

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

IS – APLIKACE PRO SERVISNÍ MANAGEMENT

IS - SERVICE MANAGEMENT APPLICATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. PAVEL LÁTAL

VEDOUCÍ PRÁCE
SUPERVISOR

ING. RADOMIL MATOUŠEK, PH.D.

BRNO 2010

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2009/2010

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Pavel Látal

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Aplikovaná informatika a řízení (3902T001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

IS – aplikace pro servisní management

v anglickém jazyce:

IS - Service management application

Stručná charakteristika problematiky úkolu:

Daná DP bude využívat nabyté inženýrské vědomosti nejen z oblasti aplikované informatiky, a to pro konkrétní potřebu servisního managementu ve výrobním závodě Lear corporation, s.r.o. Účelem aplikace bude management při zpracování evidence poruch, servisních zásahů a řízení lidských zdrojů, s následným vyhodnocením rentability na základě sledovaných charakteristik a vypočtených matematických statistik daných procesů. Aplikace bude realizována pomocí MS C#, volba databáze je podmíněna pouze funkčností, pro řešení jsou vyžadovány relace (např. MySQL, MS SQL, Oracle).

Cíle diplomové práce:

- Provést systematický rozbor potřeb zákazníka a zpracovat teoretickou analýzu daného problému. Na tomto základě algoritmizovat návrh řešení (např. pomocí UML).
- Vytvořit aplikaci IS dle charakteristiky zadání, resp. dle firemních požadavků. IS musí, krom jiného, zahrnovat: klientskou aplikaci, databázový datový sklad a statistiku sledovaných veličin (ekonomika, čas, lidské zdroje).
- Vytvořit adekvátní elektronickou dokumentaci k vytvořené aplikaci.

Seznam odborné literatury:

[1] Jon Holt, UML for Systems Engineering: Watching the Wheels IET, 2004, ISBN 0863413544.

Vedoucí diplomové práce: Ing. Radomil Matoušek, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2009/2010.

V Brně, dne

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

Abstrakt

Práce pojednává o elektronickém informačním systému pro evidenci a sledování servisních zásahů, prováděných na lisovacích a stříhacích nástrojích, na vstříkovacích formách a na montážních strojích a montážních linkách ve výrobním závodě společnosti LEAR Corporation Czech Republic, s. r. o. ve Vyškově.

Abstract

This diploma thesis deals with electronic information system for recording and tracking service calls on pressing and cutting tools, the injection molds and assembly machines and assembly lines in manufacturing plant of Lear Corporation Czech Republic, s. r. o. in Vyškov.

Klíčová slova

Informační systém, Databáze, UML, Myšlenková mapa, Lear, Projektová příprava, Microsoft SQL server

Keywords

Information system, Database, UML, Mind map, Lear, Project preparation, Microsoft SQL server

Obsah

1	Úvod	11
1.1	Popis firmy Lear Corporation Czech Republic, s. r. o.....	12
1.2	Informační systém	12
1.3	Historie vývoje a implementace projektu	13
2	Projektová příprava.....	17
2.1	Základní požadavky zadavatele	17
2.2	Definování řešeného problému	18
2.2.1	Definice řešeného problému	18
2.3	Základní rozbor problému.....	18
2.4	Analýza procesu	19
2.4.1	Analýza procesu servisních zásahů.....	19
2.4.2	Popis procesu	20
3	Myšlenková mapa	21
4	UML.....	23
4.1	Historie UML.....	24
4.2	Diagramy UML 2.2	26
4.2.1	Strukturní diagramy	26
4.2.2	Diagramy chování	26
4.3	UML Diagramy použité při projektové přípravě	27
4.3.1	Stavový diagram.....	27
4.3.2	Diagram aktivit	29
4.3.3	Diagram případů užití.....	33
4.3.4	Struktura aplikace	35
5	Databázový server	37
5.1	Historie Microsoft SQL serveru	37
5.2	Datový model.....	38
6	Klientská aplikace	41
6.1	Systém LVIS.....	41
6.2	Správa uživatel.....	42
6.3	Hlavní okno aplikace Opravy nástrojů 2.0	43
6.4	Zadání nového servisního požadavku	44
6.5	Správa oprav	44
6.6	Náhled opravy.....	45
6.7	Správa strojů.....	47
6.8	Správa nástrojů.....	48
6.9	Správa typů závad.....	48
6.10	Správa uživatelských oprávnění	49
6.11	Přehled uzavřených oprav	50
6.12	Přehled nejopravovanějších nástrojů a strojů	51
6.13	Reporty	53

7	Závěr.....	55
8	Seznam použité literatury	57
	Seznam příloh	61

1 Úvod

Hlavním cílem této diplomové práce bylo vytvoření elektronického informačního systému (IS) pro evidenci a sledování servisních zásahů prováděných na lisovacích a stříhacích nástrojích, na vstřikovacích formách a na montážních strojích a montážních linkách.

V teoretické části práce je popsán celý proces vývoje projektu od počáteční analýzy rozsahu problematiky, identifikace požadavků, algoritmizace procesů, volba vývojových a realizačních prostředků, postupný vývoj a implementace aplikace, testovací provoz až pro plné nasazení do výroby. Jsou zde popsány aplikované metody práce se zadavatelem, sloužící k identifikaci všech významných i méně důležitých procesů spojených s životním cyklem servisního zásahu. Dále práce obsahuje popis jednotlivých prvků klientské aplikace a také rozbor databáze a datového modelu informačního systému.

Praktickou část tvoří vlastní informační systém pro servisní management. Tento je tvořen databázovým serverem, který slouží jako centrální datový sklad. Druhým prvkem, který tvoří významnější část této práce je klientská aplikace v pozici prostředníka mezi uživatelem a databází.

IS je vyvíjen pro výrobní závod společnosti LEAR Corporation Czech Republic, s. r. o. ve Vyškově. Závod je rozdělen na několik oddělení, takzvaných středisek, přičemž tři základní, výrobní střediska jsou ve firemním názvosloví pojmenovány *Plechý, Plasty, Montáž*. Další, pro IS důležitá střediska jsou *Nástrojárna* a *Strojárna*, což jsou oddělení, které výrobním střediskům zajišťují potřebný servis v podobě provádění oprav, přestaveb a výroby náhradních dílů.

Na všech nástrojích a strojích probíhají tři základní skupiny akcí, které jsou v agendě servisního managementu zaznamenávány. První z nich jsou pravidelné údržby, tedy podle potřeb jsou nástroje a stroje čištěny, ošetřovány, případně jsou vyměňovány některé součásti. Druhou skupinou jsou plánované přestavby. Velká část, ne-li většina nástrojů a samozřejmě hlavně strojů, jsou víceúčelové, tedy lze na nich vyrábět až několik více či méně odlišných výrobků, ve firemním názvosloví, artiklů. Třetí a nejzávažnější skupinou servisních zásahů jsou poruchové servisní zásahy. To jsou takové, kdy v průběhu výroby dojde k nečekané situaci a například vstřikovací forma začne vyrábět zmetky.

Původní systém evidence oprav, vyplývající z historického vývoje firmy, byl veden v papírové podobě, kdy pro každou opravu byl nejprve vystaven požadavek na opravu, v podobě papírového formuláře, obsahujícího základní informace obvykle vyplněné na počítači v programu Excel, přičemž po vytištění nebyl již dále v elektronické formě uchováván. Poté spolu s nástrojem, či strojem procházel celým procesem servisu. Popis provedených akcí byl do formuláře doplňován ručně. Po dokončení všech procesů byl formulář uložen do pořadače. Příklad tohoto formuláře v podobě před zavedením elektronické evidence viz příloha číslo 1.

Vzhledem k faktu, že měsíčně je provedeno průměrně kolem šesti set evidovaných, servisních zásahů, pak takovýto „papírový“ systém se stává pouhou evidencí, bez efektivní zpětné vazby. Lze zde pouze dohledat, například v důsledku zákaznické reklamace, kdy byl na jakém nástroji proveden jaký zásah, ale i to, vzhledem k neuspořádanosti agendy, jen velmi obtížně. Jakýkoliv širší náhled na celou problematiku servisu nástrojů a strojů znamenal náročnou, „mravenčí“ práci

při procházení velkého množství obsáhlých šanonů s mnohdy obtížně čitelnými, ručně psanými poznámkami. Ne zřídka se také stalo, že během servisu byl patřičný formulář znečištěn, poškozen, či dokonce ztracen. Navíc každé středisko mělo své návyky ve vedení této agendy, což činilo celou situaci do značné míry ještě více nepřehlednou.

Toto vše bylo impulzem pro vedení výroby na vytvoření jednotné elektronické evidence celé této agendy.

1.1 Popis firmy Lear Corporation Czech Republic, s. r. o.

Společnost byla založena roku 1917 v Detroitu, ve Spojených státech amerických pod názvem American Metal Products. V současnosti nese společnost název Lear Corporation a patří k největším dodavatelům automobilního průmyslu. Zaměstnává přes 75 000 lidí ve 197 pobočkách umístěných ve 35 zemích světa. V České republice je Lear zastoupen dvěma výrobními závody a to v Kolíně, který je součástí divize Seating, zabývající se výrobou sedadel a závodem ve Vyškově, patřící do divize T&C, neboli Terminals and Connectors (terminály a konektory). Pro upřesnění jsou ve Vyškově vyráběny konektory, pojistkové a propojovací skříně pro automobilní průmysl. Vyškovská pobočka byla začleněna do korporace v roce 2004, kdy LEAR odkoupil společnost GHW. LEAR Vyškov aktuálně zaměstnává přes 350 zaměstnanců a vzhledem k současnému nárůstu objemu výroby, tento počet stále stoupá.



Obr. 1 Logo společnosti Lear Corporation.

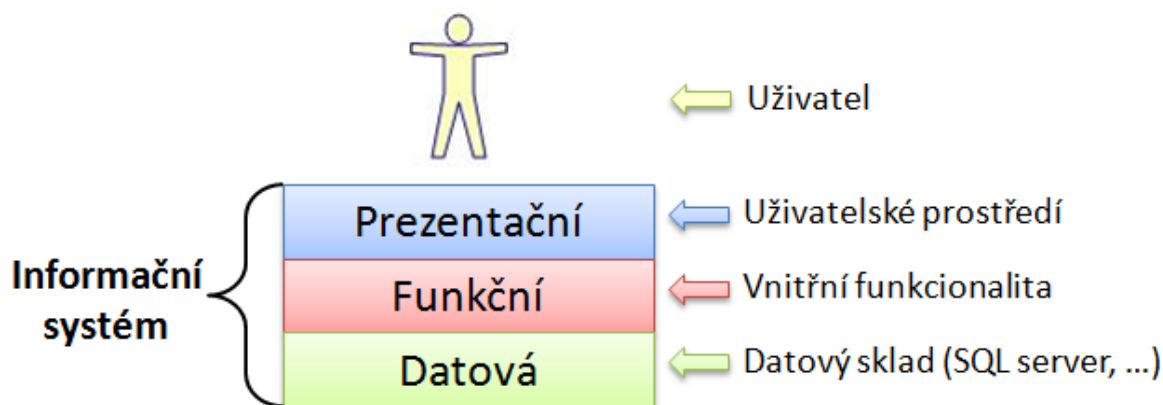
1.2 Informační systém

Pod pojmem informační systémem, můžeme chápat jakoukoliv agendu, sloužící k zaznamenávání, uchovávání a zpětnému získávání informací. Příkladem informačního systému může být třeba kartotéka, telefonní seznam, kniha došlé pošty nebo třeba účetnictví. Systém nemusí být nutně automatizovaný pomocí počítačů, může být veden i v papírové podobě.[1]

Jedním z hlavních důvodů pro zavádění strukturovaných elektronických informačních systémů je eliminování lidského faktoru. Je-li umožněno provádět nesystémové zásahy ať už do jakéhokoliv systému sběru dat, pak dříve či později, vlastní nekázní lidí vstupujících do tohoto procesu, nevyhnutelně dochází k nekonzistenci takto sbíraných dat. Je-li vyžadována jednotná struktura vedených záznamů a efektivní zpětná vazba, pak toto vede jednoznačně k zavedení specializovaného informačního systému, komplexně postihujícího danou problematiku s pevně danými kompetencemi a zabudovanou účinnou verifikací vstupních dat.

Elektronický informační systém bývá obvykle rozdělen na tři základní vrstvy. Jak je znázorněno na obrázku číslo 2, první z nich je vrstva prezentační. Tu lze chápat

jako uživatelské rozhraní IS. Jedná se tedy o tu část systému, která je viditelná prostému uživateli, a tudíž ta, s níž uživatel pracuje. Druhá je funkční vrstva. Ta se stará o verifikaci vstupních dat a zprostředkování informací obsažených v IS a zajišťuje propojení mezi prezentační a datovou vrstvou. Třetí a poslední vrstvou je právě vrstva datová, obvykle ztělesněná databázovým serverem, či jiným úložištěm dat.



Obr. 2 Vrstvy elektronického informačního systému.

1.3 Historie vývoje a implementace projektu

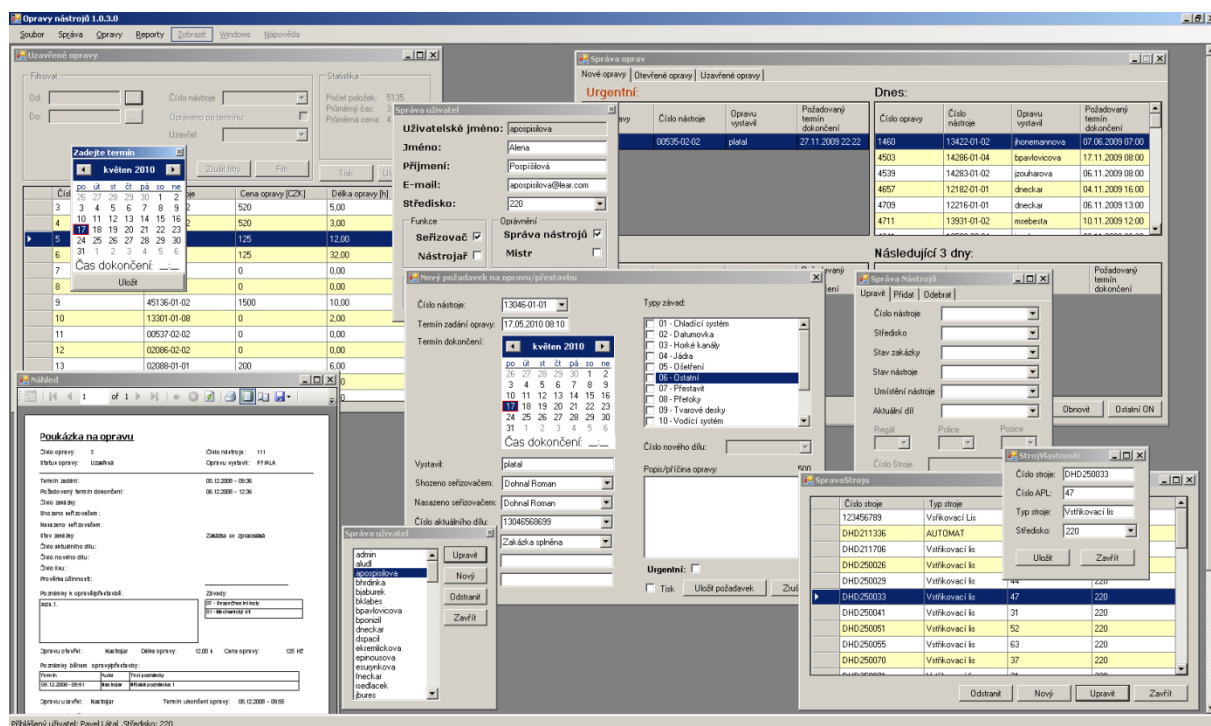
Požadavek na vytvoření elektronické evidence servisních zásahů na výrobních odděleních vznikl koncem roku 2008. Tehdy management výroby usoudil, že současná situace v evidenci servisů se, vzhledem k intenzitě nárůstu množství výrobních zakázek a přísunu nových výrobních nástrojů a linek z jiných evropských závodů společnosti LEAR, stává neudržitelnou. Nový informační systém se měl stát, spolu s dalšími opatřeními, nástrojem k udržení úrovně kvality výroby vyjádřené zejména hodnotou zmetkovitosti v PPM (Parts Per Million, neboli počet vadných kusů na milion vyrobených) i nadále pod plánovanou hranicí. Jinak řečeno měl posloužit jako pomocník při zefektivňování a zkvalitňování výroby. Hlavním požadavkem bylo zavést elektronickou evidenci co nejdříve.

Začátkem roku 2009 byla dána do provozu zkušební verze aplikace Opravy nástrojů, jak byl systém pojmenován, ve verzi 1.0. Systém obsahoval zatím jen základní funkcionalitu, která umožňovala v podstatě stejnou evidenci jako původní papírová podoba, s tím rozdílem, že již veškeré požadavky na opravu, či přestavbu byly trvale, elektronicky evidovány. Dále byly již, díky zavedení uživatelských přístupů, pevně vyhrazeny kompetence zásahů jednotlivých zaměstnanců do procesu servisu. V počátcích testovacího provozu byly veškeré požadavky, tzv. opravenky také ze systému tisknuty a to hlavně z důvodu, aby si zaměstnanci snadněji zvykli na nový systém, k němuž měli v období jeho zavádění značné výhrady. Vzhledem, ale k významné podpoře a spolupráci vedení výroby v prosazování nového systému, byly postupně všechny problémy a nedostatky vyřešeny k všeobecné spokojenosti.

V první polovině února 2009 byly Opravy nástrojů schváleny do „ostrého“ provozu a původní papírová evidence byla odsunuta do pozice zálohy pro interní či zákaznický audit. Prvotní cíl, zahájit co nejdříve elektronickou evidenci, byl splněn.

Navíc bylo rozhodnuto, že data z původní papírové evidence nebudou nijak zahrnuta do nového IS.

V průběhu roku 2009 byla postupně doplňována další funkcionalita, například prostředí pro filtrování v dokončených opravenkách, či doplnění reportů sloužících jako podklad pro personální oddělení při ohodnocování pracovníků nástrojárny a podobně. Vznikaly postupně verze 1.0.1 – 1.0.3 Na obrázku níže je náhled poslední verze prvního vydání Oprav nástrojů.



Obr. 3 Opravy nástrojů 1.0.3.

V červnu 2009 bylo rozhodnuto o vytvoření aplikace Opravy nástrojů 2.0. Toto rozhodnutí bylo založeno zejména na narůstajícím množství požadavků na funkcionalitu, jež měla aplikace splňovat, s nimiž v původním projektu nebylo počítáno a jejich implementace by vyžadovala komplikovaný a nesystémový zásah do stávající aplikace. Druhým důvodem pro kompletní přepracování, bylo nastolení myšlenky jednotného informačního systému, jež by zahrnoval vícero firemních procesů a zastřešoval tak hlavní potřeby informačních systémů ve firmě. Tím se měl stát LVIS, tedy Lear Vyškov Information System, domácíky Elvis.

Myšlenka vznikla na základě inspirace dalších oddělení systémem Opravy nástrojů 1.0, kdy si manažeři těchto oddělení a stejně tak jejich podřízení uvědomili možnosti, které jim skýtá specializovaný software, takzvaně „šitý na míru“ jejich požadavkům. Agenda servisního managementu totiž nebyla jedinou oblastí, která si žádala přehodnocení. Příkladem je oddělení kvality s evidencí ICHH, tzv. Interních Chybových Hlášení, nebo také MIS, neboli Management Information System, který bude sloužit vedení firmy jako sběrné a vyhodnocovací pole pro nejrůznější data a ukazatele. Všechny tyto informační systémy tvoří moduly, nebo také jinak subsystémy právě zmíněného, hlavního informačního systému LVIS.

Hlavním účelem zastřešení všech informačních systémů do jediného je, aby uživatelé pracující se všemi těmito systémy, kdy je vyžadována autorizace, tedy přihlášení, měli jen jedno jediné uživatelské jméno a heslo na jednom místě a zároveň jom sloužil jako rozcestník.

Nakonec vlastní přepracování aplikace do verze 2.0 mělo také přinést zefektivnění kódu a přepracování systému přístupových oprávnění a také revizi zpracování uživatelského rozhraní.



Obr. 4 Časová osa vývoje IS pro servisní management.

Nová verze byla na rozdíl od původní, jež byla tvořena v jazyce Visual basic, naprogramována v rozšířenějším a v současnosti ve firmě preferovanějším programovacím jazyku C#.

Hlavními přínosy nové verze Oprav nástrojů jsou tedy kromě jejich začlenění do systému LVIS také, právě flexibilní management přístupových oprávnění, přehlednější a příjemnější uživatelské rozhraní, zavedení efektivnějšího přístupu k aktualizacím aplikace a také zdrojový kód, který je daleko otevřenější případným budoucím modifikacím a rozšířením.



Obr. 5 Logo systému LVIS a systému Opravy nástrojů 2.0.

2 Projektová příprava

Jedním z nejdůležitějších aspektů každého projektu, nezávisle na jeho rozsahu je bezpochyby projektová příprava. Na počátku je obvykle nejasná představa zadavatele, kterou je nezbytné cílenou metodikou konkretizovat a to ještě před jeho vlastní realizací. Ať už se jedná o komplikovanou stavbu visutého mostu, nebo malý, krátkodobý projekt, vždy má projektová příprava své opodstatnění. Vždy je třeba předem promyslet všechny aspekty připravovaného projektu, identifikovat takzvaná úzká a kritická místa projektu, určit prostředky a zdroje, jimiž bude projekt realizován a tak dále. Rozsah projektové přípravy je obvykle úměrný rozsahu projektu, pro nějž je vytvářena. Takovéto projektové přípravy se pak liší nejen obsáhlostí, ale i použitými metodikami při jejich vytváření.

Příprava softwarového projektu má svá specifika, která však vycházejí z obecně platných pravidel. Před započítím projektu je důležité vyjasnit několik základních aspektů, které představují hlavní rysy projektu.

- O jaký software půjde?
- Jaký je časový fond, neboli kdy má být projekt hotov?
- Pro koho je software určen?
- Jaké metodiky budou při tvorbě software použity?

Při projektové přípravě byly využity dvě základní metodiky. V první fázi rozvíjení myšlenek a vizí toho co všechno by měl nový informační systém zahrnovat, posloužila jako účinná pomůcka metoda myšlenkové mapy. Ta byla zvolena pro svou otevřenost a názornost s jakou lze snadno a rychle převést matnou představu v komplexnější náhled na řešenou problematiku efektivnějším zapojením potenciálu lidského mozku. Ve druhé fázi, kdy byla diskutována konkrétní funkcionalita, byly uplatněny grafické techniky jazyka UML, které se ukázaly jako výborný nástroj pro komunikaci se zadavatelem, ale hlavně pro vytvoření uceleného modelu budoucí aplikace.

2.1 Základní požadavky zadavatele

Na počátku projektu je vždy delší, či kratší seznam požadavků, které zadavatel od nového systému očekává. Tento seznam se v průběhu projektové přípravy obvykle rozšiřuje. Toto se nazývá proces identifikace požadavků.

U systému Opravy nástrojů to byly základní požadavky tyto:

- Sjednocení systému evidence servisních zásahů a jeho převedení do elektronické podoby.
- Vymezení kompetencí jednotlivých účastníků procesu servisu.
- Vytvoření vypovídajících reportů a přehledů ze získaných dat.
- Přednostní využití stávajících, firemních prostředků.

