Acta Informatica Pragensia
2(1), 2013, 91–100, DOI: 10.18267/j.aip.16
Online: aip.vse.cz

Sekce / Section:
Recenzované stati / Peer-reviewed papers

Uplatnění systémového myšlení v analytické fázi vývoje aplikací

Anna Exnarová[1]

[1] Katedra systémové analýzy, Fakulta informatiky a statistiky,
Vysoká škola ekonomická v Praze
nám. W. Churchilla 4, 130 67 Praha 3

anna.exnarova@vse.cz

Abstrakt: Předkládaný článek představuje návrh uplatnění systémového myšlení v procesu vývoje aplikace. Uvedené modely systémového myšlení čtenáři představují možnosti jejich využití v procesu vývoje softwarové aplikace. Návrh nediktuje tvorbu konkrétního modelu v určitý okamžik, ale navrhuje jak je možno využít vybrané modely. Autorským záměrem je dosáhnout změny/rozšíření v mentálním modelu čtenáře – konkrétně části, která obsahuje obraz aplikace systémového myšlení při vývoji aplikací.

Klíčová slova: modelování, mentální modely, systémové myšlení, vývoj informačních systémů, softwarová aplikace

Title: Applying systems thinking in the analytical stage of application development

Abstract: This paper presents the proposal for application of systems thinking in the application development process. The models of systems thinking present to the reader the possibility of their use and application in the software application development process. The proposal does not mandate the creation of a specific model at some point, but suggests how the selected models can be used. Authorial intention is to change the mental model of the reader – particularly the part that contains an image of the application of systems thinking in developing applications.

Keywords: Modelling, Mental models, Systems thinking, Development of the information system, Software application

1 ÚVOD

Čím dále častěji se v pracovním životě v IT praxi při vývoji aplikací setkávám se situacemi, jejichž řešení běžným analytickým přístupem nepřináší žádoucí výsledky. I přesto, že neustále dochází k vývoji metodik, technik, nástrojů, přístupů v dílčích činnostech a procesech vývoje informačních systémů, dle dlouhodobých výzkumů se ukazuje, že výsledky nejsou uspokojivé. Vývoj informačních systémů a aplikací je poměrně drahý a jeho výsledky jsou nepředvídatelné. [10] Výsledná zpráva výzkumného projektu „CHAOS“ Standish Group z roku 2011 [10] dokladuje, že více jak polovina projektů vývoje informačních systémů jsou neúspěšné. Jedná se o projekty z let 2002 až 2010, mezi kterými pouze 37% bylo vyhodnoceno jako úspěšné. Bohužel 19% je zcela neúspěšných, přičemž z tohoto objemu je 50% projektů neúspěšných z toho důvodu, že nevyhovovaly byznysovým požadavkům. V 80% případů je konstatováno, že důvodem neúspěchu projektu byla realizace požadavků, které nebyly vhodně uchopeny a implementovány. S určitou mírou zjednodušení si dovolím konstatovat, že neúspěchy projektů velmi úzce souvisí se schopností úspěšně definovat požadavky a úspěšně a efektivně realizovat činnosti v analytické fázi vývoje.

Uchopení problematiky, pochopení zákazníka a jeho požadavků, vytvoření návrhu řešení je nepochybně podchyceno v běžně používaných přístupech. Pro to, aby tyto činnosti byly efektivní a poskytovaly očekávané kvalitní výsledky, je vhodné je propojit a současně aplikovat společně se systémovým myšlením. To umožňuje odstranit neschopnost pracovat se systémy jako s celky, pomáhá pochopit skutečnou podstatu problému a nalézt efektivní řešení. [9]

Následující článek předkládá návrh, jak je možné uplatnit systémové myšlení právě při vývoji informačního systému – konkrétně se jedná o uplatnění systémového myšlení v analytické fázi vývoje softwarové aplikace. Nástrojem pro toto uplatnění je použití modelů systémového myšlení.

