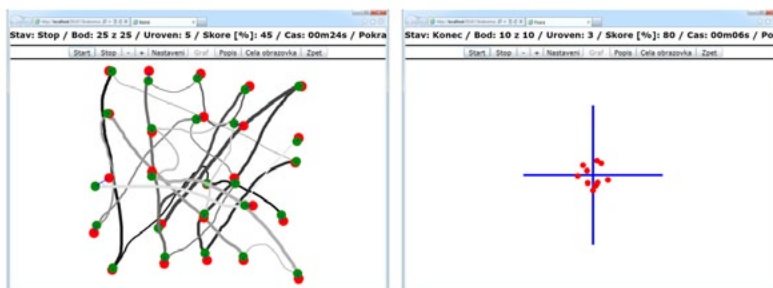
5. Příklady již realizovaných a navrhovaných testů a úloh

Pomocí popisovaného frameworku již byly vytvořeny některé základní testy a úlohy. Zde jsou uvedeny příklady s jejich popisem. Jako první lze uvést v podstatě běžný dotazník založený na textových otázkách (Obr. 3). Výsledek poskytovaný frameworkem neobsahuje pouze vypracování otázek uživatelem, ale rovněž celý záznam počínání uživatele (pohyby formulářem, časy zpracování položek a mnoho dalšího). Z tohoto lze hodnotit složitost otázek pro uživatele, nejistotu, opravy a částečně i aktuální psychický stav uživatele.



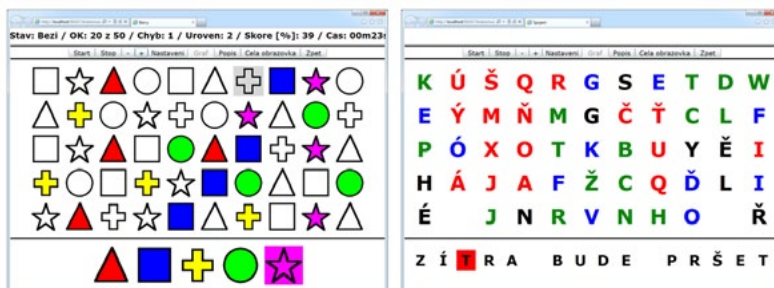
Obrázek 3 — Příklad základního dotazníkového formuláře ve webové aplikaci

Následující skupinu úloh tvoří testy zaměřené na přesnost vystavení bodu na cílovou polohu uživatelem, což je vstup úlohy (Obr. 4). Výstupem úlohy je nejen názorné grafické hodnocení úspěšnosti pacienta, ale je rovněž zachyceno jeho počínání například záznamem trasy myši (trasa od černé po stupních šedi až do bílé, tlustá značí pomalý pohyb a tenká naopak rychlý pohyb). Lze tedy detekovat v jakých směrech (částech zorného pole) pacient reaguje rychleji a naopak.



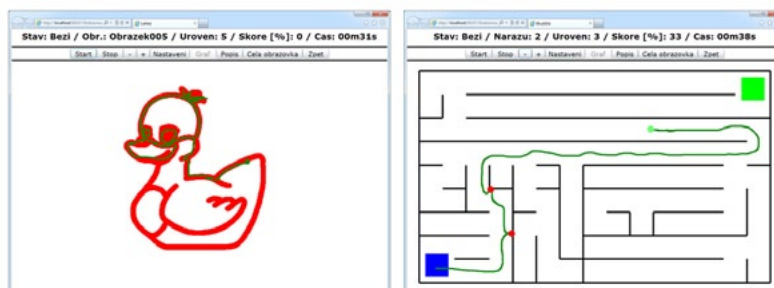
Obrázek 4 — Úloha na přesné umístění bodů, tedy lokalizace polohy

Další skupinou jsou úlohy na rozpoznávání tvarů nebo barev (Obr. 5). Úkolem uživatele (pacienta) je podle vzoru v dolní části označit tvar(y) v horní části. Výstupem úlohy je rovněž záznam celkové činnosti uživatele. Z něho lze usuzovat jaké tvary / barvy jsou pro uživatele problematické (chybné určení, časová prodleva). Tyto úlohy jsou vhodné na dlouhodobé posuzování vývoje pacienta jako je zlepšující se ostrost vidění nebo barvocitu.