2.2 Definování řešeného problému

Jako odpověď na základní otázky projektové přípravy můžeme využít myšlenku takzvaného „výťahového prodeje“. Scénář myšlenky je asi takový: jste na cestě na schůzku s investorem v 87. podlaží mrakodrapu, máte notebook s pečlivě připravenou 15 minutovou prezentací vypilovanou do zářivé podoby. Jakmile ale nastoupíte do výtahu v hale budovy, zástupce investorské firmy nastoupí společně s vámi. „Tak Františku,“ řekne „co pro nás dneska máš?“ V tu chvíli zapomeňte na 15 minutovou prezentaci. Máte 30 sekund jízdy výtahu na to, abyste dokázali vysvětlit základní rysy vaší plánované aplikace a učinili na něj přitom rozhodující dojem. Co mu řeknete?[2]

I když zadavatel nemá 87 patrovou budovu, tak i přesto má tato myšlenka své opodstatnění. Hlavním přínosem je přesně a krátce definovat základní myšlenku projektu a tak ujasnit nejdůležitější záměr, jehož má být dosaženo. Praktický efekt toto nabývá v okamžiku, kdy zadavatel požaduje doplnění nové funkcionality. Pokud tato funkcionality nezapadá do základní myšlenky výťahového prodeje, pak přichází na řadu dvě možnosti. Buď ji z projektu vypustit, nebo prodloužit „cestu výtahem“, neboli navýšit potřebné zdroje, tedy zejména čas, finance, případně zapojit do projektu více lidí. Největším nepřítelem dokončení projektu je právě kontinuální přidávání funkcionality softwaru.[2]

2.2.1 Definice řešeného problému

Cílem projektu je tedy vytvořit elektronický, informační systém pro evidenci, sledování a vyhodnocování servisních zásahů na stříhacích, lisovacích, vstřikovacích, montážních a dalších výrobních nástrojích a strojích ve výrobním závodě společnosti LEAR Corporation, Czech, s. r. o. ve Vyškově. Software zpřehlední a hlavně zefektivní agendu servisního managementu, což přinese zejména zvýšení kvality a efektivity práce servisních oddělení podniku.

2.3 Základní rozbor problému

Informační systém je určen zcela výhradně pro výrobní závod Lear ve Vyškově. Není tedy s ním počítáno s nasazením v jiných firemních závodech, ani není určen pro jiné firmy. Bude se tudíž jednat o úzce specializované řešení, v němž nebude nutné primárně zohledňovat faktory jako jazykové mutace, či propojení systému napříč několika, vzdálenými závody. Také z hlediska zabezpečení přenášených dat není nezbytné zavádět zvláštní opatření, neboť lokální síť, na níž bude systém provozován, je sama o sobě již dostatečně zabezpečena proti vnějším útokům a přístup do ní zevnitř je podmíněn uživatelským jménem a heslem.

Bude nutné vytvořit datové úložiště pro ukládání záznamů o opravách a dalších režijních dat. Toto bude realizováno na vlastním aplikačním serveru s nainstalovaným Microsoft SQL serverem. Veškerá, požadovaná funkcionality bude zahrnuta do klientské aplikace obsluhující datové úložiště.

Celá aplikace bude rozvrstvena do několika zón s odstupňovanými požadavky na přístupová oprávnění. Vzhledem ke zkušenostem při tvorbě první verze bude vytvořen takzvaný flexibilní systém oprávnění, kdy bude v jistém rozsahu umožňovat konkrétním osobám, či skupinám osob přístup k jednotlivým operacím v systému.

2.4 Analýza procesu

Velmi důležitou etapou projektové přípravy informačního systému, je seznámení se s vlastním procesem, jenž bude nově vznikající systém zastřešovat. Je zcela nezbytné, aby analytik projektu byl plně zainteresován do všech podrobností procesu, aby tyto mohly být vhodně zakomponovány do komplexního celku aplikace.

V prvotní fázi se jeví ideálním postupem, fyzicky si projít všemi etapami procesu, nejlépe v doprovodu osoby, která je správcem tohoto procesu a vytvořit si soubor poznámek a náčrtů zachycujících všechna podstatná zjištění. Poté závisí na vzájemné komunikaci obou hlavních aktérů, tedy průvodce, ale hlavně analytika, aby dokázali odhalit všechny, či alespoň většinu klíčových bodů pro nástin základní analýzy procesu. Je možné a někdy i přímo vhodné své případné dotazy směřovat i na přímé účastníky procesu, zaměstnance provozu a podobně, na jejich návyky a rutiny.

V další fázi následuje zpracování získaných poznatků, například pomocí metody myšlenkové mapy. Je důležité, aby zadavatel byl v zastoupení například správcem procesu, byl dostatečně zapojen do této fáze projektu. Je vhodné mít domluvený pevně daný termín schůzek, například jednou týdně, kdy jsou diskutovány návrhy a závěry analýzy procesu se zadavatelem, s účastníky procesu a dalšími osobami, které mají k celému procesu takzvaně co říci. Tato fáze zapadá do fáze identifikace požadavků. V průběhu této etapy je již tvořena konkrétní představa celé koncepce a obsahu projektu.

Ke komplexnějšímu vyjádření důležitých aspektů projektu je vhodné využít například jazyk UML, jako prostředníka mezi analytikem, programátorem a zadavatelem. Obecně platí, že precizně provedená analýza je základem kvalitního systému.

Před dokončením analytické fáze projektu je velmi vhodné mít od zadavatele písemně odsouhlasené závěry provedených analýz. Dojde-li v době předání hotové aplikace k nesrovnalostem týkajících se jejího obsahu, či funkcionality, pak se jedná o velmi účinnou obranu proti případným reklamám.

Jsou-li identifikovány všechny požadavky a odsouhlaseny všechny závěry provedených analýz, pak teprve lze přistoupit k započetí tvorby aplikace.

2.4.1 Analýza procesu servisních zásahů

Prvotní seznámení s procesem servisních zásahů proběhlo formou exkurze servisními odděleními v doprovodu hlavního mistra oddělení plechů a představením základních, nejen administrativních kroků, které jsou v průběhu tohoto servisu prováděny. Během následujících konzultací s hlavními a směnovými mistry jednotlivých výrobních a servisních oddělení, byly představy konkretizovány do výsledné podoby. Výsledky analýzy znázorněné UML diagramy jsou předmětem přílohy číslo 3.

Jelikož se jedná o vnitrofiremní zadání, pak nebylo považováno za nezbytné písemné potvrzování získaných závěrů z jednotlivých kroků procesní analýzy. V průběhu realizace projektu se však tato domněnka projevila jako mylná. Zkušenost prokázala, že nejsou-li doloženy písemná potvrzení, pak nezbyvá než se podvolit vůli zadavatele bez nároku na navýšení časového, případně finančního fondu na vývoj aplikace.

2.4.2 Popis procesu

Životní cyklus servisního zásahu, nebo také ve vnitrofiremním názvosloví *Opravenky*, začíná u zadavatele požadavku na servis. Tento požadavek je poněkud odlišný pro servis nástroje a servis stroje. Zatímco pro střediska Plasty a Plechy jsou zadávány požadavky na servis nástrojů, u střediska Montáž jsou takto evidovány zásahy na montážních strojích.

Pro zadávání servisních požadavků platí pravidlo, že požadavek na servis konkrétního nástroje či stroje může primárně zadat jen osoba, která svým pracovním zařazením přísluší středisku tohoto stroje nebo nástroje. Není tedy možné, aby zaměstnanec střediska Montáž zadal servisní požadavek na nástroj vyrábějící plastové dílce, protože s ním nemá prvoplánově nic společného. Zadavatel tedy vyplní základní údaje jako číslo nástroje, případně stroje, požadované servisní zásahy, zda byla výrobní zakázka splněna, či nikoliv, bližší popis požadovaného servisu, a tak dále.

Poté je nástroj, případně stroj servisován. Pro středisko Plastů a Plechů tento servis zajišťují nástrojaři, pro středisko Montáž jsou to seřizovači. Pro střediska Plasty a Plechy jsou také zaměstnanci vedeni jako seřizovači, kteří mají na starosti stroje těchto středisek, ale evidence jejich zásahů není předmětem firemního zadání. Přesto jsou vedeni u každé opravy nástroje jako osoby, které nástroj „shodil“, neboli vymontoval ze stroje či jej naopak „nasadil“. Také jsou zde zaměstnanci, kteří plní současně funkci seřizovače i nástrojaře. V průběhu servisu jsou zaznamenávány důležité poznámky, ale hlavně čas na nich strávený, případně cena například náhradních dílů, které bylo nezbytné zakoupit. Všechny tyto záznamy souvisejí s vlastním prováděním servisem, tudíž je mohou zadávat pouze osoby, které tento servis provádějí, tedy nástrojaři, případně seřizovači. Zvláštním případem je ještě vliv zaměstnanců na pozici OTK (osoba technické kontroly), kteří taktéž k „otevřené“ servisní zakázce, tedy zakázce která je v řešení, mohou přidávat své poznámky.

Vzhledem k tomu, že se v průběhu servisu stává, že je potřeba provést další servisní zásahy, které nebyly v původním požadavku, pak je třeba, aby bylo umožněno zadaný požadavek v průběhu servisu do jisté míry i měnit. Tyto změny jsou konzultovány s nadřízeným, tedy mistrem daného střediska, který je oprávněn tyto změny schválit.

Na závěr jsou zaznamenány poznámky o provedeném servisu. Nástroj či stroj je vrácen do provozu, případně uskladněn a servis je uzavřen. Toto vše spadá do kompetence servis provádějících osob. Tímto končí proces, který má být zahrnut do nového elektronického informačního systému.

3 Myšlenková mapa

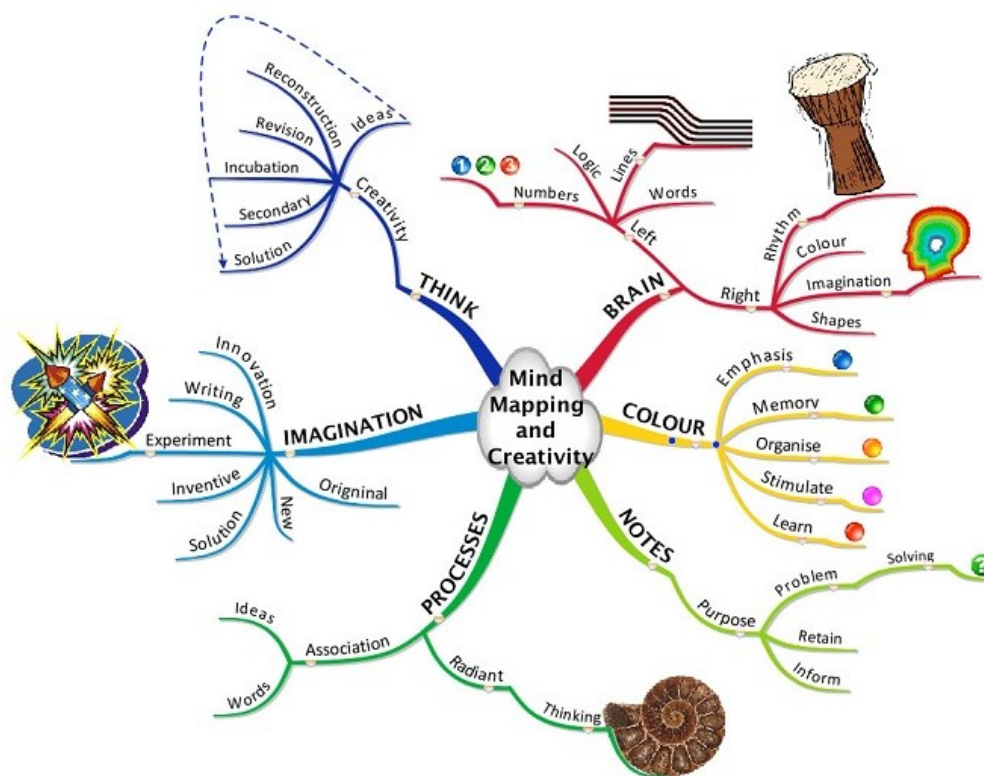
Myšlenková mapa je grafická technika sloužící pro uvolnění mozkového potenciálu, tzv. brainstorming. Jedná se v podstatě o diagram v dvojrozměrném prostoru, využívající kombinaci textu, grafiky, barev, logiky a znázornění návaznosti prvků diagramu. Myšlenkovou mapu si lze představit jako druh dvojrozměrného přehledu, kterým popisujeme informace jinak těžko popsatelné slovy.[2]

Za otce metody myšlenkové mapy je považován Tony Buzan. Narodil se 2. 6. 1942 v Londýně. Je předním spisovatelem knih, článků a pojednání o lidském mozku a o výuce. [3]

Metoda myšlenkových map vychází z principu propojení pravé a levé hemisféry mozku a tím výkonnějšího a hlavně efektivnějšího využití mozku. Propojením levé hemisféry (nositelka logického a exaktního myšlení, plánování, zabývá se slovy, čísly, analýzou ...) a pravé hemisféry (ta dominuje v představivosti, snění, umění a zabývá se takovými pojmy, jako jsou barvy, rytmus, prostorové vědomí aj...) můžeme lépe využít možnosti a schopnosti našeho mozku a myšlení.

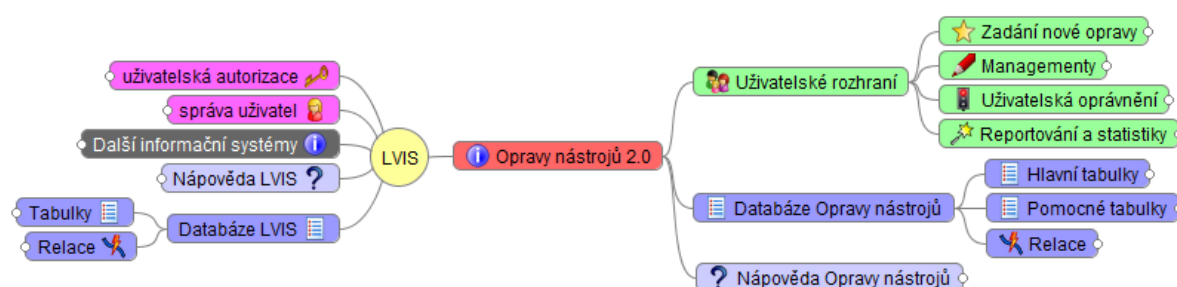
Druhý základní princip vychází z poznatků, že lidský mozek „nemyslí“ jednoduše a lineárně, ale expanzivně a synergicky. Je tak schopen vymyslet nekonečné množství řešení problémů nebo úvah. [5]

Myšlenkové mapy lze využít například pro zefektivnění učení, pro osobní rozvoj, nebo právě při vývoji softwarových aplikací. Na obrázku níže je zobrazen příklad myšlenkové mapy popisující prvky a aspekty vyskytující se při tvorbě vlastních myšlenkových map.



Obr. 6 Ukázka myšlenkové mapy.[4]

Metoda myšlenkové mapy se ukázala jako silná a efektivní pomůcka pro identifikaci požadavků a ucelování představ zákazníka a také pro získání lepšího náhledu na obsáhlost projektu již v počátečních fázích jeho plánování. Otevřenost a názornost myšlenkové mapy umožňuje začlenění do procesu identifikace požadavků i osoby, které nejsou až do takové míry zainteresovány do procesu vývoje, například budoucí, koncové uživatele, a umožnit i jim vyjádřit se k obsahu aplikace. Na obrázku číslo 7, jsou zobrazeny základní bloky prvků informačního systému Opravy nástrojů 2.0 s provázáním na IS LVIS. K vytváření myšlenkových map byl využit program FreeMind, který je volně dostupný, ale který je spíše zaměřen na textové, než grafické vyjádření prvků mapy. Pro znázornění myšlenkové mapy IS byl ale zcela vyhovující.



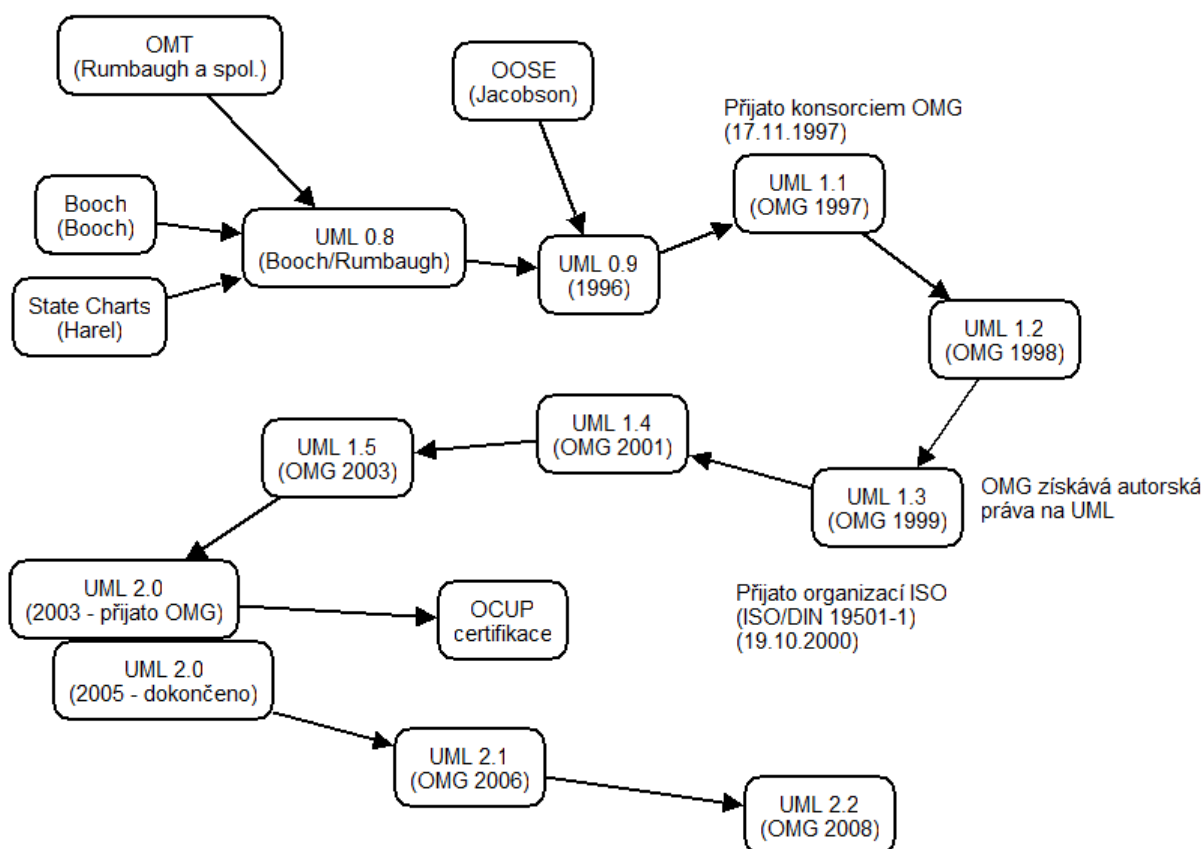
Obr. 7 Částečná myšlenková mapa informačního systému LVIS.

Z kompletního znázornění myšlenkové mapy (viz příloha číslo 2) je patrná základní struktura a obsáhlost celého systému. O správu uživatel a uživatelskou autorizaci se stará nadřazený systém LVIS. Prvky a aspekty klientské aplikace jsou zahrnuty v sekci Uživatelské rozhraní. Struktura datové vrstvy zase v sekci Databáze. Pro nápovědu, tedy uživatelskou dokumentaci je vyčleněna také vlastní větev diagramu. Od prvotní podoby se myšlenková mapa značně změnila, ale to je v průběhu procesu identifikace požadavků zcela běžné.

4 UML

Vlastní zkratka UML je složena z počátečních písmen slov Unified Modeling Language, což v překladu znamená unifikovaný modelovací jazyk. UML slouží jako výkonný nástroj pro softwarové inženýrství. Zahrnuje množství grafických technik pro vytvoření vizuálních modelů, znázorňujících celou šíři spektra softwarového návrhu objektivě orientovaného softwarového projektu. UML poskytuje standardizovanou cestu jak graficky zobrazit architekturu vznikající aplikace, interakce s uživateli, interní firemní procesy, komponenty, strukturu aplikace, schémata databází, a podobně. Jedná se o kombinaci technik z oblastí objektového, datového a obchodního modelování, které nacházejí uplatnění napříč celým životním cyklem softwaru, od prvotního plánování až po finální nasazení do provozu, nezávisle na technologiích, které jsou využity pro vlastní implementaci konkrétního projektu. UML se za dobu své existence stal jistým průmyslovým standardem.

Metodiky jazyka UML je zastřešována společností OMG (Object Management Group), což je konsorcium zaměřené na nastolování standardů pro objektivě orientované systémy.

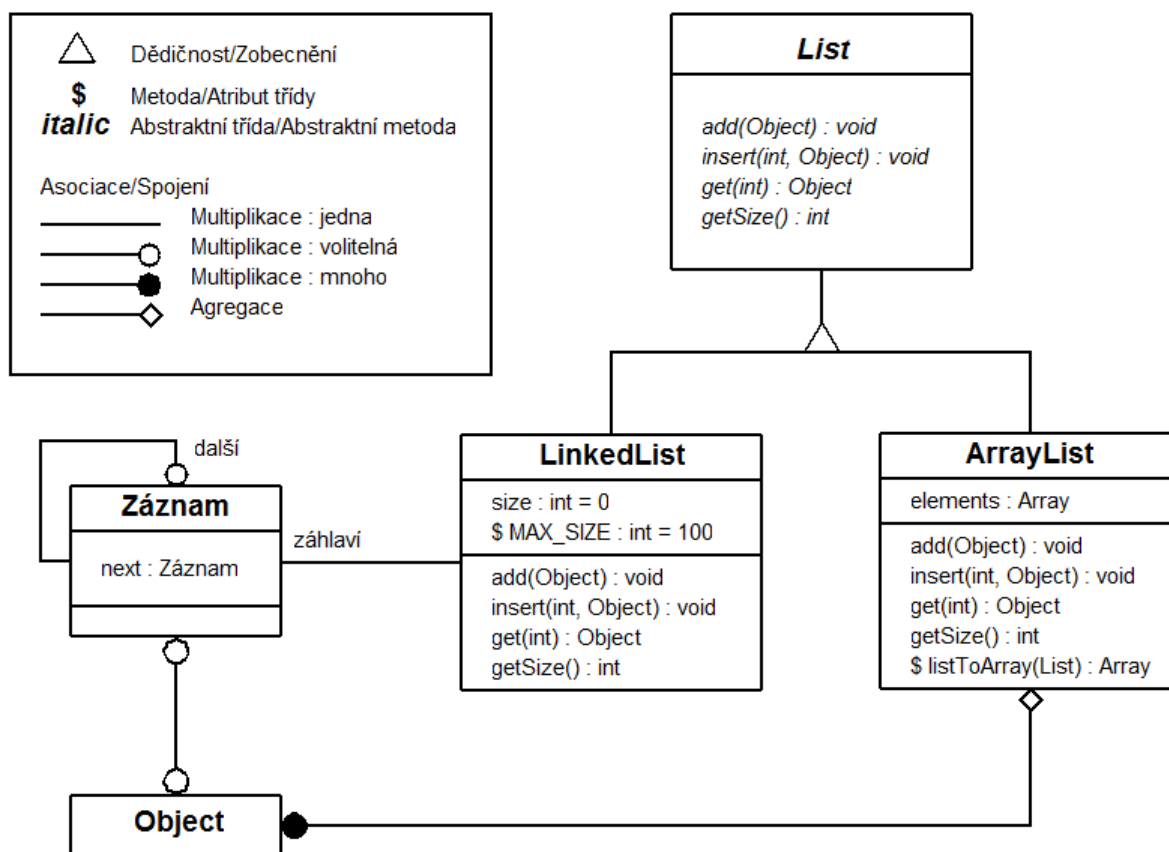


Obr. 8 Diagram historického vývoje jazyka UML.

4.1 Historie UML

Modelovací jazyk UML vznikl v průběhu osmdesátých a devadesátých let minulého století. Jednalo se o snahu vytvořit techniky umožňující popsat objektově orientovanou analýzu a návrh. V polovině 90. let byly velmi rozšířené metody OMT (Object Modeling Technique) vyvinuté skupinou Amerických vědců pod vedením Jamese Rumbaugh roku 1991 ve společnosti Rational software corporation. K základním prvkům OMT patřilo testování fyzikálních entit před jejich vlastním vytvořením (simulace), komunikace se zákazníkem a vizualizace (alternativní prezentace informací). V rámci OMT byly navrženy tři základní typy modelů pro analýzu řešené problematiky softwarového, objektově orientovaného návrhu. Objektový, dynamický a funkcionální model.