2 SYSTÉMOVÉ MYŠLENÍ A VÝVOJ APLIKACÍ

Z pohledu *systémového myšlení* je chápání zkoumaného systému zakotveno ve způsobu komplexního myšlení nad prvky systému, včetně jejich vazeb v rámci vnitřních i vnějších souvislostí. Dle Mildeové [6] a [7] je základem paradigmatu „vhodnost postavení“ a schopnost myslet. Pod pojmem „vhodnost postavení“ je chápán postoj k problému, vnímání a chápání jak problému, tak jeho okolí. Efektivní pohled na problém je vyhledáním jakési rovnováhy vzdálenosti, tj. ani ne zblízka příliš detailní, ani ne z velké dálky, tak aby nebyly zanedbané podstatné skutečnosti. To co vnímáme, je závislé na našem způsobu nahlížení. Pozornost by měla být věnována nejen struktuře, ale také vzorům chování a jeho důsledkům.

Použitím systémového myšlení pro řešení problémů lze dosáhnout přiměřeného nalezení příčin a jejich následků, akcentování existence zpětněvazebných smyček mezi prvky v systému a jejich dopadů na chování systému a integrace časového hlediska do popisu systému i jeho chování. Systémové myšlení je v souladu se základními principy definovanými systémovými přístupy, kterými jsou holismus, emergence a synergický efekt; systémová hierarchie; existence zpětné vazby; komunikace. [7]

Nutno podotknout, že při vývoji aplikace je tvůrce konfrontován se dvěma systémy: (které se ale mohou výrazně prolínat, či dokonce překrývat)

- za prvé zkoumá systém, nad kterým je aplikace budována;
- a za druhé vlastní aplikace je novým systémem, který tvůrce vytváří.

Jako příklad uveďme aplikaci, která bude evidovat docházku zaměstnanců ve firmě (tento příklad je používán i v dalším textu článku). Zkoumaný systém vykazuje vlastnosti, které řešitel při návrhu aplikace musí akcentovat (např. to, zda v realitě zaměstnanci musí být fyzicky přítomni na pracovišti, nebo mohou pracovat z domu či u zákazníka, zda je možné, aby jeden den pracovali pouze 3 hodiny a druhý den si chybějící čas tzv. napracovali, zda je možné, aby fyzicky v objektu existovala čtecí zařízení evidující příchody a odchody zaměstnanců).

Vytvářený systém – tedy nová aplikace, je rovněž samostatným systémem. Způsobů, jakým tento systém může být navržen, vytvořen a následně s ním bude pracováno, je také mnoho. A rovněž i zde je možné využít principů systémového myšlení.

2.1 PRINCIPY SYSTÉMOVÉHO MYŠLENÍ

Při zkoumání vývoje aplikací se objevují místa, která přímo ukazují na možnost aplikace systémového myšlení. Postupně uveďme základní principy a pokusme se okomentovat jejich význam z pohledu vývoje aplikace.

Princip *holismu* a *emergence* hovoří o tom, že na systém se nelze dívat odděleně na základě pouhé dekompozice systému. Je potřebné sledovat jeho chování a to nejen chování izolovaných prvků, ale v souvislostech s ostatními prvky a okolím systému, pozorovat jejich vazby a vnější i vnitřní interakce. V případě aplikace, která je nasazována do informačního systému, jsou tyto vlastnosti akcentovány např. v integračních testech. Aplikace může být otestována uživatelsky, ale může mít při nasazování velké problémy, právě z důvodu interakce s ostatními komponentami informačního systému.

Příklad vzniku *synergie* při vývoji systému uvádí [1, str. 45], ve kterém autoři Bébr a Doucek uvádí, že pokud řešitel na začátku definuje požadované funkce na elementární úrovni, a je vynakládáno poměrně velké úsilí na sladění funkcí a pochopení systému. V určité fázi nastává zlomový okamžik, kdy systém náhle začne snadno a dobře plnit požadované funkce.