Obrázek 5 — Úlohy na rozpoznávání nebo doplňování obrázků, vzorů nebo struktur

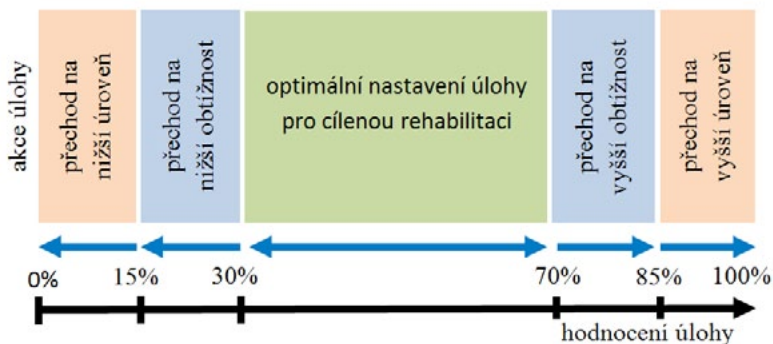
Poslední příkladovou skupinou jsou úlohy na pohyb ruky nebo-li přesněji řečeno na koordinaci oko – ruka (Obr. 6). Ty jsou založeny zejména na obkreslování různě složitých obrázků (klasifikovány podle typu čar: rovné, obloučky, vlnovky, ...) nebo dokonce současně schopnost odhadu vzdálenosti od zobrazených předmětů (průchod řešitele bludištěm a to středem cesty). Matematickým hodnocením přesnosti kresby (míra shody kresby pacienta se vzorem) nebo vzdáleností od zobrazených předmětů (skutečný střed cesty mezi zdmi bludiště) lze detekovat mnoho stavů pacienta od prosté nepozornosti, přes nedostatečný odhad až po příznaky parkinsonovy nemoci.



Obrázek 6 — Úlohy zaměřené na koordinaci oko-ruka využitelné v mnoha oblastech

Všechny uvedené úlohy (tedy kromě textového dotazníku) jsou schopny adaptovat se na aktuální stav pacienta [5]. Tato schopnost úlohy je velmi důležitá nejen pro vytvoření stále efektivní léčby dané poruchy, ale současně i pro vhodnou stimulaci pacienta. Pokud by byla úloha příliš lehká nebo naopak příliš složitá, tak nejen že by nebyla pro pacienta léčebná, ale přímo by jej odrazovala od svého řešení.

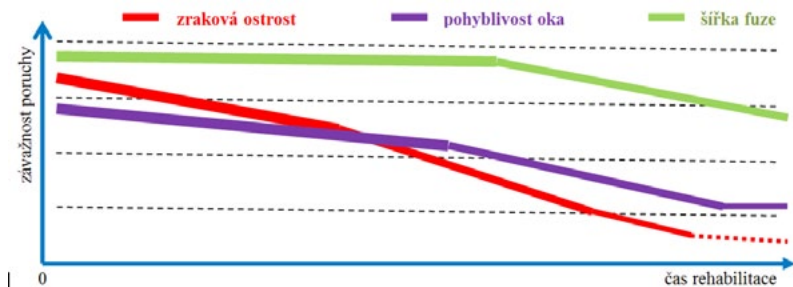
Po každém řešení úlohy pacientem je vypočteno tzv. skóre, což je číselná hodnota v rozsahu 0% (zcela neúspěšné řešení) až 100% (zcela úspěšné řešení) (Obr. 7). Podle této hodnoty je automaticky upravován stav, tedy obtížnost úlohy. Pro vysvětlení principu automatické adaptace úlohy je vhodná úloha na



Obrázek 7 — Princip automatické adaptace úlohy podle aktuálního hodnocení, tedy podle schopností uživatele

obkreslování obrázků, kde úkolem pacienta je obtáhnout perem předložený obrázek bez vybočení z čáry představující kresbu vzorového obrázku. Při poklesu úspěšnosti pacienta pod 30% se pouze zvyšuje šířka čáry předlohy, nikoli přímo snižuje obtížnost obrázku. Ale při poklesu hodnocení pod 15% již úloha použije jednodušší obrázek. Obdobná situace nastává při hodnocení na 70%, kdy úloha snižuje šířku čáry podkladové kresby, ale ještě nezvyšuje složitost obrázku. A při úspěšnosti nad 85% již úloha poskytne složitější obrázek.