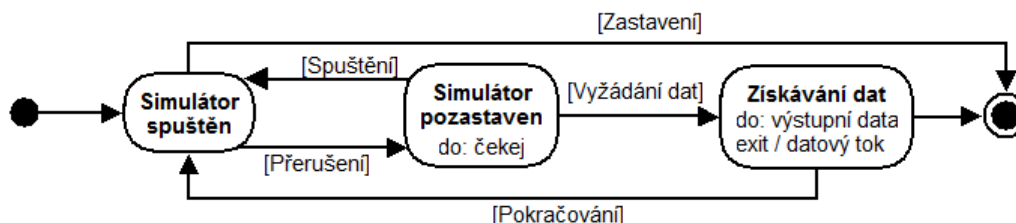
Objektový model představoval statickou část modelované domény. Nejhlavnějšími prvky tohoto modelu byly třídy s jejich metodami a vlastnostmi. Příklad využívaných prvků u tohoto modelu je na obrázku číslo 9.



Obr. 9 Vzorový diagram objektového modelování OMT.

Dynamický model v sobě zahrnoval stavový náhled na modelovanou doménu. Hlavním konceptem byly stavy, přechody mezi stavy a události vedoucí ke změnám stavu. Vzorový příklad stavového modelu, viz obrázek číslo 10.

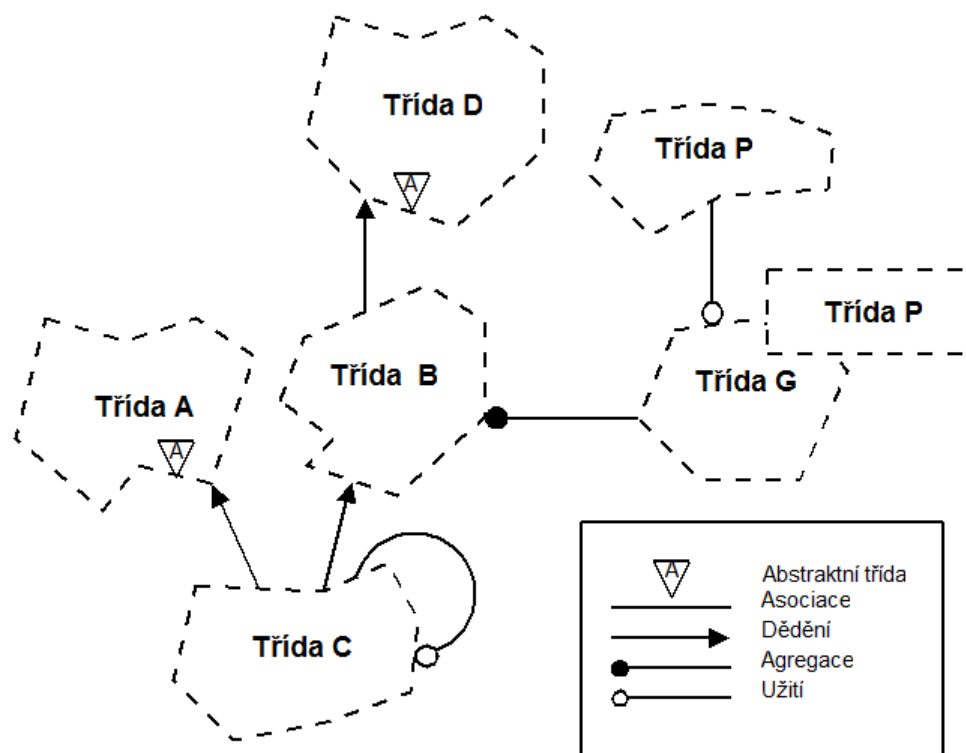
Nakonec funkcionální model se stará o tok dat v modelované doméně. Hlavními zahrnutými pojmy jsou proces, úložiště dat, tok dat a aktéři. Jazyk UML podědil z OMT mnohé modelovací prvky.



Obr. 10 Příklad diagramu stavového modelu OMT.

Dalším významným předchůdcem UML byla metodika objektově orientovaného softwarového inženýrství OOSE (Object-oriented software engineering) vyvinuté roku 1992 Švédským vědcem Iivarem Jacobsonem, v té době pracujícím pro společnost Objectory AB. Jednalo se o první objektově orientovanou techniku zanesení případů užití (use cases) do softwarového designu.

Třetím významným, přímým předchůdcem prvního jazyka UML byla metoda Booch vyvinutá stejnojmenným americkým softwarovým inženýrem Grady Boochem, v té době zaměstnancem společnosti Rational Software. Jedná se o metodiku určenou primárně pro objektové programování. Někdy je tato metoda také nazývána Cloud station. Za zmínku k předchůdcům jazyka UML ještě stojí *Statecharts* (stavové grafy) Davida Harrela.



Obr. 11 Ukázka diagramu metody Booch.

Poté co roku 1995 Rational Software Corporation odkoupila společnost Objectory AB, započaly práce na sjednocení různých metodik a syntaxí, jejichž výsledkem byl roku 1997 jazyk UML 1.1, jenž byl také přijat právě konsorciem OMG.

V průběhu vývoje UML verze 1.x bylo od některých metodik a diagramů upuštěno, jako například „Cloud notation“ Gradyho Booche, jiné byly přehodnoceny či doplněny. Příkladem nových metod způsobů zápisů je OOSD (objektově orientovaný strukturovaný design) od autorů Tonnyho Wassermanna a Petera Pirchera, analýza případů užití a časová analýza od Archieho Bowena, nebo stavové diagramy od Davida Harela. Výsledkem bylo, že se UML mohlo konečně pokrýt nejrůznější vývojářské problematiky od jednoduchých procesů a jednoduchých aplikací, až po komplikované, rozsáhlé, distribuované systémy. Nejvýznamnější vývoj UML prodělalo do verze UML 1.1, poté bylo ještě uvolněno několik méně významných verzí (UML 1.3, 1.4 a 1.5), které však spíše odstraňovaly nedostatky a chyby v prvotním vydání. Verze 1.3 byla spíše významná tím, že následující rok po jejím uvedení, byla uznána jako ISO norma pro objektové modelování.

Významnější revize se UML dostalo v letech 2003 až 2005, na jejichž konci byla dokončena a konsorciem OMG přijata verze UML 2.0. Spolu s touto verzí jazyka UML byly společností OMG vytvořeny certifikační zkoušky s označením OCUP, které představují komplexní test znalostí specifikací jazyka UML, na jehož základě jsou udělovány tři úrovně certifikátů.

Současným průmyslovým standardem je UML ve verzi 2.2. Průběh hlavní linie vývoje jazyka UML s jeho nejdůležitějšími milníky je znázorněn na obrázku číslo 8.

4.2 Diagramy UML 2.2

UML 2.2 obsahuje celkem 14 typů diagramů rozdělených do dvou základních skupin. První skupina představuje diagramy popisující strukturální informace řešeného problému. Tyto diagramy popisují, co vše bude nově vznikající aplikace obsahovat a jak budou jednotlivé prvky uspořádány a provázány. Druhá popisuje interakční aspekty, neboli komunikace a interakce mezi prvky aplikace.

4.2.1 Strukturní diagramy

Strukturní diagramy představují strukturu aplikace, a proto jsou používány převážně k dokumentaci architektury softwarových systémů.

Diagramy tříd popisují strukturu systému zobrazením systémových tříd, jejich atributů a vztahy mezi třídami. Diagram komponent popisuje, jak je softwarový systém rozdělen na jednotlivé komponenty. Kompozitní, strukturní diagram popisuje interní strukturu tříd a spoluprací, jež tato struktura umožňuje. Diagram nasazení v celkovém modelu specifikuje hardware a prostředí použité při implementaci. Diagram objektů představuje kompletní, nebo částečný náhled na strukturu modelovaného systému v konkrétním čase. Diagram balíčků zobrazuje přehled dělení systému do logických skupin znázorněním závislostí mezi těmito skupinami. Profilový diagram postihuje úroveň metamodelu pro zobrazení stereotypů a profilů.

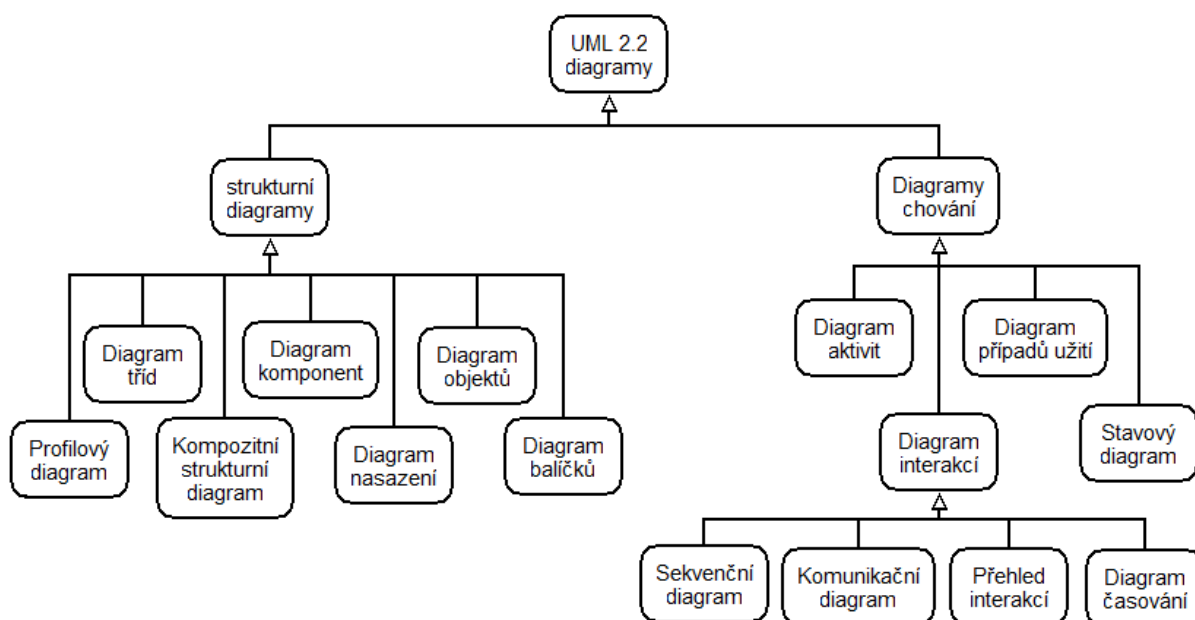
4.2.2 Diagramy chování

Významem diagramů chování je modelování dějů, které se v softwarovém systému musí odehrávat. Patří sem diagram aktivit znázorňující kroky obchodních

a pracovních postupů a tak představuje celkový tok řízení systému. Dalším diagramem je stavový diagram, pro zobrazení stavů a přechodů mezi stavy modelovaného systému. Posledním z trojice diagramů chování jsou diagramy případů užití sloužící k znázornění funkcionality poskytované systémem přímým uživatelům při konkrétních situacích využití softwaru.

Podskupinou diagramů chování představují diagramy interakcí zdůrazňující datové a funkcionální toky mezi jednotlivými prvky modelovaného systému. Komunikační diagram má na starosti znázornit interakce mezi objekty, nebo jeho částmi. Obsahuje kombinaci informací z diagramů tříd, sekvenčních diagramů a diagramů případů užití. Popisuje současně statické i dynamické chování systému.

Diagram přehledu interakcí je typem diagramu aktivit, jehož uzly představují diagramy interakcí. Sekvenční diagramy znázorňují, jak objekty navzájem komunikují v sekvencích zpráv. Také indikují životnost v závislosti na těchto zprávách. Na závěr diagram časování je specifický typem interakčního diagramu zaměřeného na časová omezení v modelovaném systému.



Obr. 12 Přehled UML diagramů.

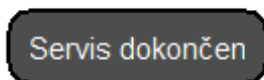
4.3 UML Diagramy použité při projektové přípravě

Při projektové přípravě informačního systému Opravy nástrojů, bylo z celkového výčtu metodiky UML 2.2 vybráno vzhledem k rozsahu a druhu řešené problematiky, několik diagramů, které svým obsahem a významem zcela postačovaly pro získání dostatečně komplexního náhledu na modelovanou oblast.

4.3.1 Stavový diagram

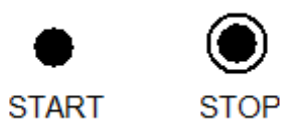
Slouží pro znázornění stavů, kterých může nabývat konkrétní objekt modelovaného systému. Základem syntaxe diagramu jsou stavy objektů. Dynamiku představují přechody mezi stavy a události, které tyto přechody vyvolávají.[10] Stav

objektu je charakterizován tak, že se jedná o konkrétní situaci, která nastala za doby života objektu.



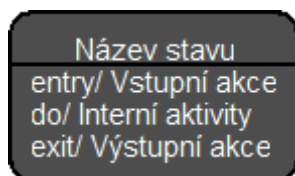
Obr. 13 Symbol stavu.

Základními symboly každého stavového diagramu jsou *Start* (bod zahájení diagramu) a *Stop* (bod ukončení diagramu). Každý stavový diagram musí mít právě jeden stav *Start* a nejméně jeden stav *Stop*. Stavy typu *Stop* můžeme označit alternativními názvy. Do symbolu *Stop* musí vést alespoň jeden přechod. [10]



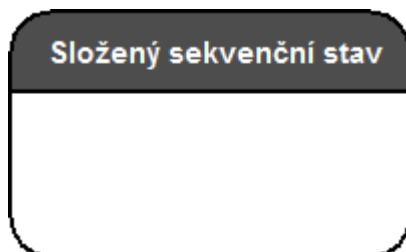
Obr. 14 Symboly Start a Stop.

Stav může také obsahovat akce a aktivity. Za *akci* považujeme nepřerušitelný, rychle probíhající proces, *aktivita* je přerušitelný, jistou dobu trvající proces. Každý stav má dvě akce – vstupní a výstupní, které jsou svázány s událostmi *entry* a *exit*. Událost *entry* proběhne jako první při přechodu do daného stavu, událost *exit* jako poslední před jeho opuštěním. K záznamu aktivit uvnitř stavu slouží klíčové slovo *do*. [10]



Obr. 15 Stav s vstupními a výstupními akcemi a vnitřní aktivitou.

Užitečným prvkem stavových diagramů jsou takzvané složené sekvenční stavy. Jedná se o možnost vyjádření vyšší úrovně stavu pro více stavů nižší úrovně současně. Na diagramu tak není nutné kreslit řadu přechodů současně z různých stavů, ale pouze jeden ze složeného stavu. Slouží také pro zjednodušení a zpřehlednění komplexních stavových diagramů. Symbol pro složený sekvenční stav je na obrázku číslo 16. [10]



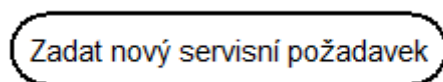
Obr. 16 Symbol složeného sekvenčního stavu.

Pro potřeby projektové přípravy informačního systému Opravy nástrojů byl vytvořen stavový diagram znázorňující všechny stavy, v nichž se v průběhu výroby nacházejí nástroje a stroje (viz příloha číslo 3, strana VIII). Součástí diagramu je blok složeného sekvenčního stavu s označením „V servisu“, v němž jsou zahrnuty stavy, které lze také chápat jako životní cyklus servisního požadavku.

Hlavním přínosem tohoto diagramu bylo hlubší pochopení návazností a propojení všech možných stavů, v nichž se mohou nástroje a stroje nacházet.

4.3.2 Diagram aktivit

Diagramy aktivit modelují procesy jako kolekce aktivit a přechodů mezi nimi. Jedná se svou podstatou o variantu stavového diagramu. Skládají se z pěti hlavních elementů. Prvním z nich jsou vlastní akce, jinak také *activity*. Za aktivitu přitom považujeme stav děláním čehokoliv. Akce se znázorňují obdélníkem se zaoblenými rohy s popisem názvu akce, většinou ve tvaru slovesné vazby jako stav konkrétního pracovního postupu. [10]



Obr. 17 Symbol akce.

Pro akce můžeme definovat následující vlastnosti:

- Jsou dále nedělitelné
- Není možné je přerušit – jednou zahájená aktivita musí být dokončena
- Jsou okamžité (probíhají rychle)
- Mají jeden vstupní a jeden výstupní přechod

[10]

Tak jako u stavových diagramů jsou i zde symboly pro *zahájení* a *zakočení* aktivit. Popis symbolů je volitelný, avšak u diagramů aktivit se spíše nepoužívá.

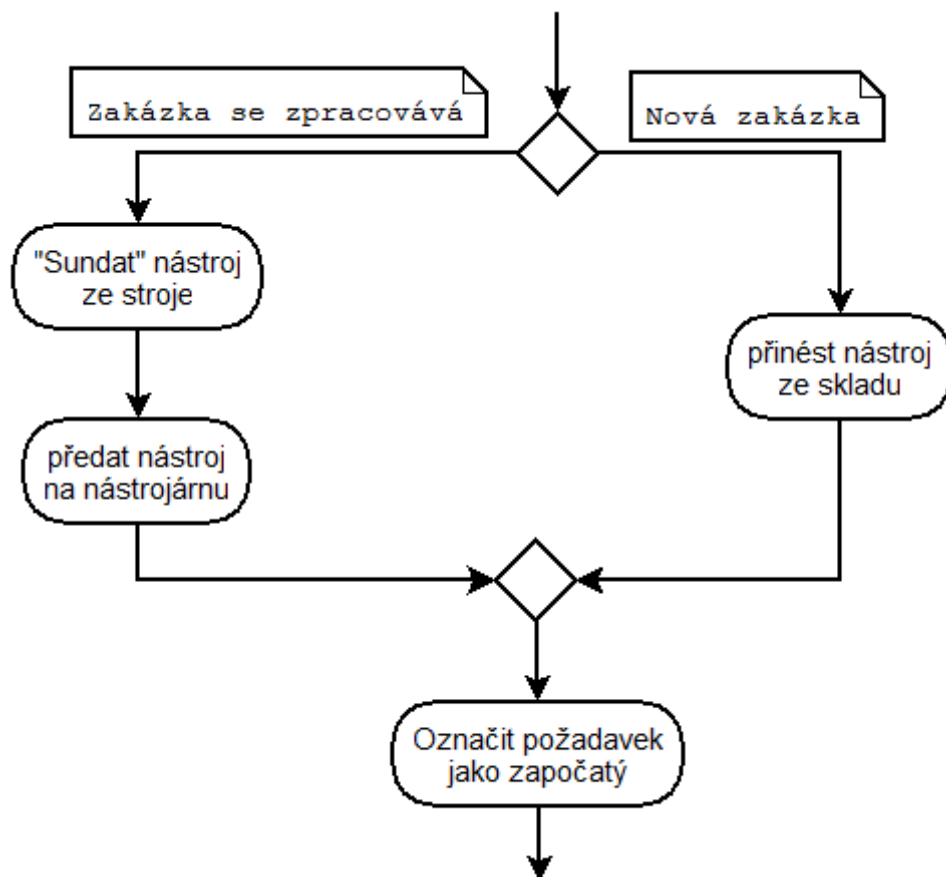


Obr. 18 Symboly pro zahájení a ukončení diagramu.

Přechody mezi akcemi jsou také, analogicky ke stavovým diagramům, značeny šipkami. Diagramy aktivit podporují i takzvané nelineární zpracování. Toto se znázorňuje elementem nazývaným *hodnocení přechodu*.

Jedná se v podstatě o vyjádření logické podmínky, která podmiňuje konkrétní přechod. Symbolem pro hodnocení přechodu je kosočtverec. Tento symbol má hned dvojí využití, jednou jako hodnocení přechodů, a jednou jako sloučení větví oddělených v hodnocení. [10] Podmínky hodnocení přechodu se uvádějí jako

poznámky u výstupních přechodů. Symbolika pro větvení vychází právě ze standardu UML 2.2, kde byla oproti předchozí verzi změněna.



Obr. 19 Příklad hodnocení přechodů.

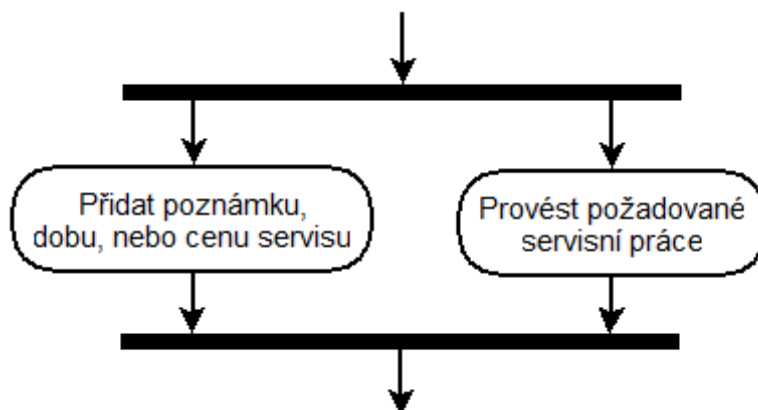
Na obrázku číslo 19 je znázorněn výřez z diagramu aktivit pro servisní zásahy. Ohodnocení přechodu zde zajišťuje zaznamenání rozdílného zpracování procesu v případě, že se pro konkrétní nástroj výrobní zakázka již zpracovává, tedy nástroj již vyrábí. V tom případě započetí servisních prací předchází ještě fyzické vyjmutí nástroje ze stroje. V opačném případě, tedy jedná-li se o novou zakázku, pak je nástroj přinesen ze skladu nástrojárny.

Pro hodnocení přechodů můžeme definovat následující podmínky:

- Do hodnocení vstupuje pouze jeden vstupní přechod.
- Počet výstupních přechodů není omezen.
- Každý výstupní přechod má definovanou podmínku.
- Podmínky výstupních přechodů musí být definovány tak, aby současně byla splněna pouze jedna z nich.
- Kromě definovaných podmínek, lze použít klíčové slovo „JINAK“. Přechod s touto podmínkou nastane tehdy, není-li splněna žádná z podmínek u ostatních přechodů.

[10]

Dalším z řady prvků a symbolů využívaných při konstrukcích diagramů aktivit jsou takzvaná *rozvětvení*. Díky nim lze právě modelovat zpracování paralelních procesů. K tomuto se používají symboly pro rozcestí (fork) a spojení (join). Rozcestí má vždy jeden vstupní a několik výstupních přechodů, spojení má naopak několik vstupních přechodů, ale jen jeden výstupní. Na rozdíl od hodnocení přechodu, kde dochází k výběru jen jednoho výstupního přechodu na základě vyhodnocení platnosti podmínek přiřazených k jednotlivým přechodům, zde v okamžiku aktivace vstupního přechodu dojde k aktivaci všech výstupních přechodů současně. [10]



Obr. 20 Ukázka rozvětvení diagramu aktivit.

Na obrázku číslo 20 výše je výřez diagramu aktivit popisující paralelní děj provádění vlastních servisních zásahů, při němž lze současně do systému zaznamenávat poznámky o průběhu servisu, ale také časy a částky, které si daný servis vyžádal. Obě tyto větve se setkávají v okamžiku, kdy se blíží dokončení všech servisních prací.

Posledním prvkem vyskytujícím se v diagramech aktivit jsou takzvané *plavecké dráhy*, nebo jinak také zóny. Ty slouží pro rozdělení modelované problematiky do oblastí podle toho, které zodpovědné osobě, či skupině náleží konkrétní aktivity. Aby bylo možné použít plavecké dráhy, je nutné upravit diagramy do podoby vertikálních pruhů vzájemně oddělených čarami. Každá taková dráha pak představuje oblast zodpovědnosti konkrétní osoby či oddělení. Použití plaveckých drah kombinuje vykreslení logiky pomocí diagramů aktivit se znázorněním zodpovědnosti interakčních diagramů. [10]

Na obrázku číslo 21 je znázorněn výřez diagramu aktivit s plaveckými dráhami, vytvořeném pro servisní zásahy na nástrojích a strojích.