Problematiku *systémové hierarchie* určitě zná každý analytik. Systémová hierarchie zkoumá systém jako součást systému vyšší úrovně a naopak považuje každou jeho část za autonomní systém, který ale na rozdíl od tradičního pojetí rozpadu pouze na části bere v potaz to, že se jedná o systémy, které mají vazby na okolí, a vazby uvnitř systému. Vazby i struktura systému teprve společně určují chování, vlastnosti a funkce systému. V praxi se často lze setkat se snahou rozpracovat složitou problematiku zkoumaného systému na menší uchopitelné části, které ale jsou dále již zkoumány izolovaně. Typickým příkladem jsou procesní modely a jejich dekompozice. Vytvoření procesního modelu složitého zkoumaného systému vyžaduje velkou míru znalostí, dovedností, ale také zkušeností, které analytik musí mít, aby správně dokázal vytvořit dekompozici systému, „nepřestřelil“ míru podrobnosti, dokázal akcentovat všechny podstatné a potřebné vlastnosti systému. Mnoho začínajících analytiků se „utopilo“ při snaze o vytvoření dokonalého procesního modelu, který byl detailní a dopodrobna modeloval dílčí kroky procesu, ale pro vyšší úrovně pohledu byl nepoužitelný.

Regulace (příp. samoregulace) je jednou ze základních vlastností systému. Čím vyšší komplexitu systém vykazuje, tím bohatší je také struktura zpětných vazeb. Při návrhu vyvíjené aplikace (vyvíjeného systému) jsou běžně zabudovávány principy regulace – např. na úrovni databázových operací jsou řešeny optimalizace a regulování výkonu. Z pohledu zkoumaného systému je existence zpětných vazeb tím složitější, čím složitější je systém. Se zpětnými vazbami velmi úzce souvisí schopnost prozkoumání příčin a následků a akcentování dynamiky změn.

Komunikace systémů je vlastně výměnou informací (dat) mezi prvky systému nebo jeho okolím. Pro úplný popis systému je nutné akcentovat všechny typy komunikace, které systém vykazuje. Ve výše uvedeném příkladu vývoje aplikace pro docházkový systém je z pohledu vyvíjeného systému nutné uvažovat např. o formě, jakou jsou schvalovány práce mimo pracoviště, principy schvalování odpracovaných hodin, nebo informační toky směrem od zaměstnance k personálnímu či mzdovému oddělení. Z pohledu vyvíjeného systému jde např. o to, zda komunikace, které probíhají v realitě, bude nově vytvářený systém plně akcentovat, nebo dojde k jejich úpravě, či v případě technické implementace jakým způsobem budou komunikovat databázové objekty s aplikační vrstvou.

3 MODELY SYSTÉMOVÉHO MYŠLENÍ

Systémové myšlení poskytuje nástroje, které umožňují či usnadňují jeho používání. Jedním z těchto nástrojů jsou jeho modely. V tomto článku se věnuji těmto typům modelů, které lze chápat jako *modely systémového myšlení*:

- mentální modely,
- myšlenkové mapy,
- diagram příčin a následků,
- příčinně-smyčkový diagram,
- diagram hladin a toků.

K tomuto výčtu uvádím dvě stručné poznámky.

1) První uvedený typ modelů (mentální modely) chápu jako nutnou součást modelů systémového myšlení, protože bez nich nelze provádět jakoukoliv kognitivní činnost. Pokud člověk ovládá schopnost systémového myšlení, jeho mentální modely vykazují rovněž znaky systémového myšlení. Na druhou stranu, pokud je nevykazují, jsou tyto modely nutnou podmínkou pro vznik dalších modelů. Tudíž se domnívám, že tento typ modelů náleží do modelů systémového myšlení.

2) Pojem model a diagram. V článku chápu model jako zjednodušenou reprezentaci, obraz reálného systému. Diagram je vizualizací modelu.

Následující pasáž článku představuje základní charakteristiky jednotlivých typů modelů systémového myšlení.