Nedílnou součástí výstupu celého frameworku je rovněž vhodná prezentace výsledků (Obr. 8) [4]. Vzhledem ke skutečnosti, že jsou výstupy všech testů a úloh průběžně ukládány, tak lze toho snadno dosáhnout. Do výstupního grafu nejsou však zaneseny přímo výsledky jednotlivých testů a úloh (tedy jejich skóre), ale ty jsou v podstatě převedeny na hodnocení konkrétních poruch podle toho, jak jednotlivé úlohy stav konkrétních poruch odrážejí. Jednotlivé poruchy jsou pro názornost barevně odlišeny. Pro lékaře je však nejdůležitější posouzení celkového vývoje každé z poruch. Z tohoto důvodu je pro snadné čtení grafu úroveň poruchy indikována současně šířkou



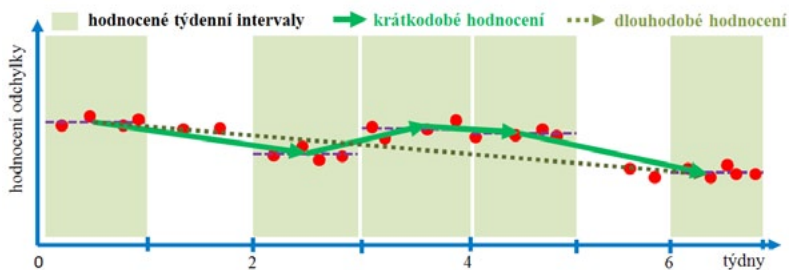
Obrázek 8 — Názorné grafické zobrazení celkového přehledu (nejen domácí) léčby pacienta

odpovídající čáry (menší šířka čáry značí menší úroveň poruchy) v grafu a rovněž dosažení uspokojivého stavu poruchy je indikováno přerušovanou čarou. Tento typ zobrazení je velmi názorný již na první pohled a není tedy nutno graf podrobněji a dlouhodobě studovat.

6. Prvotní přínos vytvořeného frameworku

Jelikož webová část frameworku je k dispozici zatím pouze několik měsíců, tak není možné vyvodit seriózní závěry. Přesto však již tato doba byla dostatečně dlouhá pro základní hodnocení a některé z toho vyplývající úpravy. Prvním a základním přínosem frameworku je informovanost lékaře o průběhu domácí léčby ve smyslu její pravidelnosti. Lze tedy zobrazit graf, převážně po týdnech, jak průběžně pacient domácí úlohy využíval. Další informace spočívá v celkovém hodnocení úloh v každém týdnu. Některé týdny však nemusí být hodnoceny, protože neobsahují dostatečné množství výsledků z dané úlohy od klienta. Z průměrného hodnocení výsledků jednotlivých týdnů je poté sestaven celkový přehled vývoje léčby pacienta (Obr. 9).

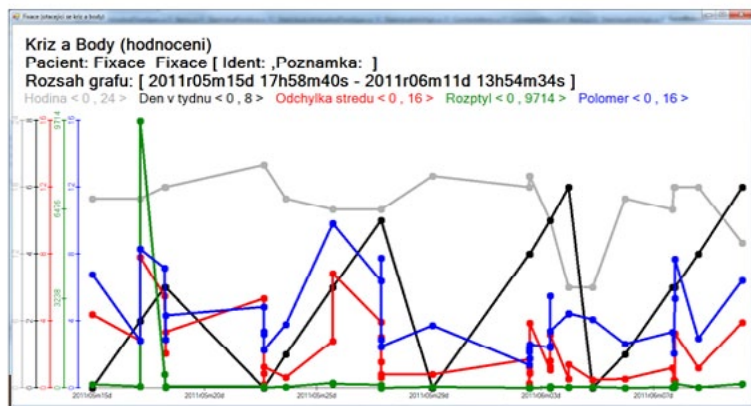
Prvním reálným výstupem projektu je hodnocení závislost výsledků některých úloh, tedy jejich nepřesnosti (červená, modrá, zelená barva – indikuje chybu)



Obrázek 9 — Princip hodnocení domácí léčby po časových intervalech (týdnech) s vyznačením celkového vývoje

na dni v týdnu (černá barva, nulová hodnota je pondělí) a hodině v průběhu dne (šedá barva). Z grafu (Obr. 10) jsou zřejmé nejhorší výsledky okolo středu týdne a naopak nejlepší v jeho přelomu. Výsledek kontroly u lékaře, bude tedy zcela jistě závislý na dni v týdnu (návštěva u lékaře), což není pro stanovení dalšího léčebného postupu lékařem zrovna vhodné.

Mezi dříve nikterak neuvažované schopnosti frameworku bylo nutno přidat jednu velmi podstatnou. Někteří pacienti považovali domácí rehabilitaci (cvičení) za zcela dostačující a již opomíjeli návštěvy u lékaře. Tato skutečnost je však velmi nežádoucí, neboť domácí úlohy jsou pouze doplňkem celkového léčebného procesu. Z tohoto důvodu bylo nutno přistoupit k časovému omezení činnosti webových aplikací a tím pacienty k návštěvě lékaře v podstatě „donutit“. Nejen celou aplikaci pro konkrétního pacienta, ale i jednotlivé jeho



Obrázek 10 — Příklad závislosti úspěšnosti řešení některých úloh pacientem na dni v týdnu a hodině

úlohy lze nyní povolit pouze na omezené časové období (pro jednoduchost nastavení ze strany lékaře na týden, dva týdny, měsíc, ...).