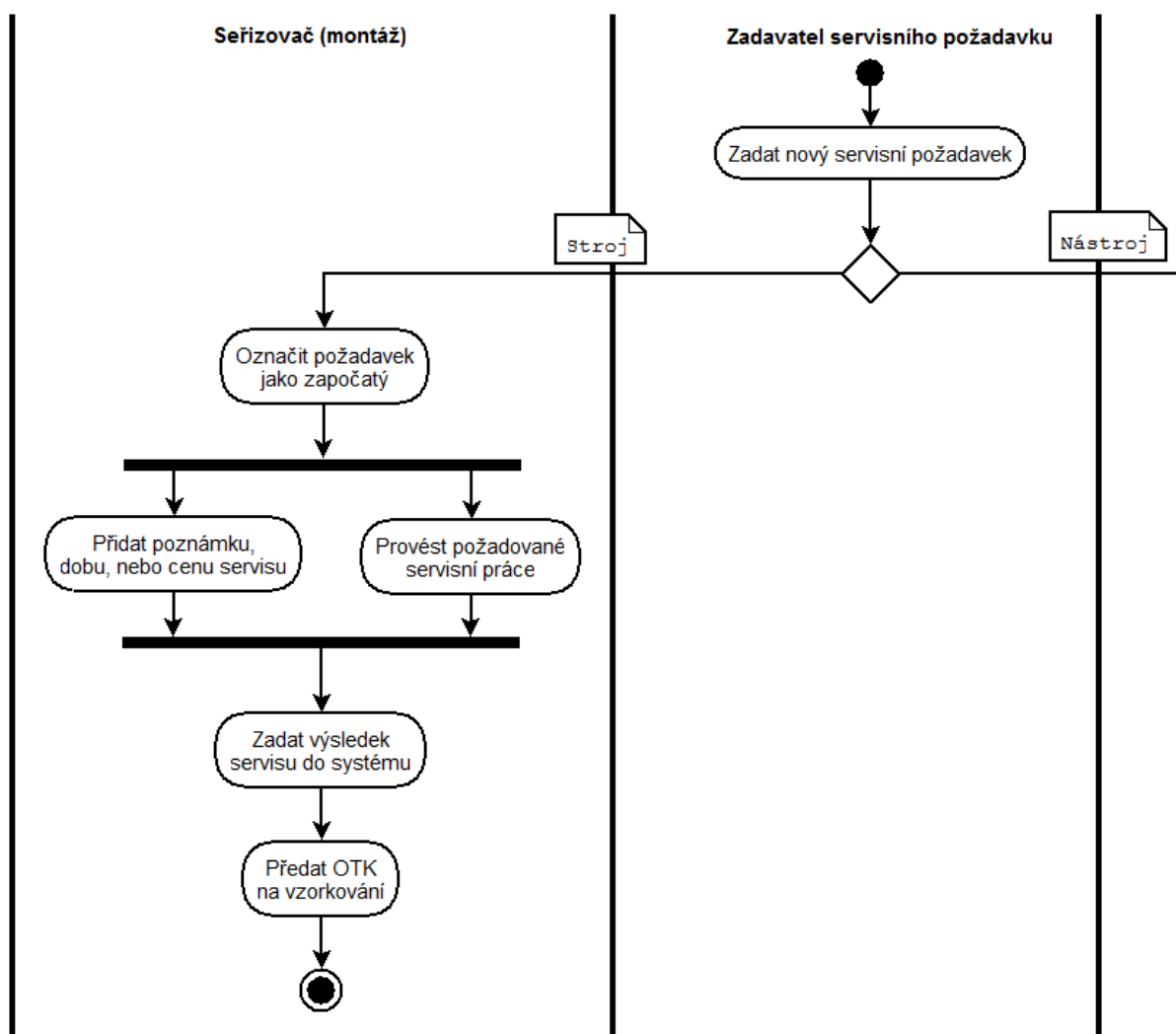
Jinou variantou zanesení zodpovědností za jednotlivé akce do diagramu je každou aktivitu označit zodpovědnou osobou, nebo skupinou osob zvlášť. Na první pohled se toto jeví jako jednodušší postup, ale ve srovnání s uspořádáním diagramu do plaveckých drah je toto mnohem méně přehledná metodika.

Diagramy aktivit také lze rozšířit o takzvané *toky objektů*. Zpravidla se jedná o důležité objekty, které v průběhu aktivit mění svůj stav.

Nevýhodou diagramů aktivit je, že nedokážou dostatečně jasně znázornit vazby mezi akcemi a objekty. Lze definovat vazby na objekty označením aktivit názvy

těchto objektů, nebo použitím plavečkových drah, který diagram rozdělí na zóny podle zodpovědnosti, ale žádný z těchto způsobů není tak bezprostřední jako u diagramů interakce. Proto je výhodné diagram aktivit kombinovat s dalšími typy diagramů.

Největší uplatnění nacházejí diagramy aktivit ještě před tím, než se dostaneme k diagramům případů užití. Jde o fázi seznamování se s firemními procesy. Diagramy aktivit lze také chápat jako objektovou obdobu vývojových diagramů. [10]



Obr. 21 Výřez diagramu aktivit s ukázkou plavečkových drah.

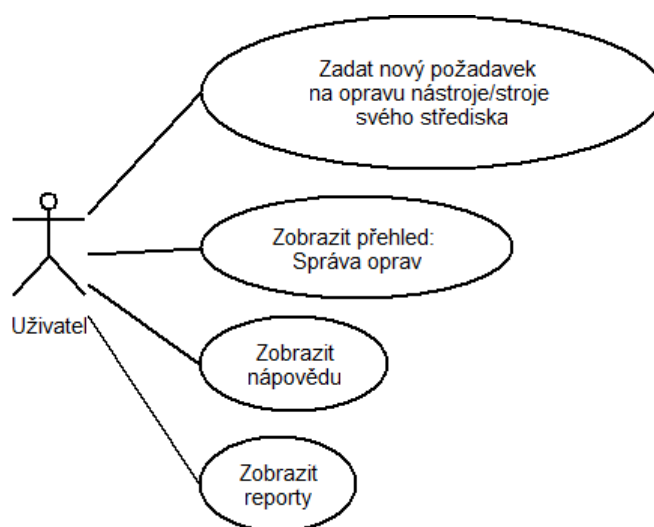
Diagram aktivit popisující proces, jímž procházejí nástroje a stroje, při jejich servisu není ve své podstatě diagramem aktivit úplně, neboť některé znázorněné akce nesplňují všechny podmínky kladené na jejich vlastnosti, jak bylo definováno výše. Zejména pak podmínku o nedělitelnosti. Je to dáno tím, že pro potřebný popis problematiky jsou popsány akce dostatečně podrobné a jejich detailnější rozbor nemá pro analýzu informačního systému význam.

Celý diagram (viz příloha číslo 3, strana VII) je rozdělen do 4 plavečkových drah, které přísluší hlavním aktérům servisních zásahů, tedy zadavateli požadavku, nástrojařům a seřizovačům. Na počátku je akce zadání nového požadavku a odtud se

diagram dělí na dvě základní větve podle toho, zda se jedná o servis nástroje či stroje. Poté proces prochází všemi významnými aktivitami své větve až do konce.

4.3.3 Diagram případů užití

V diagramech případu užití vystupují dva základní prvky. Zaprvé *aktér* neboli uživatel v rámci své komunikace se systémem a za druhé *případ užití* neboli reakce systému na akce vyvolané uživatelem (aktérem). Doplnujícími prvky diagramu jsou spojnice, tedy plné čáry spojující aktéra s případy užití symbolizující interakce mezi aktérem a případem užití a někdy také rámeček označující hranice systému. Na obrázku 22 je zobrazen jednoduchý diagram případu užití, kde postavička znázorňuje aktéra v roli „Uživatel“ a ovály s popisy jsou právě případy užití.

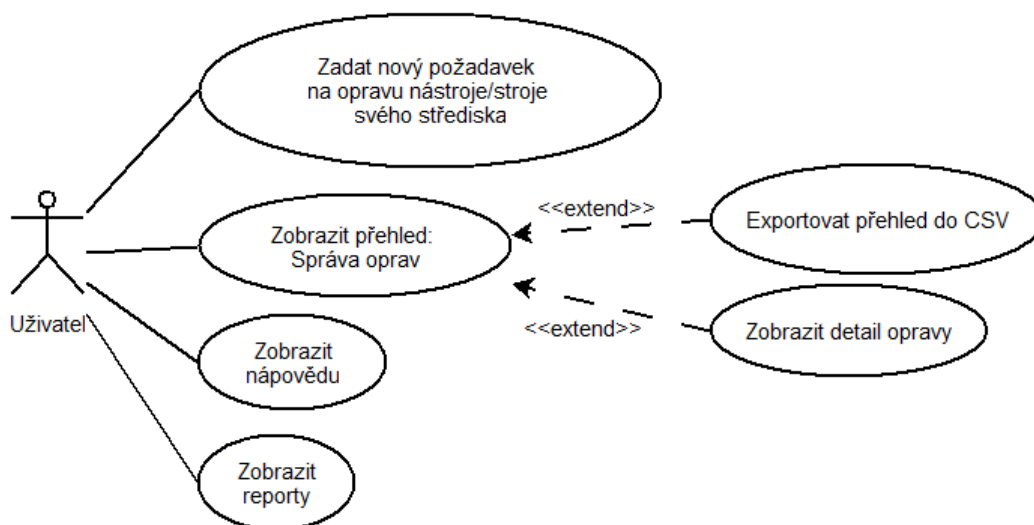


Obr. 22 Příklad základního diagramu případu užití.

Aktér je ve své podstatě role, v níž vystupují zaměstnanci vůči systému. Není výjimkou, že jeden zaměstnanec může vystupovat ve více rolích a stejně tak více rolí může mít přístup k jednomu a tomu samému případu užití. Aktér také nemusí být vždy fyzická osoba. Jako aktér může vystupovat i systémová událost, jako například časově spouštěné pravidelné vytváření reportů, rozesílání notifikací o změnách stavu systému a podobně.

Mezi jednotlivými případy užití mohou existovat dva typy relací. Typ relace `<<include>>` se objevuje tam, kde se vyskytuje podobná, nebo stejná část sekvence scénáře opakující se ve více případech užití. Proto se takováto část scénáře vyčlení do samostatného případu užití, který je poté s dalšími případy užití, které využívají jeho funkcionality spojen právě vazbou include. Druhým typem vazby je vazba `<<extend>>`. Představuje provázání případu užití s jiným, který nepovinně rozšiřuje jeho funkcionality, tudíž na rozdíl od relace include je původní případ užití zcela soběstačný. Příklad této vazby je zobrazen na obrázku číslo 23 na výřezu z diagramu užití pro aktéra s označením uživatel. Vazba include nebyla při návrhu systému využita, neboť zobrazené případy užití nebyly tak detailně rozebírány.

V problematice případů užití se vyskytuje ještě další aspekt, který se nazývá generalizace neboli dědičnost aktérů. Mohou nastat případy, kdy existují dva aktéři, kteří mají v základu totožné případy užití, a liší se pouze tím, že jeden z nich spouští například o jeden případ užití navíc. V takovém případě je možné uvažovat o zobecnění aktérů na dalšího aktéra, který bude spouštět případy užití společné pro oba původní aktéry. Většinou (ale ne vždy) bývá takový aktér abstraktním (v reálném světě neexistujícím). Tomuto procesu říkáme generalizace aktérů. [10] Tato technika však může představovat značné zkomplikování celého zápisu diagramu případů užití a dělá jej tak hůře čitelným pro běžné uživatele, pro které je primárně určen. Z tohoto důvodu se využití generalizace aktérů nedoporučuje.



Obr. 23 Příklad diagramu případu užití s vazbou extend

Pro systém Opravy nástrojů bylo identifikováno několik aktérů, které lze zároveň chápat jako úrovně uživatelských oprávnění. Diagramy případů užití pro jednotlivé aktéry jsou zobrazeny v příloze číslo 3, strany I - VI.

Aktéři:

- Uživatel
- OTK (Osoba Technické Kontroly)
- Nástrojař/Seřizovač
- Mistr
- Hlavní mistr
- Administrátor

V roli *Uživatele* vystupují prakticky úplně všichni uživatelé systému Opravy nástrojů. Účelem této role je zpřístupnit systém všem a současně umožnit řadovým zaměstnancům výrobních oddělení zadávat nové požadavky na servis. S rolí *OTK* (osoba technické kontroly) nebylo sice původně počítáno, ale nakonec byla do systému, na vyžádání zadavatele dodatečně přidána. Její specifikum oproti běžnému

uživateli je v tom, že může navíc ke všemu ještě přidávat poznámky k aktivním servisním požadavkům.

V roli *Nástrojař/Seřizovač* se nacházejí všichni zaměstnanci, kteří mají co dočinění s vlastním prováděním servisu, ať už v pozici toho, který předal nástroj ze stroje do nástrojárny, tak i přímého vykonavatele servisních zásahů.

V pozici aktéra *Mistr*, se nacházejí směnovní mistři všech tří výrobních oddělení. Této roli přísluší oprávnění měnit již zadané servisní požadavky. Stává se totiž, že po zadání požadavku, se například ukáže, že zadavatel udělal chybu v zadání, nebo při provádění servisu vyvstane potřeba provést jiné, nebo víc servisních zásahů. Schvalování těchto změn je zcela v kompetenci směnových mistrů.

Důležitou roli hraje pozice *Hlavního mistra*. Tento, kromě toho že má neomezené kompetence zasahovat do celého procesu servisu náležícímu jeho středisku, tak může již zadaný servisní požadavek odstranit z databáze. Také kromě obecně přístupných číselníků, jako jsou správa nástrojů a správa strojů, má přístup do úprav typů závad svého střediska.

Všechny dosavadní aktéry lze seřadit za sebe v pořadí, v jakém jsou uvedeni v seznamu výše. Každý další aktér má všechny akce shodné s předcházejícím a k tomu ještě vyvolává několik akcí navíc. Bylo zvažováno, zda tedy pro redukci objemu výsledných diagramů užití nevyužít metodu generalizace aktérů. Vzhledem však k faktu, že není příliš doporučována, ba spíše naopak, byl pro každého aktéra vytvořen vlastní, kompletní diagram případů užití.

Jediný aktér, který poněkud vybočuje z přímočarého schématu ostatních, je pozice označená jako *Administrátor*. V této roli vystupují zejména zaměstnanci vedení výroby, kteří nové požadavky na servis nezadávají, ale přesto potřebují mít přístup zejména k vyhodnocení získávaných dat. Pozice administrátora je také specifická tím, že může upravovat typy závad všech středisek a také měnit uživatelská oprávnění v rámci informačního systému.

4.3.4 Struktura aplikace

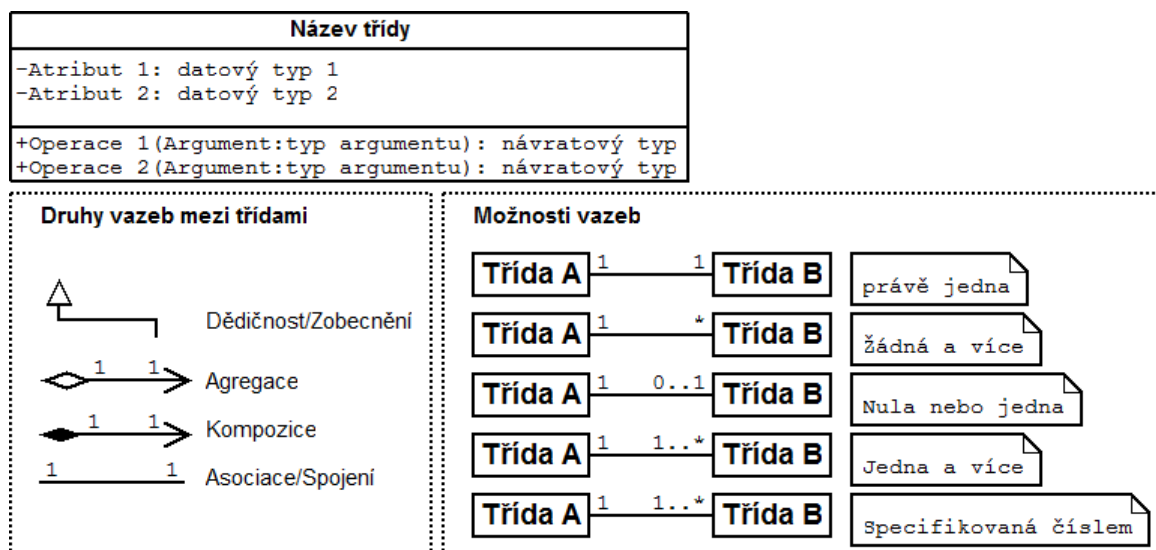
Pro zdokumentování struktury vznikající aplikace byla použita sémantika diagramů tříd. Svou podstatou se však nejedná zcela o diagram tříd podle standardů UML. Použitá forma byla zvolena pro větší názornost a přehlednost diagramu. Kompletní diagram viz příloha číslo 3, strana IX.

Základním prvkem diagramů tříd je blokový symbol znázorňující vlastní třídu. Obsahuje název třídy, atributy a operace třídy a další. V záhlaví třídy se také může nacházet informace o tom, zda je třída například abstraktní, či virtuální.

Mezi jednotlivými třídami mohou být různé druhy vazeb. Prvním typem vzájemného vztahu mezi třídami je vazba typu *agregace*. Tato vazba říká, že jedna třída je částí druhé, jako například motor je částí auta. Můžeme tedy určit, že vazba mezi objektovou třídou *Auto* a objektovou třídou *Motor* je typu agregace. Speciálním typem agregace je *kompozice*, tedy vazba, při níž víme, že podřízený objekt nemůže existovat samostatně bez nadřízeného.

K dalším symbolikám využitým v digramu struktury aplikace patří vazba znázorňující dědičnost mezi třídami, takzvanou generalizaci. Udává tedy spojení mezi rodičovskou třídou a třídou dědice.

U jednotlivých vazeb diagramu typu asociace se uvádí také násobnost, kolikrát třída A využívá třídu B, viz obrázek číslo 24 vpravo.



Obr. 24 Základní symboly diagramů tříd.

Diagram struktury aplikace je pro větší přehlednost rozdělen do tří hlavních stavebních bloků zahrnujících kontextově příbuzné třídy aplikace. Prvním a nejdůležitějším z nich je blok označený jako *Uživatelské prostředí*. Tento v sobě zahrnuje všechny formuláře, které jsou v aplikaci obsaženy. Kromě formulářů je do této skupiny zahrnut i třída *DTextBox* s atributem *datumCas*, která je dědicem třídy *TextBox*. Jedná se ve své podstatě o datumové textové pole, které bylo vytvořeno pro korektní zapisování a zobrazování datumových hodnot ve formulářích.

Ke skupině uživatelského prostředí náleží ještě blok označený jako *Ostatní*. Jedná se o třídy, poskytující formulářům uživatelského rozhraní potřebnou funkcionalitu. Třída *Uživatel* slouží pro založení a uchování aktuálně přihlášeného uživatele při spuštění modulu Opravy nástrojů. Nejdůležitější metodou této třídy je pak metoda *AccessRightTest*, která podle uživatelských oprávnění konkrétní osoby vrací hodnotu *True*, nebo *False* podle toho, zda má být přihlášenému uživateli umožněn přístup ke konkrétní akci v systému. Třída *ExportToCSV*, jak už název napovídá, zajišťuje převod dat do formátu .csv.

Třetím blokem diagramu je blok s označením *LINQ*. Tento slouží jako prostředník mezi uživatelským rozhraním a databází. Zahrnuje v sobě dva podbloky, jeden pro tabulky z databáze systému LVIS a druhý pro tabulky z databáze pojmenované jako Opravy nástrojů. Třídy v těchto podblocích slouží v tomto případě jako prostředníci mezi programovacím jazykem C# a dotazy jazyka SQL.

Mezi bloky jsou naznačeny symbolické vazby zastupující komplexní strukturu agregací a asociací, které mezi třídami existuje.

Čtvrtým blokem, který slouží spíše pro dokreslení situace je symbolické znázornění databázového serveru. Celý diagram lze částečně chápat jako takovou architekturu aplikace, i když také ne zcela v pravém slova smyslu.

5 Databázový server

Databázový server, plnící funkci centrálního datového úložiště, je jednou z nejdůležitějších částí informačního systému. Mezi základní, požadované vlastnosti datového úložiště patří zejména jeho dostupnost, nebo lépe spolehlivost. Databázový server by měl být vždy připraven přijmout či vydat požadované informace. Do spolehlivosti patří také odolnost datového úložiště vůči datovým ztrátám například z důvodu výskytu hardwarové, nebo softwarové poruchy serveru. Je tedy nezbytné, aby byl datový server vhodně zálohován.

Pro potřeby informačního systému servisního managementu byl jako databázový server zvolen *Microsoft SQL Server 2005 Express Edition*. Nejvýznamnějším aspektem při rozhodování bylo kritérium pocházející od zadavatele, tedy společnosti LEAR, tudíž minimalizovat náklady na jednotlivé prvky systému a to nejlépe využitím stávajících, ve firmě zavedených prostředků. Jelikož má firma svůj vlastní aplikační server, na němž kromě jiných běží i výše zmíněná serverová aplikace, pak se jednalo o téměř jasnou volbu.

Důležitou roli ve výběru hrála také vnitrofiremní politika. Vzhledem k celkové velikosti firmy zahrnující skoro dvě stovky poboček po celém světě, jsou zavedeny vnitrofiremní standardy využívaných softwarů. Každý nový software musí projít mnohdy zdlouhavým testovacím a schvalovacím procesem, na jehož konci může být také jeho zamítnutí. Obecně vzato, jsou v T&C divizi společnosti Lear corporation podporována řešení společnosti Microsoft a tak by jako možná alternativa přicházela v úvahu například novější verze zvoleného databázového serveru, tedy Microsoft SQL server 2008. Tato však nebyla zvolena zejména proto, že na stávající SQL server jsou navázány další aplikace, ale také proto, že pro potřeby informačního systému LVIS je současná verze zcela postačující.

Na závěr pro zachování databázového serveru od společnosti Microsoft také hovoří jeho snadná integrace s vývojovým prostředím Microsoft Visual Studio, v němž byla vytvářena klientská aplikace informačního systému.

Nutno ještě dodat, že aplikační server je umístěn v plně vybavené, klimatizované serverové místnosti, je napojen na dvojici záložních zdrojů a je denně zálohován. To zaručuje maximální možnou bezpečnost uchovávaných dat.

Jako možné alternativy od jiných firem by pak mohly přicházet v úvahu databázový server z dílny společnosti Oracle, případně MySQL od společnosti Sun.