3.1 CHARAKTERISTIKY MODELŮ SYSTÉMOVÉHO MYŠLENÍ

Mentální model si vytváříme v našich hlavách a potřebujeme jej k tomu, abychom byli schopni v realitě existovat. Zahrnuje v sobě ty vlastnosti systému, které považujeme za potřebné. Nedostatky, které mentální modely vykazují, souvisí s jejich vlastnostmi. Jsou nekompletní, limitované schopnostmi lidí „řídit“ a pracovat s vlastními mentálními modely, jsou nestabilní a v čase proměnlivé, nemají jasně definované hranice a vazby, nejsou vědecké, často obsahují „nevysvětlitelné“ charakteristiky a emocionální prvky, z části jsou nevědomé a jsou velmi obtížně sdílené. [3] [5] [8]

Myšlenkové mapy (mentální mapy, mind mapy) umožňují vizualizovat vybranou část mentálních modelů, umožňují vizualizovat a dokumentovat předávané informace. Myšlenková mapa je grafické znázornění pojmů, jejich vazeb a souvislostí, doplněné např. grafikou, obrázky, ikonami, barevným odlišením. Tyto modely jsou založeny na jednom z principů přemýšlení – lidé nemyslí čistě lineárně, ale vytváří si mapy myšlenek, které různě rozvíjí a propojují. [2] Mentální modely jsou určujícím faktorem pro porozumění. Pokud chce jedinec řešit komplexní úlohy úspěšně, musí aktivním kognitivním procesem trvale pracovat na korekci svých mentálních modelů i paradigmat.[8

Dnes již klasickým prostředkem pro vizualizaci příčin a následků je tzv. *diagram příčin a následků*, neboli diagram rybí páteř, diagram rybí kosti, fishbone diagram, někdy také označován jako Ishikawův diagram. Diagram slouží jako prostředek analýzy příčin a následků, v některých případech bývá používán obdobně jako myšlenkové mapy pro sumarizaci souvisejících prvků a jejich vazeb.

Hledání příčin a následků je v západním způsobu myšlení velmi pevně zakořeněný přístup k řešení problémů. Ať již v soukromém, profesním nebo vědeckém životě – všude jsme obklopeni výčtem příčin a důvodů, které mají za následek nějaké chování nebo stav. A většinou se snažíme se dopátrat úplného výčtu – čím víc tím líp. Jedná se přitom o hledání jednoduchých kauzálních vztahů, jednosměrného a bez jakéhokoliv vymezení vzájemného působení mezi příčinami a důsledky. Také zde neexistuje akcent na zpětné působení od následku k příčině. Při modelování problémové situace je absolutně zanedbána skutečnost, že důsledek může mít velký vliv i na původní příčiny. [4]

Příčinně smyčkové diagramy (causal loop diagram, CLD) umožňují modelovat příčinnosti a tendence vývoje. Diagram kauzálních smyček je nástrojem pro vizualizaci a následnou analýzu vzájemných vlivů proměnných v modelech systémové dynamiky. Vzájemné vztahy proměnných usnadňují pochopení a kvantifikaci funkcí komplexních systémů.

Diagram hladin a toků (Stock and flow diagrams, SFD) slouží k vyjádření vývoje proměnných v čase. Základními stavebními kameny tohoto diagramu jsou: hladiny (znázorňují stavy – prvky systému, charakterizující stav systému, zároveň reprezentují i zpoždění); a přítoky a odtoky s ventily (vyvolávají změny hladin dané nastavením parametrů).

Poslední dva typy (příčinně smyčkové diagramy a diagramy hladin a toků) bývají často uváděn y jako modely systémové dynamiky. Systémové myšlení a systémovou dynamiku chápu jako dvě samostatné disciplíny, které mají vzájemný přesah. V případě, že modely obsahují i simulační prvky, náleží do disciplíny systémová dynamika.

3.2 MODELY SYSTÉMOVÉHO MYŠLENÍ V ANALYTICKÉ FÁZI VÝVOJE APLIKACE

Pro výše uvedené typy modelů systémového myšlení uvádí následující kapitola stručnou charakteristiku ve vztahu k vývoji aplikací – a to jak z pohledu vyvíjeného systému, tak z pohledu zkoumaného systému.

3.2.1 MENTÁLNÍ MODELY

Prvním modelem, který v procesu vývoje aplikace vzniká, je *mentální model*. Vzniká nejdříve v myslích zadavatelů, kteří tvoří zadání a poptávají možnosti pro řešení jejich problému. Rovněž kognitivním procesem vznikají modely realizátorů, kteří zadání studují, následně vytváří analýzy, navrhují řešení, implementují řešení, testují či vytvořenou aplikaci nasazují.