7. Závěr

Navržený a vytvořený framework je vhodným nástrojem pro snadnou a rychlou tvorbu mnoha typů nejen testovacích a tréninkových úloh, ale současně mnoha různých diagnostických nástrojů. Vzhledem k možnosti využít téměř libovolné externí zařízení od běžné WEB kamery až po speciální ovladače a to i pro webové aplikace se jeho pole působnosti značně rozšiřuje a tím překonává doposud realizované systémy v tomto oboru. Velkou výhodou je v podstatě jednoduše / přenositelnost vytvořených úloh mezi desktopovým a webovým využitím, což jej výrazně posouvá do popředí zájmu. Toto vše bylo možno dosáhnout díky využití moderní technologie jako je Microsoft .NET (WPF / SilverLight).

V současné době je u nás na katedře vytvořen soubor úloh pro domácí rehabilitaci zrakových poruch pacientů, který je využíván již na 5 lékařských pracovištích a to zhruba 100 pacienti [6]. Jak z pohledu lékařů, tak i pacientů se jedná o velký přínos po všech stránkách. Již prvotní výsledky z využívání tohoto systému poskytují velmi překvapivé výsledky a informace. Lékaři sami konstatují u některých pacientů znatelné zlepšení jejich stavu při v podstatě každodenním využívání úloh pro domácí léčbu. Naším cílem je dlouhodobější studie, která by prokázala skutečný přínos domácí léčby na zkrácení celkové doby léčebného procesu. V současné době se rovněž zahajuje projekt pro domácí rehabilitaci poruch pohybového aparátu pomocí speciálních senzorů. Dále se uvažuje, po doplnění o speciální algoritmy, o vytvoření univerzálnějších úloh jako je například obkreslování vhodně připravených obrázků, řešení matematických rovnic, psaní textů a hádanky za účelem využití tohoto

frameworku pro včasnou detekci mnoha typů poruch například dyslexie, problémy s barvocitem, Parkinsonova nemoc a mnoho dalších.

Poděkování

Práce byla podporována výzkumným záměrem č.MSM 680770012“ Transdisciplonární výzkum v oblasti biomedicínského inženýrství II“.

Rovněž je na místě poděkovat lékařům zejména v Praze FN Motole a Praze Barrandově za velmi přínosnou spolupráci při vytváření některých úloh a jejich hodnocení.

Literatura:

- [1.] *Manuály na návody na Microsoft .NET Framework (prog. jazyky, komponenty, technologie, ...)*
- [2.] Novák, P., Nováková, L., (2007). *The universal data storage primary for medical applications. Proceedings of Workshop [CD-ROM], 502–503. CTU, Prague. ISBN 978–80–01–03667–9*
- [3.] Novák, P., Nováková, L., (2007). *Univerzální úložiště dat převážně pro lékařské aplikace. Znalosti, Sborník příspěvků 6. Ročníku konference. 344–347, VŠB – Technical University of Ostrava, Ostrava. ISBN 978–80–248–1279–3*
- [4.] Nováková, L., Novák, P., (2006). *Tool for data visualization 3D. Proceedings of Workshop vol. B [CD-ROM], 123–126. CTU, Prague. ISBN 80–01–03439–9*
- [5.] Mařík, V., Štěpánková, O., Lažanský, J., a kolektiv (1993, 1995, 2007). *Umělá inteligence 1, 2, 5, ACADEMIA nakladatelství AV ČR, Praha. ISBN 80–200–0496–3, 80–200–0504–08, 80–200–0502–01*
- [6.] Novák, P., (2011) *Objektivizace a podpora pro diagnostiku a rehabilitaci strabismu, Disertační práce, ČVUT FEL, Praha*
- [7.] *Materiály pro výuku a postupy různých typů rehabilitačních oblastí (zrakové, pohybové, kognitivní atd.)*

Kontakt:

Petr Novák, Ing. Ph.D.

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra kybernetiky

Technické 2, 166 27 Praha 6

tel: 22435 5718

e-mail: novakpe@labe.felk.cvut.cz

Vladimír Eck, Doc. Ing. CSc.

eck@fel.cvut.cz

Olga Štěpánková, Prof. RNDr. CSc.

step@labe.felk.cvut.cz

<http://nit.felk.cvut.cz>, cyber.felk.cvut.cz