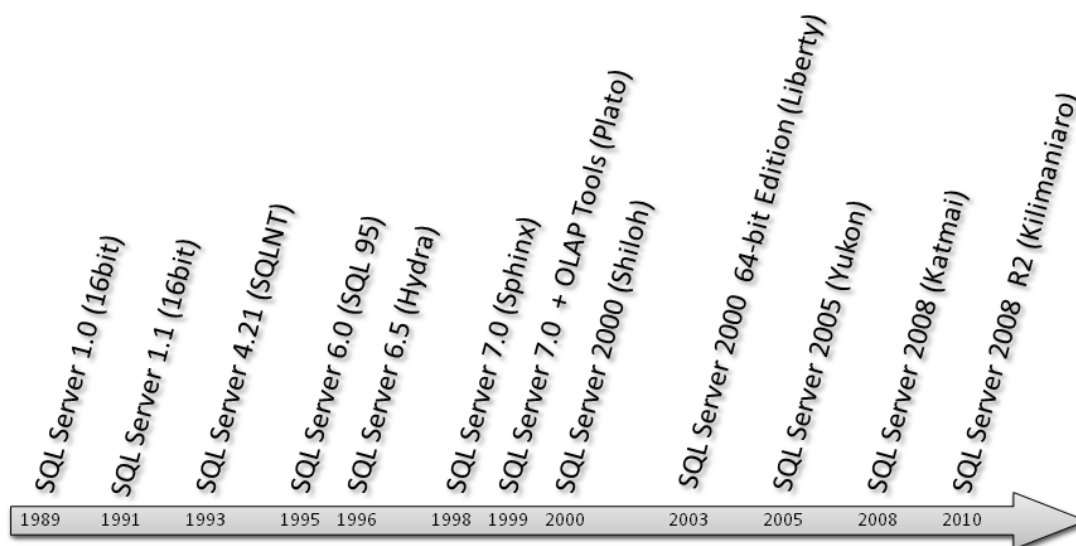
Microsoft SQL Server Express Edition je volně šiřitelná verze Microsoft SQL serveru se sníženou funkcionalitou. Není omezen počet databází, ani počet uživatelů serveru. Omezení této edice jsou dána maximální velikostí databázového souboru na hodnotu 4GB, využitím maximálně jednoho procesoru a maximálně 1GB operační paměti. Dále také chybí funkcionalita pro náročnější uživatele, jako například zrcadlení databázového serveru a podobně. Databáze jsou ukládány v jednotlivých souborech s příponou *.mdf*.

5.1 Historie Microsoft SQL serveru

SQL server 2005 byl uvolněn v říjnu roku 2005 jako nástupce serveru z roku 2000. Hlavní změnou oproti předchozí verzi bylo přidání nativní podpory správy XML (Extensible Markup Language) dat. Pro tento účel je definován XML datový typ,

který může být použit jako datový typ pro sloupce v tabulkách. Data v XML formátu jsou před vlastním uložením do databáze nejprve zkonvertována do interního binárního datového typu. V SQL serveru 2005 byly přidány další rozšíření jazyka TSQL (Transact SQL) umožňující vkládání XQuery dotazů, tedy dotazování právě na kolekce dat ve formátu XML. Navíc jsou zde také definovány nová rozšíření pro XQuery zvané XML DML umožňující modifikace XML dat. SQL server 2005 může být dostupný také přes web, využitím TDS (Tabular Data Stream) paketů zapouzdřených pomocí protokolu SOAP (Simple Object Access Protocol). Je-li k datům přistupováno přes web, pak jsou výsledky vráceny ve formátu XML. Pro relační data byl T-SQL doplněn o syntaxi pro zachytávání chyb (try/catch) a o podporu rekurzivních dotazů. SQL server byl také rozšířen o nový, indexovací algoritmus a kontrolní součty dat pro účinnější obnovu dat při systémové chybě.

První databázový server společnosti Microsoft byl uvolněn již v roce 1989 pod označením SQL server 1.0. V následujících letech prošel dlouhým vývojem, kdy se stále přizpůsoboval požadavkům aktuálních operačních systémů a potřebám společnosti vývojářů databázových aplikací až do současné podoby SQL Server 2008 R2 s obchodním názvem Kilimaniaro uvedeným v roce 2010.



Obr. 25 Časová osa vývoje Microsoft SQL Serveru.

5.2 Datový model

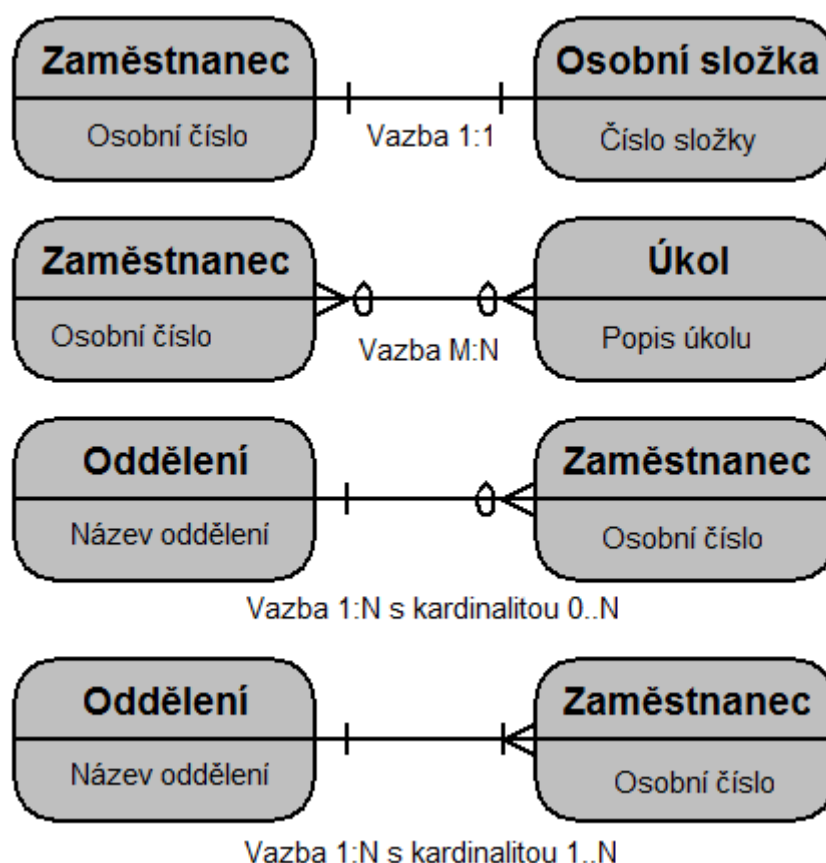
Pro potřeby návrhu informačního systému pro servisní management byl vytvořen logický, datový model. Hlavním specifikem logického, datového modelování je jeho nezávislost na budoucí implementaci. Model se skládá ze dvou základních elementů a to z entit, které lze také chápat jako tabulky budoucí databáze, které mají své atributy, tedy přeneseně sloupce těchto tabulek. Druhým elementem jsou pak vazby mezi entitami. Na obrázku číslo 26 jsou znázorněny příklady entit s atributy a typy vazeb mezi nimi.

Entita je charakterizována svým názvem a svými atributy, kterých může mít libovolné množství. Mezi entitami mohou být tři základní druhy vazeb. Prvním je

vazba typu 1:1, tedy entitě A náleží právě jedna entita B. Příkladem je vztah entity *Zaměstnanec* a jeho *Osobní složky* znázorněný na obrázku číslo 26, kdy každý zaměstnanec má právě jednu osobní složku a současně jedna složka náleží právě jednomu zaměstnanci. Vazba tohoto typu obvykle vede na spojení obou, takto svázaných, entit.

Druhým typem vazby na obrázku číslo 26 je vazba typu M:N, kdy jako příklad poslouží opět *Zaměstnanec*, kterému může být zadáno *N* úkolů. Stejně tak může být *Úkol* zadán pro *M* zaměstnanců.

Třetím základním typem vazby je vazba 1:N. Například tedy jedno oddělení firmy může mít až *N* zaměstnanců. Pro upřesnění vazeb typu 1:N a M:N slouží takzvaná kardinalita vazeb. Ta udává minimální počet entit na vícečetném konci vazby. Příkladem je třetí příklad vazby mezi entitami na obrázku níže. Opět jednomu oddělení ve firmě, může příslušet až *N* zaměstnanců, ale v extrémním případě i žádný. Z tohoto vyplývá kardinalita $0..N$, znázorněná také symbolikou na konci vazby. Čtvrtý příklad je obdobný s tím rozdílem, že je předepsána kardinalita vazby $1..N$, což znamená, že na každém firemním oddělení musí být nejméně jeden zaměstnanec.

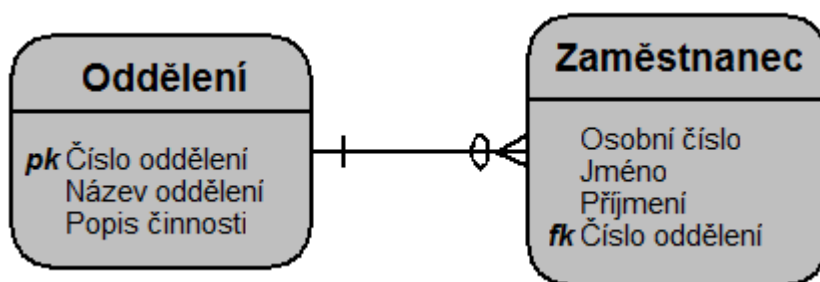


Obr. 26 Příklad entit s atributy a typy vazeb mezi entitami.

U atributů jednotlivých entit se mohou také vyskytovat symboly *pk* (primary key) pro primární a *fk* (foreign key) pro cizí klíč. Primární klíč je její atribut sloužící jako jedinečný identifikátor entity. Za primární klíč je vhodné volit číselný atribut,

nebo atribut obsahující krátké řetězce a to hlavně z pozdějších programátorských důvodů, zejména aby byl snadno a rychle porovnatelný. Primární klíč může být složen z více atributů. V takovém případě je označován jako složený primární klíč. Na obrázku číslo 27 je jako primární klíč entity oddělení označeno číslo oddělení, které je pro každé oddělení jedinečné.

Cizí klíč v entitě je atribut, který je v jiné entitě, s níž je tato entita provázána, primárním klíčem. Jak je tedy na obrázku níže znázorněno, v entitě Zaměstnanec je zaveden jako cizí klíč atribut *Číslo oddělení*, který je v entitě *Oddělení* primárním klíčem a určuje tak příslušnost zaměstnance ke konkrétnímu oddělení



Obr. 27 Příklad primárního a cizího klíče v logickém datovém modelu.

Entity datového modelu byly navrženy primárně v první normální formě, tedy jeden atribut obsahuje jedinou, atomickou informaci. V jednom případě byla použita nultá forma kdy se v jednom atributu může nacházet více informací. Na základě tohoto modelu byla následně vytvořena struktura tabulek databáze výsledné aplikace. Kompletní datový model je předmětem přílohy číslo 4.

6 Klientská aplikace

Nejdůležitější částí informačního systému je právě klientská aplikace. Slouží jako prostředník mezi člověkem a daty. Jejím hlavním úkolem je minimalizovat vliv lidského faktoru na sbíraná data a to primárně zabráněním neomezeného přístupu k těmto datům. Dále také účinnou verifikací dat (informací) vstupujících do systému.

Klientská aplikace systému LVIS byla vytvořena ve vývojovém prostředí *Microsoft Visual Studio 2008 Professional*, na platformě *.Net Framework* v současné verzi 3.5, přičemž jako programovací jazyk byl použit C#. Volba programovacího prostředí vycházela v první řadě z požadavku zadavatele, vytvořit takzvaně tlustou, neboli „klasickou“ WinForm (formulářová aplikace pro operační systém Windows), klientskou aplikaci v podobě programu, který bude nainstalován na každé pracovní stanici ve výrobních halách, případně v kancelářích. Dalším důvodem pro zvolení tohoto programovacího prostředí, byl fakt, že je již ve Vyškovském závodě používáno.

Jako programovací metodika bylo zvoleno programování do šířky a to hlavně pro to, že vedení, tedy zadavatel, chtělo brzo vidět výsledky, a tudíž se tato metoda jevila pro tento účel jako nejvhodnější. Myšlenka programování do šířky spočívá ve vytvoření těl všech základních stavebních bloků aplikace s deklaracemi základních metod pro interakci mezi bloky. Pod pojmem stavební blok lze chápat veškeré formuláře, třídy, nebo skupiny tříd tvořících společně danou funkcionalitu a podobně, z nichž se výsledná aplikace skládá. Programováním do šířky je během krátké doby možné předložit náhled toho, jak bude aplikace ve výsledku vypadat případně i chovat, i když tedy hlavně z venku. Jde ve své podstatě o takový skelet programu, který je však následně nutné vyplnit veškerou požadovanou funkcionalitou. Další výhodou tohoto postupu je také možnost průběžného testování komplexního chování celé aplikace.

Alternativou k programování do šířky je programování do hloubky, kdy jsou postupně brány stavební bloky programu jeden po druhém, přičemž je v každém bloku vytvořena veškerá funkcionalita, kterou daný blok má mít a poté se teprve přistoupí k tvorbě dalšího. Hlavní výhodou tohoto přístupu je mnohem snadnější organizace práce v týmu vývojářů, kdy se může každý zcela soustředit na svou část aplikace bez vlivu ostatních vývojářů. Nevýhodou ale zůstává, že není možné demonstrovat chování celé aplikace, dokud není kód již téměř dokončen.

Jako prostředník mezi formuláři aplikace a databází slouží model LINQ (Language Integrated Query). LINQ umožňuje dotazovat se na libovolný druh dat a pracovat s nimi v konzistentním modelu, nezávisle na datovém zdroji. [12] V případě systému LVIS byl LINQ využit v první řadě jako nástroj pro začlenění SQL dotazů do programového kódu.

Součástí aplikace je také kompletní nápověda popisující všechny hlavní oblasti systému a činnosti, které lze v systému provádět. Její zobrazení je možné kdykoliv během práce v aplikaci provést stisknutím klávesy *F1*.

6.1 Systém LVIS

Po spuštění klientské aplikace, se jako první objeví úvodní obrazovka informačního systému LVIS s přihlašovací nabídkou. Po přihlášení je zobrazena nabídka čtyř hlavních položek, přičemž první zobrazí správu uživatel systému LVIS

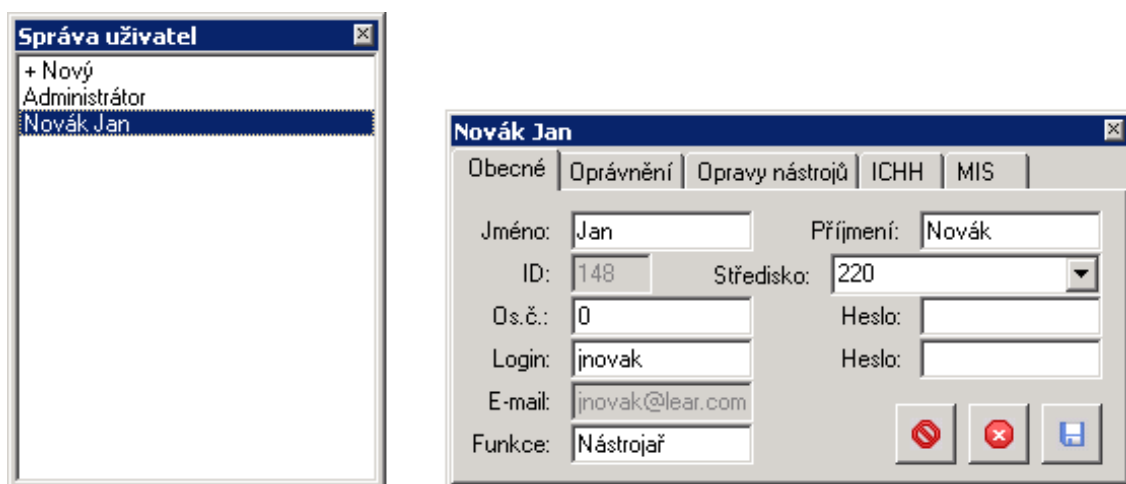
a zbývající tři slouží ke spuštění jednotlivých modulů informačního systému. Při kliknutí na kterýkoliv z těchto odkazů se hlavní okno systému LVIS minimalizuje do tray oblasti, odkud je možné jej opětovně otevřít poklepáním levým tlačítkem myši na ikonu, nebo přes kontextovou nabídku při kliknutí pravým tlačítkem. Tento způsob minimalizace byl zvolen z toho důvodu, že většina uživatel má v průběhu dne obvykle otevřeno velké množství dalších oken, které tak zcela zabírají základní panel systému Windows, přičemž systém LVIS bývá často spuštěn celý den a tak, aby „nezabíral“ místo, je „schován“ právě v tray oblasti u hodin, kde nepřekáží.

6.2 Správa uživatel

Kliknutím na první položku hlavního menu s označením *Správa uživatel* dojde ke spuštění formuláře se seznamem všech uživatel systému LVIS. Tento může být zobrazen dvojím způsobem. Pokud k němu přistupuje uživatel s oprávněním *LVIS Admin*, pak má zobrazen úplný seznam všech uživatel, včetně položky pro přidání nového uživatele. Nemá-li přihlášený uživatel toto oprávnění, pak může vidět v seznamu jen sebe.

Poklepáním na zvoleného uživatele, či stisknutím klávesy *Enter* dojde k otevření detailu tohoto uživatele. Uživatel v pozici správce aplikace LVIS může měnit většinu údajů v otevřeném detailu uživatele, jako například heslo, jméno, příjmení a podobně. Běžný uživatel může v náhledu svého vlastního uživatelského profilu měnit pouze své heslo.

Přístupová hesla všech uživatel systému LVIS jsou zašifrována pro zvýšení bezpečnosti. Není totiž žádoucí, aby byla uživatelská hesla snadno čitelná. Nežádá se totiž stává, že uživatelé, navzdory bezpečnostním doporučením, používají jediné heslo pro přístup do různých systémů. I když je nepravděpodobné, že by se zrovna do informačního systému LVIS pokoušel někdo „nabourat“ pod jiným účtem než pod svým vlastním, neboť obsažená data nejsou tak citlivá, jako například u personálních informačních systémů. Právě ale kvůli výše zmíněnému faktu o používání jednotného hesla do více systémů je lépe nevytvářet ideální podmínky pro možné útočníky.



Obr. 28 Formuláře správy uživatel.

V náhledu uživatelského profilu v záložce *Obecné* jsou základní informace o uživateli. V pravém dolním rohu se nachází trojice tlačítek, kdy prvním dojde k odstranění uživatele. Odstraněním uživatele však nedojde k jeho vymazání z databáze, ale pouze k jeho označení jako odstraněného. K tomuto řešení bylo přistoupeno z důvodu zachování veškerých informací v servisních záznamech i po odchodu zaměstnanců z firmy. Druhé a třetí tlačítko je pak obsaženo i v dalších záložkách uživatelského profilu, přičemž první slouží k uzavření formuláře a druhé k uložení změn v uživatelském profilu.

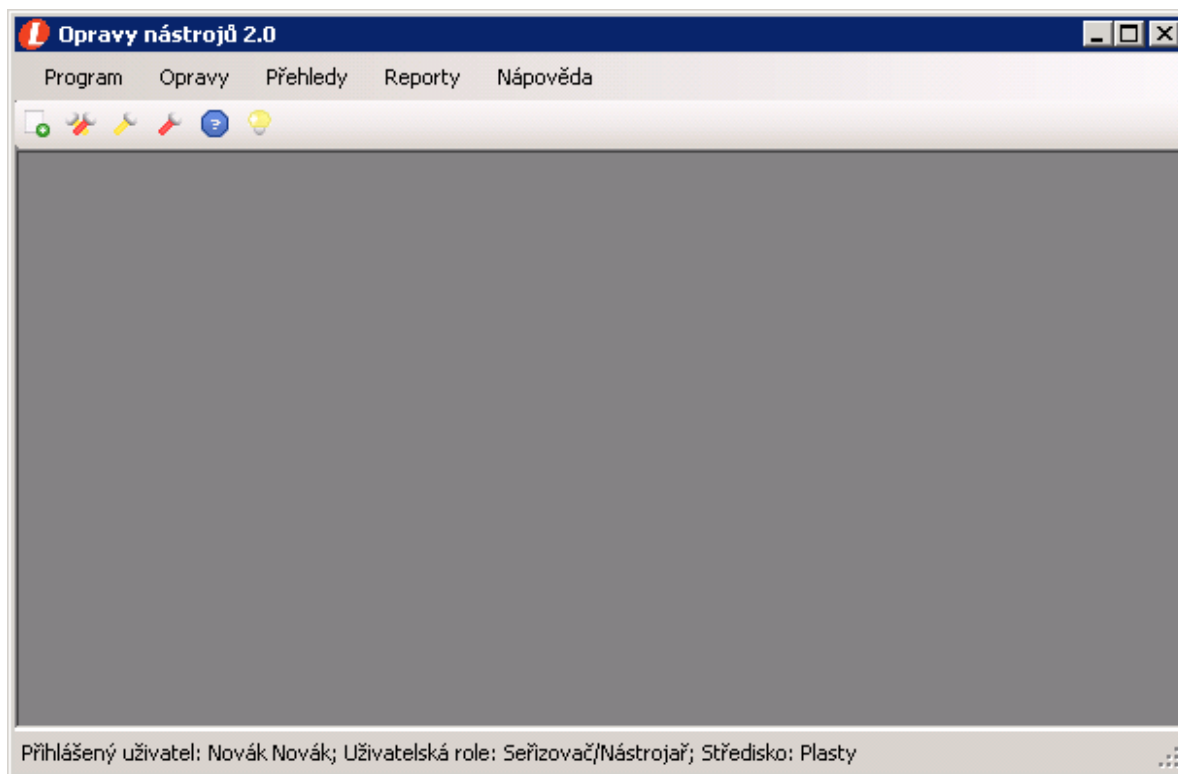
V dalších záložkách uživatelského profilu jsou obsaženy volby pro detaily přístupových oprávnění k jednotlivým modulům informačního systému.

Pro přidání nového uživatele slouží první položka v seznamu *Správy uživatel* označená jako *+Nový*. Při volbě této položky se zobrazí nevyplněný profil připravený pro zadání všech požadovaných informací. Při nevyplnění hesla je výchozí heslo stejné jako uživatelské jméno.

Pro potřeby testovací verze systému LVIS jsou zavedeny dva uživatelské účty, kdy jeden slouží jako administrátor se všemi oprávněními a druhý jako běžný uživatel. Práva lze samozřejmě pomocí administrátorského účtu kdykoliv měnit. Každý z těchto uživatelských účtů má v základu přihlašovací jméno shodné s heslem. Pro administrátora je to *admin*, pro běžného uživatele pak *jnovak*.

6.3 Hlavní okno aplikace Opravy nástrojů 2.0

Prvním z výše zmíněných modulů informačního LVIS je právě aplikace pro servisní management pojmenovaná *Opravy nástrojů 2.0*.



Obr. 29 Hlavní okno aplikace *Opravy nástrojů 2.0*.

V horní části formuláře se nachází hlavní roletová nabídka, sloužící jako primární ovládací prvek celé aplikace. Odtud lze spustit veškeré další podformuláře. Pod touto nabídkou se nachází tlačítkový panel s vybranými operacemi sloužící pro rychlý přístup k nejčastěji využívaným funkcím systému. Na spodním okraji okna se pak nachází stavový panel, zobrazující informace o aktuálně přihlášeném uživateli. Mezi panelem s rychlou volbou a stavovým panelem se pak nachází prostor pro zobrazování zmíněných podformulářů.

6.4 Zadání nového servisního požadavku

Pro zadání nového požadavku na opravu (servis) slouží nabídka hlavního menu *Opravy* -> *Nová*. Tato, podle oprávnění přihlášeného uživatele, spustí formulář pro zadání nového požadavku na servis nástroje, nebo stroje. Má-li uživatel povoleny obě tyto volby, tedy zadat servisní požadavek na nástroj i stroj, pak je zobrazen formulář, jako rozcestník, pro volbu dalšího postupu. Po vyplnění všech povinných, oranžově podbarvených polí uživatel klikne na tlačítko *Uložit požadavek*. Pokud byla všechna pole správně vyplněna, pak je zobrazeno oznámení o úspěšném zařazení požadavku do databáze a formulář je uzavřen. Pokud ne, pak je uživatel upozorněn a vrácen zpět k opravě svého zadání. Při zaškrtnutí pole *Tisk* je po úspěšném uložení vygenerována tisková sestava s detaily nově zadaného servisního požadavku. Tuto pak lze vytisknout, nebo exportovat do .pdf či .xls (Excelu) podle aktuální potřeby.

Obr. 30 Formulář pro zadání nového servisního požadavku.