Kvalita těchto modelů je klíčová pro celý proces vývoje aplikace. Je klíčová pro zabezpečení komunikace. Nedostatky mentálních modelů jsou často omezujícím prvkem pro dosažení požadovaných výsledků. Mentální modely jsou používány napříč celým procesem vývoje aplikace.

V modelovém příkladu je klíčovou osobou analytik, který má vytvořit základní koncepci návrhu aplikace pro docházku. Tento analytik dosud celý život pracoval na pozici zaměstnance, vždy docházel pravidelně na pracoviště a nikdy se nesetkal se situací, kdy pracuje dopoledne z domu a odpoledne vyjíždí k zákazníkovi. Jeho mentální model budoucí aplikace obsahuje na začátku pouze jednu variantu – v aplikaci postačuje evidovat příchod a odchod na/z pracoviště. Změna této části mentálního modelu vyžaduje detailní znalost zkoumaného systému. Zároveň ale také schopnost a ochotu analytika měnit svůj mentální model na základě nových poznatků. Pouhým zjištěním, že realita se odlišuje od mentálního modelu, je často nedostatečná. Lidský mozek musí tuto informaci dostatečně zpracovat, tak aby byl schopen svůj mentální model změnit.

3.2.2 MYŠLENKOVÉ MAPY

Myšlenkové mapy jsou modely, které pomáhají jejich tvůrcům, ale také uživatelům, lépe „přemýšlet“ – tj. efektivněji pracovat se svými mentálními modely a lépe je upravovat. Z pohledu analytické fáze vývoje aplikace tento typ modelu umožňuje:

- popsat potřeby, které má aplikace řešit;
- deklarovat důvody vzniku aplikace;
- v případě složitějších interakcí dokumentuje očekávané potřeby uživatelů;
- popisuje chování aplikace (vytvářeného systému);
- mapuje dotčené komponenty informačního systému, či okolí aplikace;
- nastiňuje možné vazby na okolí.

Tento typ modelu by neměl pouze dokumentovat, ale také by měl ověřit, zda to, když „někdo říká, že něco potřebuje“, je skutečnou potřebou. Zda existující mentální modely jsou založeny na reálné potřebě, anebo jsou z nejrůznějších důvodů modifikovány a dopady těchto modifikací jsou nežádoucí. Rovněž by měly pomoci ověřit, že mentální modely – a z nich vzniklé zadání, či popis budoucí aplikace, jsou úplné.

Z těchto modelů může plynout potřeba redefinice nebo doplnění výčtu požadavků na aplikaci, úprava zadání. Tvořením a následnou diskusí lze ověřit, zda je realizován požadovaný systém. Přínosy existence těchto modelů jsou (či mohou být):

- revize požadavku na vytvoření systému,
- revize rozsahu projektu a cíle,
- revize (doplnění nebo zrušení) seznamu požadavků,
- opětovná diskuse se zadavatelem o cílech a očekáváních,
- zjištění potřeby externích konzultací o okolí systému, o kterém nemáme dostatek informací,
- vytipování parametrů, které vyplývají ze specifikace požadavků, a které budou použité pro tvorbu případů užití,
- identifikace bodů, které je případně nutné zpracovat pomocí vybrané notace procesního modelování.

3.2.3 DIAGRAM PŘÍČIN A NÁSLEDKŮ

Diagram rybí páteře umožňuje popsat příčiny a následky chování systémů či možné stavy při řešení konkrétního problému. V kontextu vývoje aplikace v analytické fázi by tento model měl obsahovat popis příčin problémů, které by měl vyvíjený SW řešit. Při tvorbě tohoto typu diagramu je velmi důležitým prvkem ujasnění skutečných příčin a zmapování jejich možných následků.

Použitím tohoto typu modelu může vyplynout nutnost dalších změn mimo plánovaný vývoj aplikace. Příkladem může být zjištění, že aplikace evidující docházku zaměstnanců nemůže mít nastaven limit 8 odpracovaných hodin za den. Důvodem je existence přesčasů, které zaměstnanci vykazují. Příčinnou přesčasů jsou různé situace, jejichž zmapování umožní v aplikaci jejich evidenci (v případě, že tento požadavek se ukáže jako relevantní, bude následně možné vytvořit např. statistiku, proč jsou práce přesčas využívány). Přínosy existence tohoto typu modelu mohou být:

- revize požadavku na vytvoření systému,
- revize rozsahu projektu a cíle,
- opětovná diskuse se zadavatelem o cíli a očekáváních.