6.5 Správa oprav

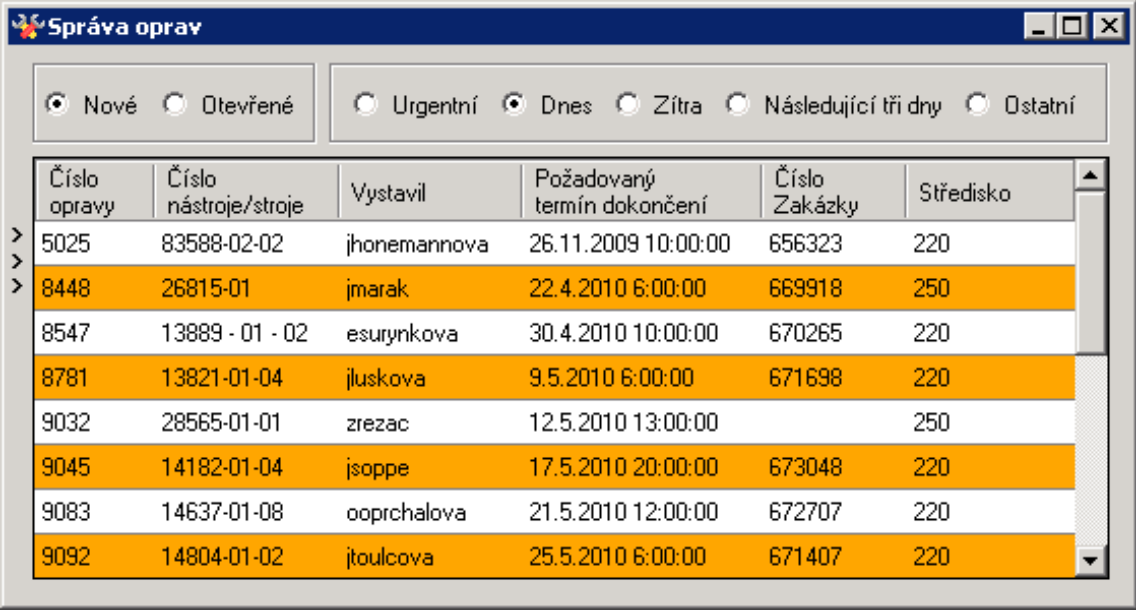
Další volbou v položce *Opravy* v hlavním menu je volba *Správa oprav*. Tento formulář slouží jako centrální místo pro souhrnné zobrazení všech aktivních,

servisních požadavků rozdělených do skupin podle urgency a požadovaného termínu dokončení pro snadnější určení prioritních servisních požadavků.

Při spuštění této nabídky jsou nejprve zobrazeny nově zadané požadavky, u nichž byla zaškrtnuta volba *Urgentní*, protože jim je třeba věnovat největší pozornost. Z toho také vyplývá červené podbarvení řádků tabulky, které ještě více dodává na důležitosti zobrazeným položkám. V horní části formuláře se pak nacházejí dvě sady voličů sloužících pro specifikaci zobrazeného výběru servisních požadavků. V první sadě uživatel volí, zda si přeje zobrazit nově zadané servisní požadavky, nebo požadavky *Otevřené*, tedy takové, u nichž byly již servisní práce zahájeny. Pro volbu *Nové* je aktivní i druhá sada voličů sloužící právě pro efektivnější určení prioritních servisních požadavků. Kromě již zmíněné volby *Urgentní*, se zde tedy nacházejí volby *Dnes*, což zahrnuje všechny požadavky, které mají datum požadovaného dokončení totožný s aktuálním kalendářním datem, nebo měly být splněny již dříve. Obdobně fungují volby *Zítřa*, *Následující tři dny* a *Ostatní*.

Na formuláři se na jeho levém okraji také nachází nenápadné tlačítko označené třemi symboly „>“. Po kliknutí na něj se zobrazí nabídka pro volitelné nastavení zobrazených sloupců přehledové tabulky servisních zásahů. Součástí této nabídky je i tlačítko sloužící pro export aktuálního zobrazení tabulky do formátu .csv. Opětovným kliknutím na postranní tlačítko se tato nabídka opět skryje.

Nejdůležitějším aspektem formuláře „*správa oprav*“ je funkce, kdy dvojitým kliknutím na položku servisního požadavku v přehledové tabulce se spustí formulář *Náhled opravy* s veškerými detaily zvolené položky.



Číslo opravy	Číslo nástroje/stroje	Vystavil	Požadovaný termín dokončení	Číslo Zakázky	Středisko
> 5025	83588-02-02	jhonemannova	26.11.2009 10:00:00	656323	220
> 8448	26815-01	jmarak	22.4.2010 6:00:00	669918	250
8547	13889 - 01 - 02	esurynkova	30.4.2010 10:00:00	670265	220
8781	13821-01-04	juskova	9.5.2010 6:00:00	671698	220
9032	28565-01-01	zrezac	12.5.2010 13:00:00		250
9045	14182-01-04	jsoppe	17.5.2010 20:00:00	673048	220
9083	14637-01-08	ooprchalova	21.5.2010 12:00:00	672707	220
9092	14804-01-02	jtolcova	25.5.2010 6:00:00	671407	220

Obr. 31 Formulář správy oprav.

6.6 Náhled opravy

Tento formulář neslouží jen jako detailní náhled všech informací, které jsou ke konkrétnímu servisnímu požadavku k dispozici, ale také pro provádění veškerých možných operací s konkrétním servisním požadavkem. Podle uživatelských oprávnění a statusu zobrazeného servisního požadavku se liší zpřístupnění

jednotlivých ovládacích prvků, například nelze uzavřít opravu, která ještě nebyla ani započata a podobně.

V pravém dolním rohu formuláře se nachází dvě tlačítka, která jsou přístupná vždy a všem. První z nich slouží opět k vygenerování aktuální, tiskové sestavy zobrazeného servisního požadavku a druhé slouží k uzavření formuláře. Nalevo od těchto tlačítek se nachází v základu čtyři další, jejichž dostupnost je již závislá právě na statusu servisního požadavku a uživatelských oprávněních.

Obr. 32 Náhled detailu otevřené opravy s oprávněním hlavní mistr.

První řada těchto tlačítek je přístupná primárně uživatelům, patřícím do skupiny nástrojař, nebo seřizovač, a jejichž středisko je shodné se střediskem servisního požadavku. Tlačítkem *Začít opravu* je, po potvrzení dialogu, status aktuální opravenky změněn z „Nová“ na „Otevřená“. Hned vedle se nachází tlačítko s označením *Uzavřít opravu*. Je-li zobrazený požadavek „otevřený“, pak je přístupné. Kliknutím na toto tlačítko je zpřístupněno textové pole, označené jako *Popis odstranění závady* kam uživatel vyplní poznámky k dokončenému servisu a poté klikne na nově zobrazené tlačítko pod tímto polem s označením *Uzavřít*. Zadaný popis je uložen do databáze a status servisu je změněn na *uzavřený*. Kliknutím na tlačítko *Zrušit* je formulář vrácen do původního stavu a uzavírání opravy je

přerušeno. Tyto ovládací prvky slouží k obsluze základních operací, které lze se zadanými servisními zásahy provádět.

Tlačítka druhé řady přísluší vyšším instancím uživatelských oprávnění, zejména mistrům a hlavním mistrům. Patří sem tlačítko *Upravit opravu*. Toto zpřístupní vybraná pole formuláře pro provádění změn, jako například seznam „*Závady*“ vlevo uprostřed formuláře zobrazeném na obrázku číslo 32, kam lze pomocí kontextové nabídky, vyvolané pravým tlačítkem myši, přidávat nové typy závad, eventuálně aktuální odebírat. Dále lze změnit požadovaný termín dokončení a podobně. Po provedení všech požadovaných úprav uživatel klikne na tlačítko „*Uložit změny*“ a po potvrzení dialogu jsou změny uloženy do databáze. Posledním tlačítkem ve spodní části náhledu opravy je „*Odstranit opravu*“, které jak už název napovídá, vymaže aktuální servisní záznam z databáze. Toto slouží zejména pro případy, kdy je servisní požadavek zadán pro chybně zvolený nástroj a podobně. Toto je v kompetenci hlavních mistrů.

U otevřeného servisního požadavku, jsou na tomto formuláři k dispozici ještě další ovládací prvky. Jedním z nich jsou tlačítka pro přidání délky a ceny opravy. Kliknutím na ně se příslušné textové zpřístupní pro zápis. Do něj pak uživatel zapíše číselný údaj o hodinách či částce v korunách, kterou chce k servisnímu požadavku připočíst a znovu klikne na tlačítko vedle textového pole. Tímto se hodnota připočte ke stávající. V případě přidávání hodin je ještě zobrazen dialog, zda si uživatel přeje připočíst částku $150 \text{ CZK/h} \cdot \text{počet zadaných hodin}$, což je sazba stanovená jako paušální hodinová sazba za práci zaměstnance provádějícího servis.

Na pravém okraji formuláře, uprostřed se nachází, podobně jako u *Správy oprav* (viz kapitola 6.5), nenápadné tlačítko značené třemi symboly „<“. Kliknutím na něj se na pozici seznamu *Poznámek během opravy/přestavby* zobrazí panel pro zadání nové poznámky. Do textového pole uživatel vyplní text své poznámky o maximální délce 1500 znaků a kliknutím na tlačítko „*Přidat*“ a potvrzení dialogového okna, uloží záznam do databáze. Panel pro zadání poznámky se poté opět skryje.

6.7 Správa strojů

Prostřednictvím hlavní nabídky přes *Program -> Spravovat -> Stroje*, případně pomocí ikony na panelu se spouští formulář pro správu strojů. Zobrazení se opět liší podle uživatelských oprávnění. Formulář je přístupný všem v režimu náhled, tedy uživateli se zobrazují informace o jednotlivých strojích, ale bez možnosti tyto informace jakkoliv modifikovat. Druhý typ zobrazení je v režimu úpravy, kdy lze jednotlivé stroje do databáze přidávat, odebírat či je měnit.

Odebírání strojů probíhá obdobně jako ve správě uživatel. Záznam není tedy z databáze smazán, ale jen označen jako odstraněný a tudíž už není dostupný při zadávání nových servisních požadavků, ale přitom je stále k dispozici pro historii uzavřených servisních zásahů provedených na tomto stroji.

Přidání nového stroje se provádí tak, že se v režimu úpravy vybere první položka seznamu pojmenovaná + *Nový*, čímž se formulář vyprázdní a je připraven pro zadání nového stroje. Poté kliknutím na tlačítko *Uložit* a potvrzení dialogového okna, je nový stroj přidán do databáze.

6.8 Správa nástrojů

Jedná se o obdobnou problematiku jako ve správě strojů, s tím rozdílem, že místo přidávání, odebírání a úprav strojů se zde pracuje s nástroji. Rovněž má formulář dva módy zobrazení, jeden pro prohlížení a jeden pro úpravy. Odstranění i přidání záznamu probíhá také stejným způsobem jako ve správě strojů, tedy odstraněný nástroj je pouze označen jako odstraněný. Za zmínku zde stojí pole označené jako „Díly“. Jelikož lze na mnoha nástrojích vyrábět několik různých druhů výrobků, tak jsou zde uvedeny jejich čísla.

The image shows two screenshots of software windows for tool and machine management.

Left Window: Správa nástrojů

- Tool List:** A list of tools with IDs like 00535-02-02, 12905-01-04, 13007-02-01, etc.
- Form Fields:**
 - Číslo nástroje: 00535-02-02
 - Středisko: 220
 - Stav nástroje: Oprava/přestavba
 - Stav zakázky: Zakázka se zpracov
 - Číslo aktuálního dílu: 02679111007
 - Díly: 00535111007, 02679111007
- Umístění nástroje:**
 - Stroj: 80
 - Sklad: (empty)
- Buttons:** Odstranit..., Uložit..., Zavřít...

Right Window: Správa strojů

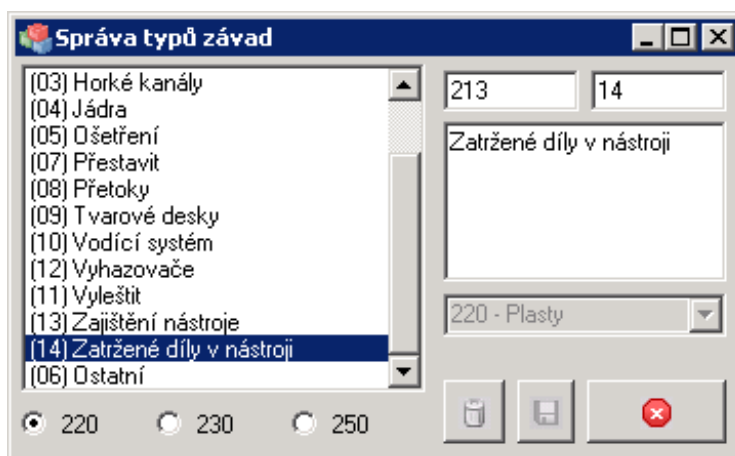
- Machine List:** A list of machines with IDs like 46, 47, 50, 52, 525, 56, 573, 574, 59, 61, 63, 80.
- Form Fields:**
 - Číslo stroje: DHM 260038
 - Středisko: 220
 - Číslo APL: 80
 - Typ stroje: Vstřikovací lis
- Buttons:** Odstranit..., Uložit..., Zavřít...

Obr. 33 Formuláře správy nástrojů a správy strojů.

6.9 Správa typů závad

Při zadávání každého servisního požadavku, uživatel povinně vybírá ze skupiny typů závad vždy nejméně jeden. Tyto typy závad jsou v podstatě zobecněním všech obvyklých druhů servisních požadavků jednotlivých středisek. Lze je editovat právě ve *Správě typů závad*. Tento formulář se spouští opět z hlavní nabídky *Program -> Spravovat*. Formulář je dostupný primárně pro uživatele skupiny hlavní mistr a administrátor. Úplně stejně jako u předchozích číselníků je možné typy závad přidávat, upravovat a také odstraňovat. Opět také podle uživatelských oprávnění jsou přihlášenému uživateli dostupné všechny typy závad, nebo jen závady jednoho konkrétního střediska.

Pro zadání nového typu závady, uživatel vybere první položku seznamu v levé polovině formuláře s označením + *Nový*. Textová pole v pravé části formuláře jsou poté zpřístupněna pro zápis. Uživatel vyplní pole pro popis závady, nacházející se uprostřed a z roletového menu pod ním zvolí středisko, jemuž bude nový typ závady náležet. Tato akce vyvolá automatické vygenerování identifikátorů závady. Poté je kliknutím na tlačítko *Uložit změny* nový typ závady přidán do databáze. Každý typ závady má svoje číslo závady, což je jeho jednoznačný identifikátor pro uložení v databázi a také údaj *ID*. Jedná se o zažítý systém označení sloužící k usnadnění identifikace typu závady při komunikaci mezi zaměstnanci.



Obr. 34 Formulář správy typů závad

Odstranění probíhá zvolením typu závady ze seznamu, kliknutím na tlačítko *Odstranit* a po potvrzení následného dialogu, je zvolená položka označena jako odstraněná a není tudíž již dostupná při zadávání nového servisního požadavku.

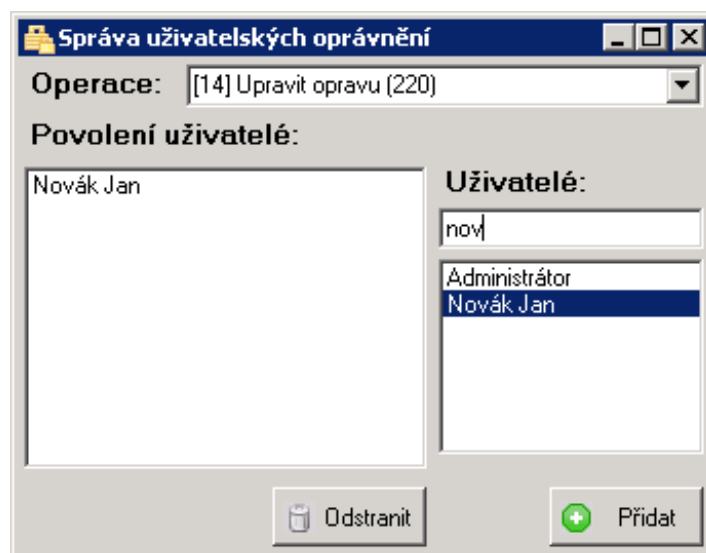
6.10 Správa uživatelských oprávnění

Z důvodu nejednoznačnosti kompetencí jednotlivých zaměstnanců firmy a tudíž množství dodatečných požadavků na dodatečné změny přístupových oprávnění, byl do druhé verze Oprav nástrojů zakomponován variabilní systém uživatelských oprávnění. Vybraným akcím, jako například odstranění servisního požadavku, či zadání nové poznámky byly přiděleny číselné identifikátory. Byla také zavedena tabulka uživatelských oprávnění (viz příloha číslo 4 - Datový model) kde jsou každé operaci přiděleny minimální požadavky na úroveň oprávnění uživatele. Při spuštění těchto akcí jsou pak porovnávány požadavky operace s uživatelským profilem přihlášeného uživatele. Tento způsob uložení uživatelských oprávnění se jeví jako nejschůdnější vzhledem k faktu, že je třeba centrálně uchovávat přístupová práva všech uživatel, ať už jsou přihlášení odkudkoliv. Významnou roli při volbě hrál i fakt, že ne-každý zaměstnanec má založený svůj vlastní uživatelský účet pro přihlášení do systému Windows a tak mnohdy i několik uživatel sdílí počítače, na hale, na nichž je obvykle přihlášen směnový mistr, nebo jiná zodpovědná osoba, která má uživatelský účet do systému Windows.

Úrovně uživatelských oprávnění lze rozdělit na dvě části. První z nich jsou uživatelské skupiny v podobě uživatelských rolí, které jsou uživatelům přiděleny

podle jejich pracovní pozice ve správě uživatel systému LVIS. Kompletní přehled těchto oprávnění je znázorněn tabulkou v příloze číslo 5. Oprávnění těchto skupin vychází z diagramů případu užití. Editace této části uživatelských oprávnění není pro uživatele systému LVIS k dispozici. Jako druhou část lze brát individuální oprávnění jednotlivých uživatel, tedy každému uživateli systému LVIS s povoleným přístupem do modulu Opravy nástrojů lze přidělit nebo odebrat oprávnění ke konkrétní operaci. K tomuto účelu slouží právě *Správa uživatelských oprávnění*.

Na obrázku číslo 35 je náhled provedení správy uživatelských oprávnění. Tato se spouští z hlavní nabídky v menu *Program -> Nastavení*. V horní části formuláře je roletové menu pro výběr operace, u níž chceme přidat, nebo odebrat oprávnění uživatelů. Číslo v hranatých závorkách na začátku popisu operace je právě její neměnný identifikátor. V levé části se pak nachází výpis všech uživatel, kteří mají navíc k oprávněním uživatelské role, k níž přísluší, povolený přístup k aktuálně vybrané operaci. V pravé části se pak nachází seznam všech uživatel s povoleným přístupem do systému opravy nástrojů s textovým polem pro vyhledávání.



Obr. 35 Formulář správy uživatelských oprávnění.

Zvolením uživatele v pravém seznamu a kliknutím na tlačítko *Přidat*, nebo stiskem klávesy *Enter* je vybraný uživatel přidán do seznamu povolených uživatel zvolené operace. Naopak pro odstranění uživatele se seznamu povolených, jej stačí v tomto seznamu označit a kliknout na tlačítko *Odstranit*.

6.11 Přehled uzavřených oprav

Jako nástroj pro procházení uzavřených, tedy již neaktivních oprav slouží *Přehled uzavřených oprav*. Součástí tohoto formuláře jsou i nástroje pro filtrování záznamů podle období, kdy byly uzavřeny, případně podle osoby kterou byly uzavřeny. Dalšími filtry jsou volba čísla nástroje či stroje, na němž byl servis prováděn a nakonec zda byl servisní požadavek dokončen před jeho požadovaným termínem dokončení či nikoliv. Veškeré tyto filtry lze podle potřeby mezi sebou kombinovat. Panel filtrů se nachází v horní části tohoto formuláře. Pod ním se nacházejí jednoduché statistické údaje o aktuálně zobrazeném výběru a dále vlastní

záznamy uzavřených servisních zásahů. Obdobně jako ve *Správě oprav*, lze i zde poklepnutím na libovolnou položku ze zobrazeného seznamu spustit formulář s jejím, detailním náhledem. Rovněž také analogicky se na levém okraji formuláře nachází tlačítko označené třemi symboly ">" které opět skrývá seznam zobrazitelných sloupců tabulky uzavřených oprav a tlačítko pro export aktuálního zobrazení do formátu csv.

Uzavřené opravy

☒ ☒

Od: 01.03.2010 00:00 Do: 31.03.2010 23:59

Uzavřel: [vypustit] Uzavřeno po termínu ☒

(Ná)stroj: 14461-01-02

Statistiky:

		Celkem	Průměr	Min	Max	
Počet položek:	8	Cena opravy: 3300	412,5	300	900	[CZK]
		Delka opravy: 22	2,75	2	6	[hod]

Číslo opravy	Číslo nástroje/stroje	Vystavil	Požadovaný termín dokončení	Popis odstranění závady
7151	14461-01-02	bpavlovicova	27.2.2010 13:00:00	Změna přestavby.
> 7282	14461-01-02	ooprchalova	4.3.2010 12:30:00	02 04 05 06 07
> 7310	14461-01-02	jluskova	5.3.2010 19:00:00	Vyčištěno a přestavěn...
7336	14461-01-02	jzouharova	6.3.2010 19:30:00	05,07-prestavba na díl ...
7361	14461-01-02	jhonemannova	8.3.2010 3:30:00	05,07
7554	14461-01-02	jhonemannova	16.3.2010 4:00:00	Vyčištěno a přestavěn...
7688	14461-01-02	jtoulcova	20.3.2010 20:00:00	05,07

Obr. 36 Náhled formuláře s přehledem uzavřených servisních zásahů.

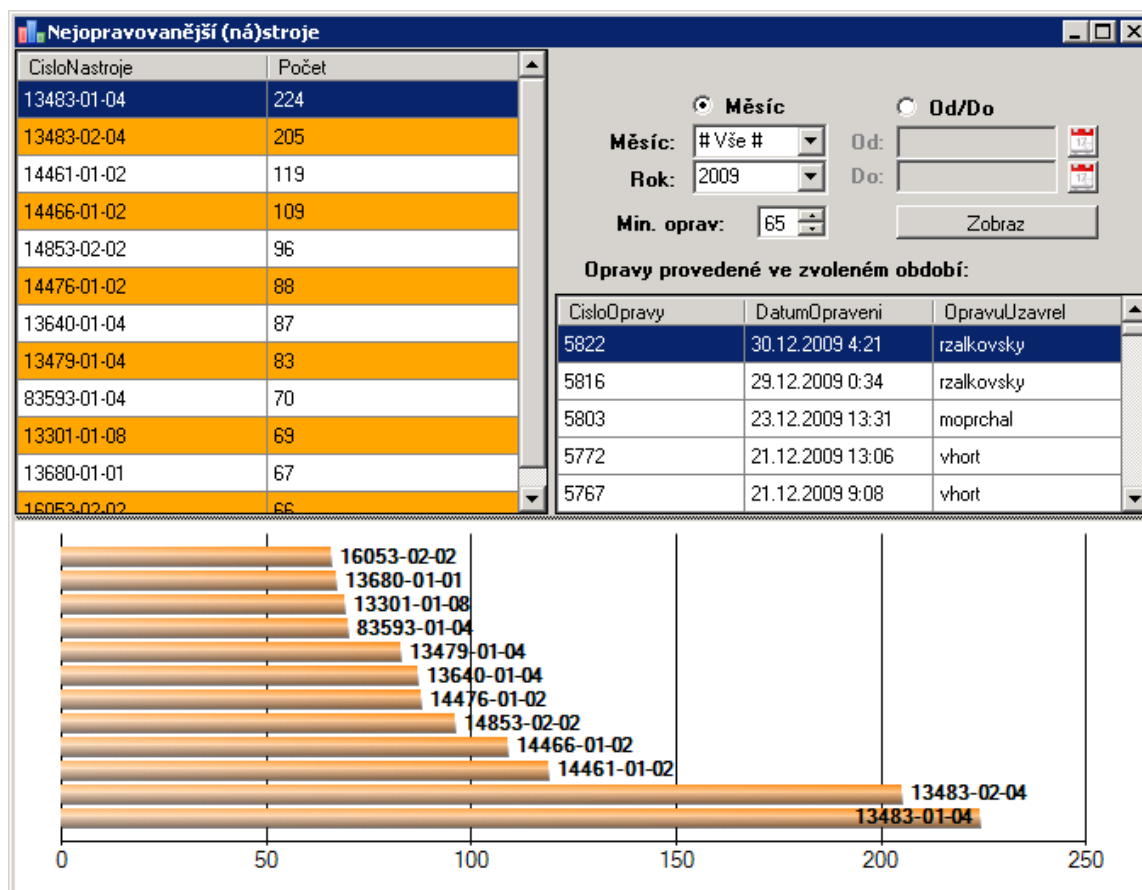
Tento formulář má využití zejména při zpětném vyhledávání v uzavřených opravách například z důvodu ať už zákaznické, nebo vnitrofiremní reklamace. Umožňuje snadné a rychlé nalezení všech servisních zásahů provedených v určitém období na konkrétním nástroji, či stroji. Nabídka pro jeho spuštění se nachází v hlavním menu pod odkazem *Přehledy*.

6.12 Přehled nejopravovanějších nástrojů a strojů

Mezistupněm mezi přehledem uzavřených oprav a reporty je *Přehled nejopravovanějších nástrojů a strojů*. Tento formulář umožňuje uživateli si pro zvolené období zobrazit rychlý přehled všech nástrojů, které byly v tomto období v servisu a zejména také kolikrát. Pro větší názornost je součástí formuláře i grafické znázornění získaných dat.

Celý formulář vyobrazený na obrázku číslo 37, lze rozdělit do čtyř oblastí. První z nich je oblast ovládacích prvků se sadou nástrojů sloužících k variabilní volbě období, pro něž jsou následně data vyhodnocována. U textových polí pro zadání konkrétních termínů „Od“ a „Do“, se nachází dvojice tlačítek pro zobrazení datumového a časového voliče. Uživatel má tedy na výběr, zda vybere volbu datumového filtru pro konkrétní měsíc konkrétního roku, nebo navolí přesný

počáteční a konečný termín, mezi nimiž se nachází jím požadovaná data. Důležitým ovládacím prvkem je číselný volič udávající minimální počet uzavřených oprav jediného nástroje či stroje ve zvoleném období, které budou zahrnuty do výsledného zobrazení. Všechny nástroje, či stroje, které byly ve zvoleném období servisovány méněkrát, jsou tedy vypuštěny. Kliknutím na tlačítko *Zobraz*, dojde k aplikaci zadáných nastavení, vyhodnocení a zobrazení požadovaných informací.



Obr. 37 Formulář přehledu nejopravovanějších nástrojů a strojů.

V levé horní čtvrtině se nachází seznam nástrojů a strojů včetně počtu provedených servisů ve zvoleném období. Stejná data jsou také graficky znázorněna v dolní polovině formuláře pomocí horizontálního sloupcového grafu. Po kliknutí na libovolný záznam zmíněné tabulky v levé horní čtvrtině formuláře, následuje vypsání všech servisních zásahů, souvisejících s tímto záznamem ve zvoleném období, do tabulky pod ovládacími prvky. Poklepáním myši na kterýkoliv servisní záznam v této tabulce opět otevře náhled detailu tohoto záznamu.

6.13 Reporty

Jedním z nejdůležitějších aspektů každého informačního systému je vedle bezpečného uchovávání dat také zpětná vazba v podobě souhrnných informací které lze z těchto dat získat. K tomuto účelu je v aplikaci Opravy nástrojů vytvořeno několik reportů poskytujících širší náhled na problematiku servisního managementu ve firmě.

Všechny formuláře reportů jsou složeny ze dvou základních prvků. Prvním je nástrojová lišta v horní oblasti formuláře. Ta obsahuje všechny potřebné ovládací prvky, kterými se nastavují parametry zobrazeného reportu. Společným prvkem nástrojových listů všech reportovacích formulářů je tlačítko „Start“. Jakmile uživatel nastaví volitelné parametry reportu, klikne na toto tlačítko a vygeneruje tak požadovaný report.

Pod tímto panelem se pak nachází prohlížeč reportu, který má u svého horního okraje sadu ovládacích prvků, které kromě rolety pro volbu zvětšení či zmenšení zobrazení či nástroje pro listování vícestránkovým reportem také zahrnují možnost exportu reportu do formátů *pdf* či *xls*, což je souborový formát tabulkového procesoru Microsoft Excel. Umožňuje také vygenerovaný report vytisknout.

Všechny formuláře s reporty jsou spouštěny z jediného místa a to z roletového menu *Reporty* v hlavní nabídce.

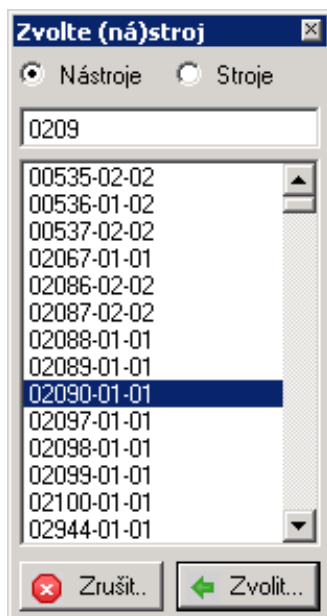
Prvním ze skupiny reportů je *Měsíční výkaz uzavřených oprav*. Jedná se o součet všech uzavřených servisních požadavků ve zvoleném měsíci rozdělených podle osob, kterými byly tyto požadavky uzavřeny. Tento report slouží zejména pro vedení výroby jako jeden z podkladů pro ohodnocení zaměstnanců. V nástrojové liště tohoto reportu se nacházejí dva roletové voliče, jeden pro volbu roku a druhý měsíce, pro nějž má být report vygenerován.

Dalším je *Roční přehled uzavřených oprav*. Jedná se o souhrnný report měsíčních počtů, cen a délek uzavřených oprav ve zvoleném roce. Na data lze aplikovat dva hlavní filtry a to buď zobrazit souhrn pro zvolené výrobní středisko, nebo přímo pro konkrétní nástroj či stroj. Číslo nástroje případně stroje lze zadat do textového pole ručně, nebo jej zvolit pomocí dialogového okna, které se zobrazí po kliknutí na ikonku vedle tohoto textového pole.

Výběrový formulář znázorněný na obrázku číslo 38, se skládá z voliče pro určení, zda mají být ve výběrovém seznamu zobrazeny nástroje nebo stroje. Pod těmito volbami se nachází vyhledávací textové pole, které je obdobné jako u správy uživatelských oprávnění. Stačí vepsat několik prvních znaků hledaného čísla nástroje (stroje) a nachází-li se zapsaný řetězec v seznamu, pak je takováto položka automaticky označena. Poté již stačí stisknout klávesu *Enter* či kliknout na tlačítko *Zvolit* a hodnota zvolené položky je vepsána do textového pole v nástrojové liště reportu.

Třetím reportem v řadě je koláčový graf rozdělující do svých výsečí servisní zásahy uzavřené ve zvoleném roce podle rozdílu hodnot požadovaného a skutečného termínu dokončení. Intervaly, podle nichž jsou servisní zásahy rozdělovány do jednotlivých výsečí, jsou nastavitelné. Uživatel může zadat v textovém poli nástrojové lišty reportu seznam intervalů ve formátu celých čísel, představujících dny, oddělených čárkami. Lze zadávat i záporné hodnoty intervalů. Ve výsledku to pak znamená, že výseč grafu s touto zápornou hodnotou rozdílu termínů udává poměrnou část všech uzavřených servisů dokončených před jejich požadovaným termínem

dokončení o hodnotu zadanou v tomto intervalu. Report je pojmenován jako *Roční přehled oprav uzavřených po termínu*, neboť primárně slouží jako rychlý náhled na míru plnění stanovených servisních termínů.



Obr. 38 Dialogové okno pro výběr nástroje/stroje

Předposledním reportem je *Roční přehled typů závad*. Jedná se o souhrn počtů všech výskytů typů závad všech servisních zásahů ve zvoleném roce. V nástrojové liště reportu lze opět navolit detailnější filtry pro výběr zobrazovaných dat. Přehled počtů výskytů jednotlivých typů závad tak lze vygenerovat pro konkrétní měsíc konkrétního roku a dokonce i pro konkrétně zvolený nástroj či stroj. Na výsledném reportu lze pak jednoduše vidět množství opakování jedné a téže závady vztahené k nastaveným kritériím.

Posledním nástrojem pro získání komplexnějších informací je report označený jako *Četnosti*. Jsou zde zahrnuty výpočty dvou základních, statisticky zajímavých informací a to počtu oprav a délky trvání oprav. Četnost počtu oprav je založena na denní bázi, tedy kolik oprav je denně uzavřeno. Poté se určí, kolikrát se jaký počet denně uzavřených oprav ve zkoumaném vzorku vyskytnul. Četnost délky oprav je jednodušší v tom, že nejsou určovány denní součty délek oprav, ale přímo délky oprav každého servisního požadavku a následně jejich četnost výskytu ve statistickém vzorku.

7 Závěr

Na základě zadání byl vytvořen informační systém pro servisní management společnosti LEAR Corporation Czech ve Vyškově. Byl úspěšně nasazen do provozu a v současnosti je denně využíván přibližně stovkou uživatel. Množství záznamů v databázi servisních zásahů se blíží počtu 10 000.

K jednomu z největších přínosů nasazení elektronického, informačního systému patří ucelení záznamů servisních zásahů. Nyní jsou všechny uspořádány přehledně na jednom místě, přičemž lze kdykoliv během okamžiku zjistit, například jaké zásahy byly na kterém nástroji kdy provedeny. To nachází uplatnění zejména při řešení interních, ale i zákaznických reklamací, pro dohledání veškerých důležitých informací o daném nástroji z období, kdy na něm byly vyráběny reklamované kusy. Lze tak snadno a rychle odhalit například chybu při servisu nástroje.

Hlavním výstupem, nebo jinak by se také dalo říci, zpětnou vazbou informačního systému je v kapitole 6.13 zmíněná pětice reportů. Tyto, při správném výkladu, velmi dobře slouží vedení firmy při rozhodování jako podklad k objektivnějším, manažerským rozhodnutím.

Statistické údaje servisů dávají zřetelný náhled na nákladovost provozu jednotlivých nástrojů či strojů. Z toho lze tedy následně usuzovat například jejich rentabilitu. Na tomto základě pak lze přijmout rozhodnutí, zda bude na daném nástroji dále vyráběno, nebo zda bude výroba například přesunuta k partnerské firmě. Co se týká nejproblematictějších nástrojů a strojů, o těch se vědělo již před zavedením elektronického, informačního systému, avšak teprve po jeho nasazení a získání zpětné vazby ze získaných dat byly odhaleny některé další, byť méně, ale přesto nezanedbatelně problematické nástroje a stroje. Tímto se dostaly také do hledáčku, a byla u nich přijata opatření pro jejich plynulejší chod.

Statistika počtu výskytů typů závad u jednoho stroje či nástroje zase pomáhá odhalit, na první pohled skryté problémy, které by se jinak jen těžko odhalovaly a to zejména proto, že by se na ně vůbec nepřišlo. Pokud se například u jednoho nástroje neustále opakuje stejný typ závady, pak to poukazuje na možné konstrukční chyby tohoto nástroje, eventuelně na opakovanou chybnou manipulaci s nástrojem, tedy chybu v pracovním postupu či nedbalost obsluhy. Je pak na inženýrech, aby skutečnou příčinu odhalili. Informační systém LVIS v tomto slouží „jen“ jako pomocník při odhalování možných problémových oblastí.

Neopomenutelným přínosem bylo také zobjektivnění ohodnocení zaměstnanců servisních oddělení na základě jejich skutečné výkonnosti. Nyní lze u každého servisního zaměstnance snadno zjistit, na jakých servisních zakázkách daný měsíc pracoval a kolik jich celkem uzavřel. Na tomto základě jsou pak vypláceny, či nevypláceny prémie jednotlivým zaměstnancům.

Informační systém se také stal neocenitelným pomocníkem pro plánovače výroby, ať už z pohledu toho, že nyní mohou snadno zjistit, zda je daný nástroj či stroj připraven pro výrobu, nebo se nachází na servisu, tak i ze strany plánování servisních zásahů. Systém umožňuje plánovači zadat například požadavek na připravení nástroje pro výrobu s dostatečným předstihem, aby bylo zaručeno jeho včasné nasazení do výroby.

Výsledkem zpětné vazby informačního systému jsou tedy zejména aktuální, přesné a hodnotné informace sloužící jako pomůcka při důležitých rozhodnutích spojených se servisním managementem firmy.

Do budoucna je plánováno rozšíření reportovací oblasti systému o další statistické přehledy, které budou mít vzhledem k nárůstu počtu záznamů ještě vyšší vypovídací hodnotu. Při dostatečném objemu dat je možné také zavést nástroje například pro predikci nákladů či vytíženosti servisních oddělení v budoucích měsících. K tomu jsou však zapotřebí data získaná během několika let. Již nyní však z reportu četností lze tento předpoklad částečně odhadovat.

Dalším možným rozvojem informačního systému je jeho přetvoření do podoby „tenké“, webové aplikace, k jejíž hlavním výhodám patří daleko menší nároky na hardwarové vybavení pracovních stanic, a nevyžaduje zvláštní distribuci aktualizací.

Značným osobním přínosem této práce bylo seznámení se s kompletním životním cyklem vývoje informačního systému od nuly přes jednání se zadavatelem, utváření základních rysů budoucí aplikace, psaní kódu, testování až po finálního nasazení fungujícího informačního systému.

8 Seznam použité literatury

- [1] Informační systém. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, datum poslední revize 31.3.2010 [cit. 2010-04-25]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Informační_systém>.
- [2] GUNDERLOY, Mike. *Z kodéra vývojářem*. Miroslav Müller, Václav Kadlec. Brno : Computer Press, a. s., 2007. 290 s. ISBN 978-80-251-1517-6.
- [3] *ThinkBuzan : Official mind mapping software by Tony Buzan* [online]. 2010 [cit. 2010-04-18]. Dostupné z WWW: <<http://www.thinkbuzan.com/>>.
- [4] *ThinkBuzan : Official mind mapping software by Tony Buzan* [online]. 2010 [cit. 2010-04-18]. Mind Map Gallery. Dostupné z WWW: <<http://www.thinkbuzan.com/uk/support/mindmapgallery>>.
- [5] Myšlenkové a mentální mapy – co to je? *Nadané dítě* [online]. Datum poslední revize 29.6.2010, [cit. 2010-04-18]. Dostupné z WWW: <<http://www.nadanedite.cz/view.php?nazevclanku=myslenkove-a-mentalni-mapy-%96-co-to-je&cislocclanku=2008020010>>.
- [6] WEILKIENS, Tim; OESTEREICH, Bernd. *UML 2 Certification Guide : Fundamental & Intermediate Exams*. [s.l.] : Morgan Kaufmann Publishers, 2007. 298 s. ISBN 978-0-12-373585-8.
- [7] HOLT, Jon. *UML for system engineering : Watching the wheels*. Second Edition. London, United Kingdom : The Institution of Engineering and Technology, 2007. 358 s. ISBN 978-0-86341-354-4.
- [8] *OMG Unified Modeling Language (OMG UML), Superstructure* [online]. Version 2. 2., 2009 [cit. 2010-03-19]. Dostupné z WWW: <<http://www.omg.org/spec/UML/2.2/Superstructure>>.
- [9] Unified Modeling Language. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, datum poslední revize 14.2.2010 [cit. 2010-03-09]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Unified_Modeling_Language>.
- [10] KANISOVÁ, Hana; MÜLLER, Miroslav. *UML srozumitelně*. 2. aktualizované vydání. Brno : Computer Press, a. s. , 2007. 178 s. ISBN 80-251-1083-4.
- [11] AGARWAL, Vidya Vrat; HUDDLESTON, James. *Databáze v C# 2008 : Průvodce programátora*. Krejčí Lukáš, Matoušek Jiří. Vydání první. Brno : Computer Press, a. s. , 2009. ISBN 978-80-251-2309-6.

[12] PIALORSI, Paolo; RUSSO, Marco. *Microsoft LINQ : Kompletní průvodce programátora*. Fadrný Jiří, Matoušek Jiří. Vydání první. Brno : Computer Press, a. s. , 2009. ISBN 978-80-251-2735-3.

[13] PIALORSI, Paolo; RUSSO, Marco. *Introducing Microsoft LINQ*. Redmond : Microsoft Press, 2007. ix, 197 s. ISBN 978-0-7356-2391-0.

[14] QVT. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, datum poslední revize 4. 5. 2010 [cit. 2010-03-09]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/QVT>>.

[15] Object Management Group. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, datum poslední revize 12. 5. 2010 [cit. 2010-03-09]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Object_Management_Group>.

[16] Object-modeling technique. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, datum poslední revize 22. 4. 2010 [cit. 2010-03-11]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Object-modeling_technique>.

[17] James Rumbaugh. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, datum poslední revize 11.3.2010 [cit. 2010-03-11]. Dostupné z WWW: <http://en.wikipedia.org/wiki/James_Rumbaugh>.

[18] Object-oriented software engineering. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 11. 5. 2003, datum poslední revize 14.4.2010 [cit. 2010-03-11]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Object-oriented_software_engineering>.

[19] Ivar Jacobson. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2.11.2004, datum poslední revize 3.9.2007 [cit. 2010-03-11]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Ivar_Jacobson>.

[20] Objectory AB. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 13.5.2009, datum poslední revize 6.6.2010 [cit. 2010-03-11]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Objectory_AB>.

[21] Grady Booch. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 13.5.2006, datum poslední revize 21.9.2007 [cit. 2010-03-11]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Grady_Booch>.

[22] Booch method. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 25.2.2002, datum poslední revize 8.5.2009 [cit. 2010-03-11]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Booch_method>.

[23] Unified Modeling Language. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 14. 2. 2005, datum poslední revize 25. 3. 2010 [cit. 2010-03-13].

Dostupné z WWW: <http://cs.wikipedia.org/wiki/Unified_Modeling_Language>.

[24] Microsoft SQL Server. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 1.12.2002, datum poslední revize 19.5.2010 [cit. 2010-03-21].

Dostupné z WWW: <http://en.wikipedia.org/wiki/Microsoft_SQL_Server>.

[25] *Lear MediaRoom* [online]. 2010, [cit. 2010-04-25]. Dostupné z WWW: <<http://lear.mediaroom.com/>>.

Seznam příloh

Příloha číslo 1	Ukázka původní papírové agendy	I
Příloha číslo 2	Myšlenková mapa.....	III
Příloha číslo 3	UML diagramy	V
Příloha číslo 4	Datový model	XVII
Příloha číslo 5	Tabulka přístupových oprávnění	XIX

Poukazka na opravu

LEAK CORPORATION

Číslo nástroje:	14838 - 01 - 04	Číslo dílu:	13083 623 696
Číslo stroje:	38	Zakázka je splněna:	x
Oprava nástroje:		Přestavět nástroj:	X
		Nový díl číslo:	14 844.623 . 665

Oprava / Poznámka	Odstranění závady
Vyčistit formu Není zakázka-info mistr Otisk č. 2 špatný -viz příloha + PŘESTAVĚT; DATUM 07/06 T: 117.06 / 14	Jádru pro el. č. 2 je zadáno - move. předpokládá 6 týdnů ke 12.6.06 NÁSTROJ VYČISTĚN A OČIŠTĚN DEMONTÁŽ A MONTÁŽ VODNÁ-OT.Č.2 PŘESTAVĚNO KIA 14844 DATUM: 07/06 VYČISTĚN CHLAD. SYST.
Termín opravy:	17.6.06 0:00
Vystavil:	Honemannová J.
Podpis:	<i>[Signature]</i>
Datum, Čas:	5.6.06 15:21
Shozeno seř.:	VOLKO
Podpis:	<i>[Signature]</i>
Datum, Čas:	5.6.06 / 17 ¹⁵

☒ 01 Chladicí systém ☒ 02 Datumovka ☐ 03 Horké kanály ☒ 04 Jádra ☒ 05 Ošetření ☒ 06 Ostatní ☒ 07 Přestavět nástroj ☐ 08 Přetoky	☐ 09 Tvarové desky ☐ 10 Vodící systém ☐ 11 Vyleštit ☐ 12 Vyhazovače ☐ 13 Zajištění nástroje ☐ 14 Zatržené díly v nástroji ☐ 15 ☐ 16				
Čas opravy:	5	Podpis:	<i>[Signature]</i>	Datum, Čas:	14.06.06 16:30
Nasazeno seř.:	NAVZATIL P.	Podpis:	<i>[Signature]</i>	Datum, Čas:	22.7.06 2:30
Prověrka účinnosti:	NE	Podpis:	<i>[Signature]</i>	Datum, Čas:	21.7.06

Poznámka:	NEJDE VYTOPIT TRYSKY Opus
-----------	---------------------------

ZABLOUDIL - 7.6 = 2H
NAVRATIL - 7.6 = 2H.

ČIŠTĚNÍ A ROZDĚLÁNÍ.