3.2.4 PŘÍČINNĚ-SMYČKOVÝ DIAGRAM

Pohled na entity, veličiny, parametry, či části systému a jejich vzájemné vztahy, vazby a procesy umožňuje modelovat *příčinně-smyčkový diagram*. Tento typ diagramu umožňuje:

- identifikovat a modelovat klíčové parametry, které by systém měl ovlivňovat a naplňovat,
- identifikovat posilující a balanční smyčky, které popisují principy chování systému a umožňují tak snáze najít efektivní způsoby řešení problémů,
- modelovat vazby systému na okolí,
- podporovat vyhledávání skrytých, nezjevných, nebo nepřímých interakcí,
- v případě několika diagramů z různých úrovní pohledu vybudovat odpovídající model hierarchie dílčích částí systémů,

- zachytit vlastnosti, vnitřní závislosti, systémovou hierarchii, procesy uvnitř systému i celé organizace.

Klasické modelování entit často ohraničuje pozornost pouze na vnitřní vazby systému. Příčinně-smyčkový diagram podporuje modelování a přemýšlení o vazbách systému na okolí a pomáhá přemýšlet o procesech a vztazích v příčinných i zpětně-vazebních smyčkách.

V modelovém příkladu je díky tomuto diagramu např. možno zjistit požadavky na reporty, které aplikace bude poskytovat. Ty se odvíjí od uživatelů. Určitě mezi standardními uživateli bude personální oddělení, nadřízení sledovaných pracovníků a vedení firmy, ale také např. státní instituce a dohledové orgány.

Přínosy existence tohoto diagramu jsou:

- revize požadavku na vytvoření systému,
- revize rozsahu projektu,
- revize (doplnění nebo zrušení) seznamu požadavků,
- opětovná diskuse se zadavatelem o cílech a očekáváních,
- identifikace entit, které následně budou vystupovat v logickém modelu (příp. jejich verifikace, pokud je identifikace provedena na základě seznamu požadavků),
- nalezení kritických procesních bodů, které je vhodné modelovat pomocí specializované notace procesního modelování.

3.2.5 DIAGRAM HLADIN A TOKŮ

Detailnější prozkoumání entit, jejich vazeb z pohledu systémového myšlení je možné pomocí *diagramu hladin a toků*. V předešlých krocích se postupně definovaly entity, vytipovaly se vazby mezi entitami i mezi entitami a okolím systému, byly identifikovány příčiny a následně popsány zpětné i příčinné vazby. Následuje diagram hladin a toků, který umožňuje:

- podrobněji popsat a specifikovat klíčové parametry, které by systém měl ovlivňovat a naplňovat (některé z těchto parametrů byly identifikovány v příčinně-smyčkovém diagramu),
- podrobněji dokumentovat vazby, které v systému působí,
- konkretizovat působení balančních a posilujících smyček ve smyslu identifikace parametrů, pomocných proměnných a konstant, které v těchto procesech působí,
- vytvoření simulací, na základě kterých je možné podrobně specifikovat chování systému za konkrétních podmínek,
- identifikovat stavy systému, potažmo vybraných entit.

Příkladem pro tento typ diagramu je např. modelování stavu zaměstnance z pohledu vykázaných hodin – nevykazuje, má vykázáno, má uzavřen pracovní týden, atd. Pochopení těchto stavů a parametrů, které tyto stavy ovlivňují, jsou klíčové pro správné nastavení např. stavových modelů a použití stavů při implementaci.

Přínosy existence tohoto diagramu jsou:

- identifikace entit, které následně budou vystupovat v logickém modelu (příp. jejich verifikace, pokud je identifikace provedena na základě seznamu požadavků),
- identifikace klíčového parametru – tak může dojít k nutné revizi požadavků a cílů,
- vliv na identifikaci kritických faktorů,
- identifikace parametrů a jejich vliv na entity systému,
- simulací ověřené a potvrzené očekávané hodnoty, identifikace potřebných úprav návrhu a designu systému, nebo změna způsobu implementace.

4 ZÁVĚR

Z výše uvedených dílčích přínosů použití jednotlivých typů modelů systémového myšlení je možné generalizovat globální přínosy, které uplatnění systémového myšlení v analytické fázi vývoje aplikace má.

Systémové myšlení poskytuje nástroje na to, jak pracovat s komplexními systémy a řešit jejich problémy. Vývoj informačních systémů či dílčích aplikací je nepochybně disciplínou komplikovanou a komplexní. Využitím principů systémového myšlení za pomoci použití jejich nástrojů (tj. modelů), by bylo možné docílit efektivnějších a uspokojivějších výsledků.

Uvedené modely jsou rozšířením oproti stávajícím a běžně používaným modelům při vývoji aplikací (např. UML nebo procesní modely). Jejich souběžné použití by mohlo vykazovat významné synergické efekty. Domnívám se, že aplikací systémového myšlení do vývoje informačních systémů a spojením tradičních modelů s modely systémového myšlení vzniknou významné přínosy.

5 SEZNAM POUŽITÝCH ZDROJŮ

[1] BÉBR, R., DOUCEK, P., 2005. *Informační systémy pro podporu manažerské práce*. 1. vyd. Praha: Professional Publishing, 2005, 223 s. ISBN 80-864-1979-7.

[2] BUZAN, T., B., 2011. *Myšlenkové mapy: probuďte svou kreativitu, zlepšete svou paměť, změňte svůj život*. Vyd. 1. Brno: Computer Press, 2011, 213 s. ISBN 978-80-251-2910-4.

[3] COSTELLO, W., 2007: *Computer_Based Simulations as Learning Tools: Changing Student Mental Models of Real-Word Dynamical Systems; Creative Learning Exchange*. [online]. 2001. [cit. 2013-03-10]. Dostupné z: http://www.clexchange.org/ftp/documents/Implementation/IM2001-04ChangingStuMentMod.pdf

[4] EXNAROVÁ, A. *Uplatnění systémového myšlení v analytické fázi vývoje softwaru*. Disertační práce, VŠE FIS, Praha, 2013.

[5] CHERMACK, T. J., 2003. Mental Models in Decision Making and Imlications for Human Resource Development; SAGE Publications, Advances in *Developing Human Resources*, Vol. 5, No. 4, 408-422, [online]. 2003. [cit. 2013-03-12]. Dostupné z : http://adh.sagepub.com/cgi/content/abstract/5/4/408

[6] MILDEOVÁ, S., DALIHOD, M., EXNAROVÁ, A, 2012. Mental Shift Towards Systems Thinking Skills in Computer Science. Journal on Efficiency and Responsibility in *Education*

and Science [online], 2012, roč. 5, č. 1, s. 25–35. ISSN 1803-1617. Dostupné z: http://www.eriesjournal.com/_papers/article_165.pdf

[7] MILDEOVÁ, S., a kol., 2007. *Tvorba manažerských simulátorů: vaše virtuální firma.* Oeconomica, Praha, 2007. ISBN 978-80-245-1286-0.

[8] MOLNÁR, Z., MILDEOVÁ, S., ŘEZANKOVÁ, H., BRIXÍ, R. KALINA, J., 2012. *Pokročilé metody vědecké práce*. 1. vyd. Praha: Profess Consulting s.r.o., 2012. ISBN 978-80-7259-064-3.

[9] SENGE, P. M., 1995. Piata disciplína manažmentu, Systémové myslenie predpoklad rozvoja organizácií, Open Windows, Bratislava, 1995, ISBN 80-85741-10-5.

[10] SCHWABER, K., SUTHERLAND, J.V. The Crisis in Software: The Wrong Process Produces the Wrong Results. *Software in 30 days: how agile managers beat the odds, delight their customers, and leave competitors in the dust*. Hoboken, N.J.: John Wiley, 2012, s. 14. ISBN 978-1-118-20666-9.