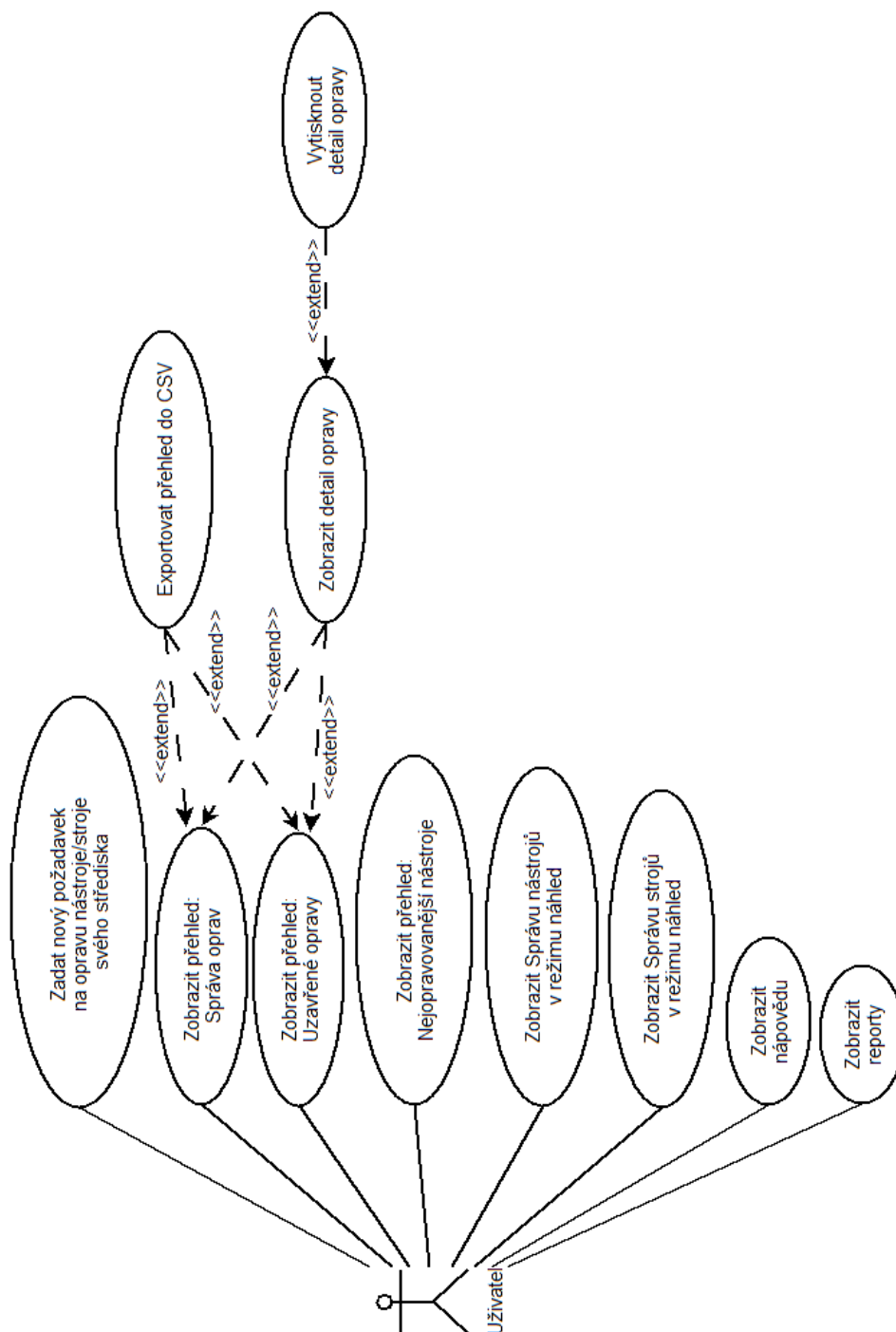
Příloha číslo 2 – Myšlenková mapa



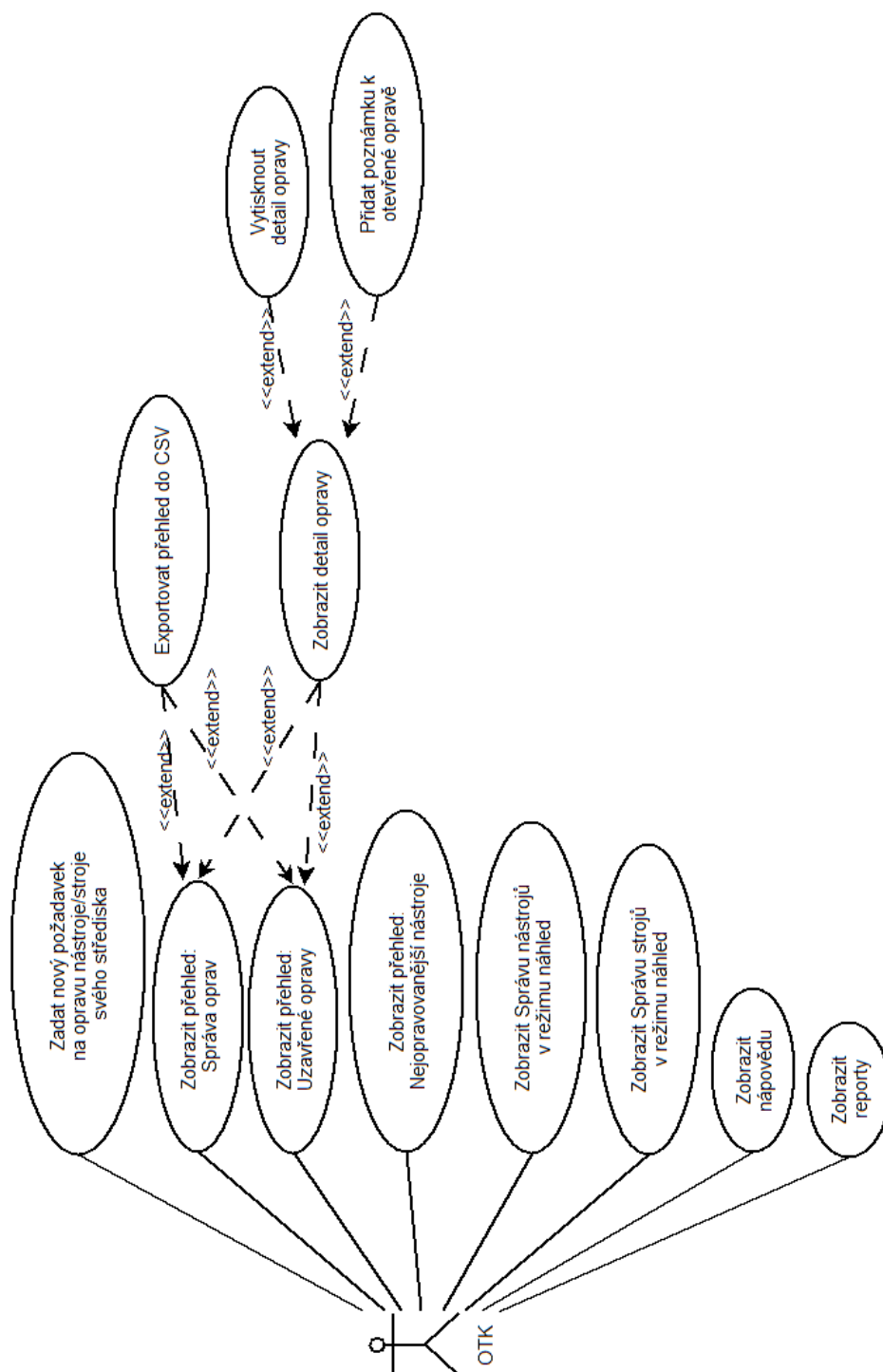
Příloha číslo 3 – UML diagramy

1. Diagramy případů užití

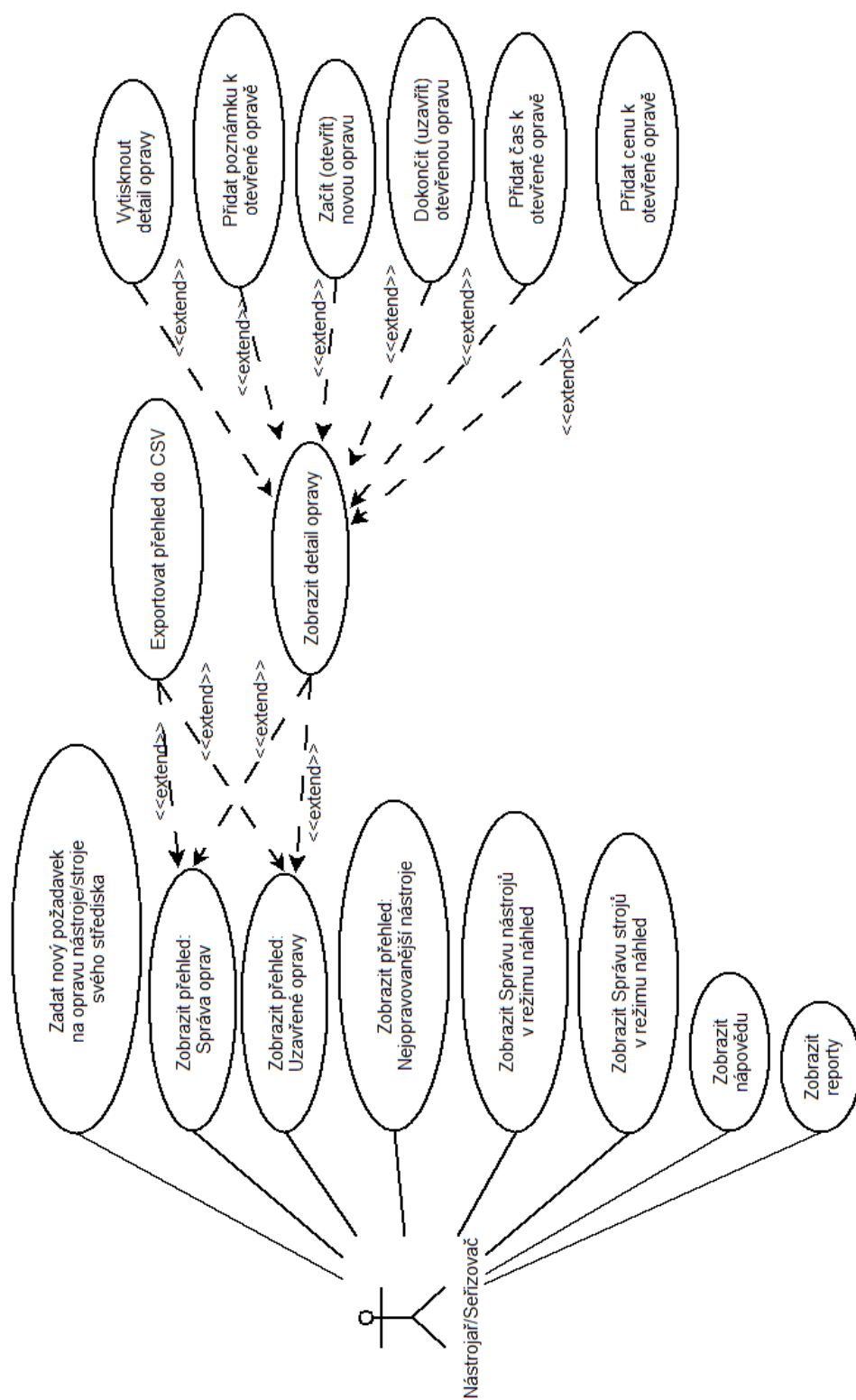
1.1. Uživatel



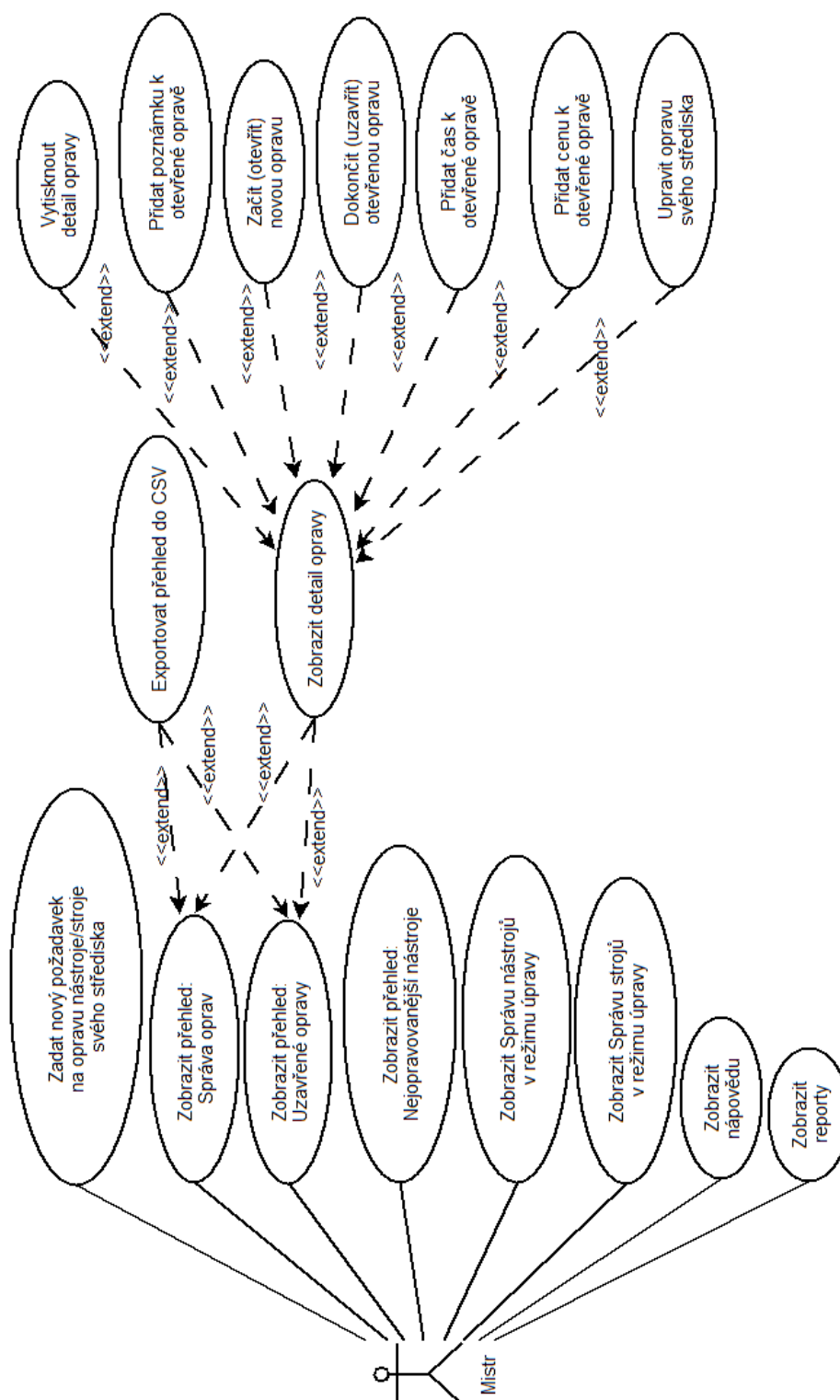
1.2.OTK



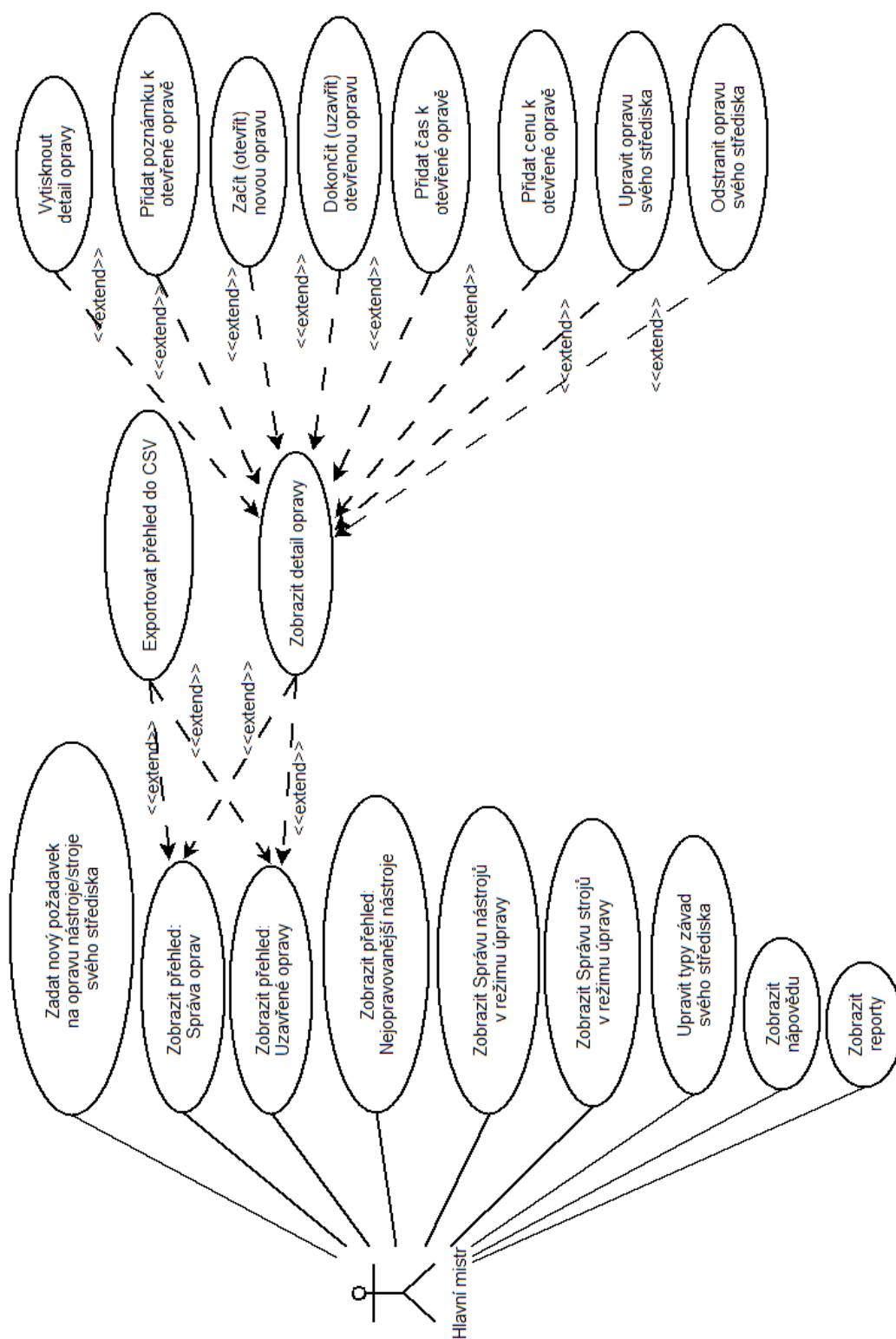
1.3. Nástrojař/Seřizovač

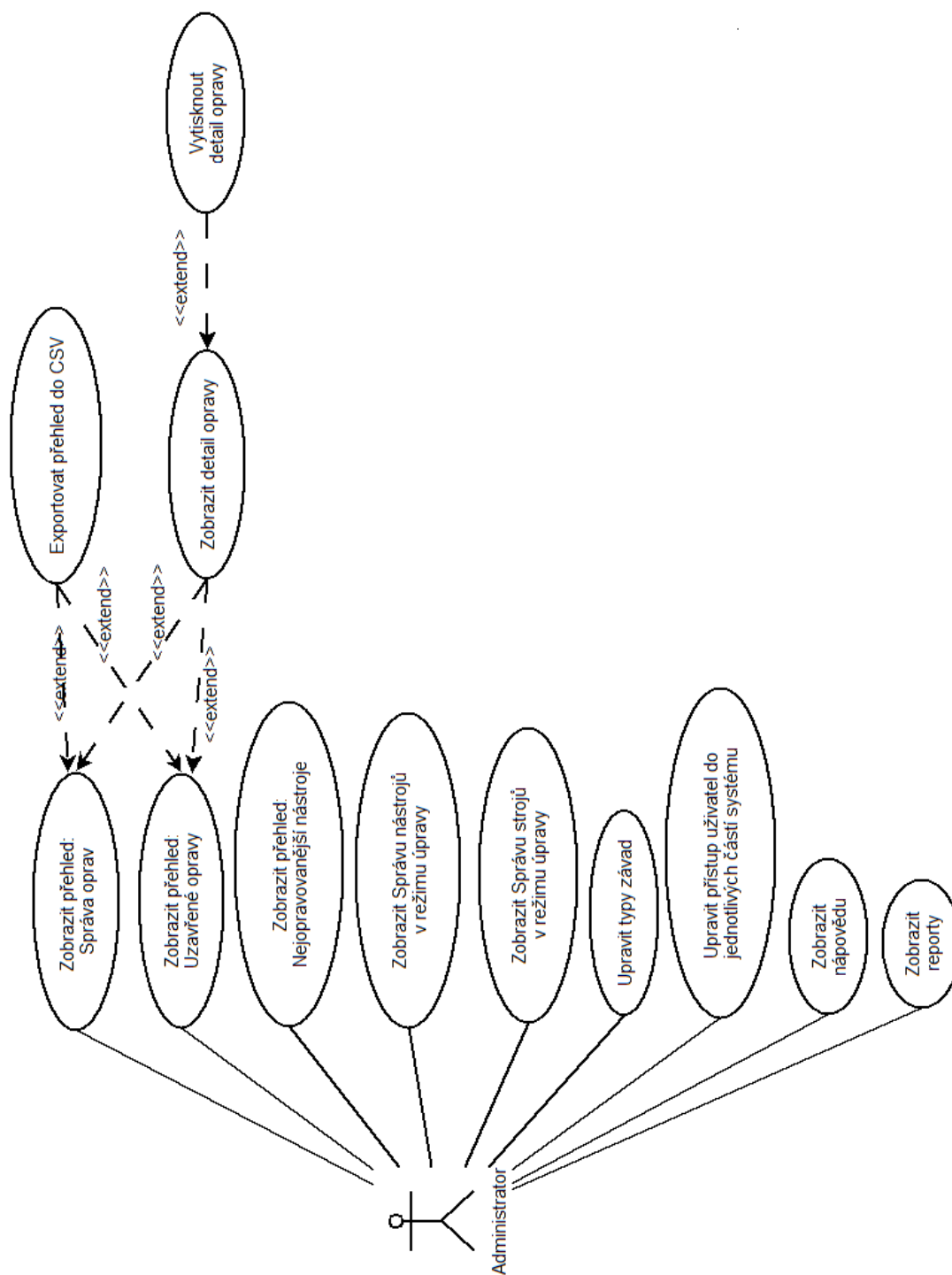


1.4. Mistr

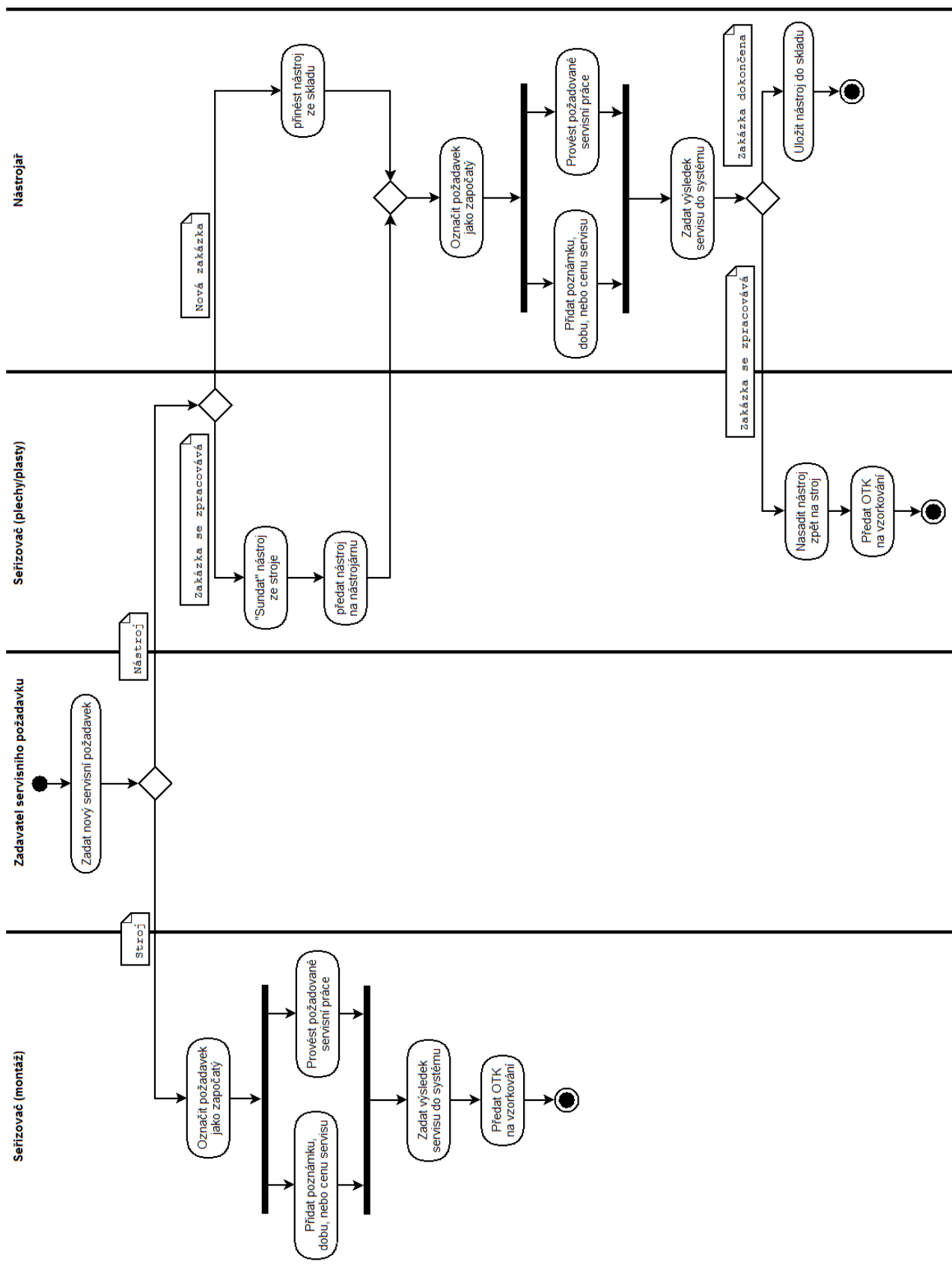


1.5. Hlavní mistr

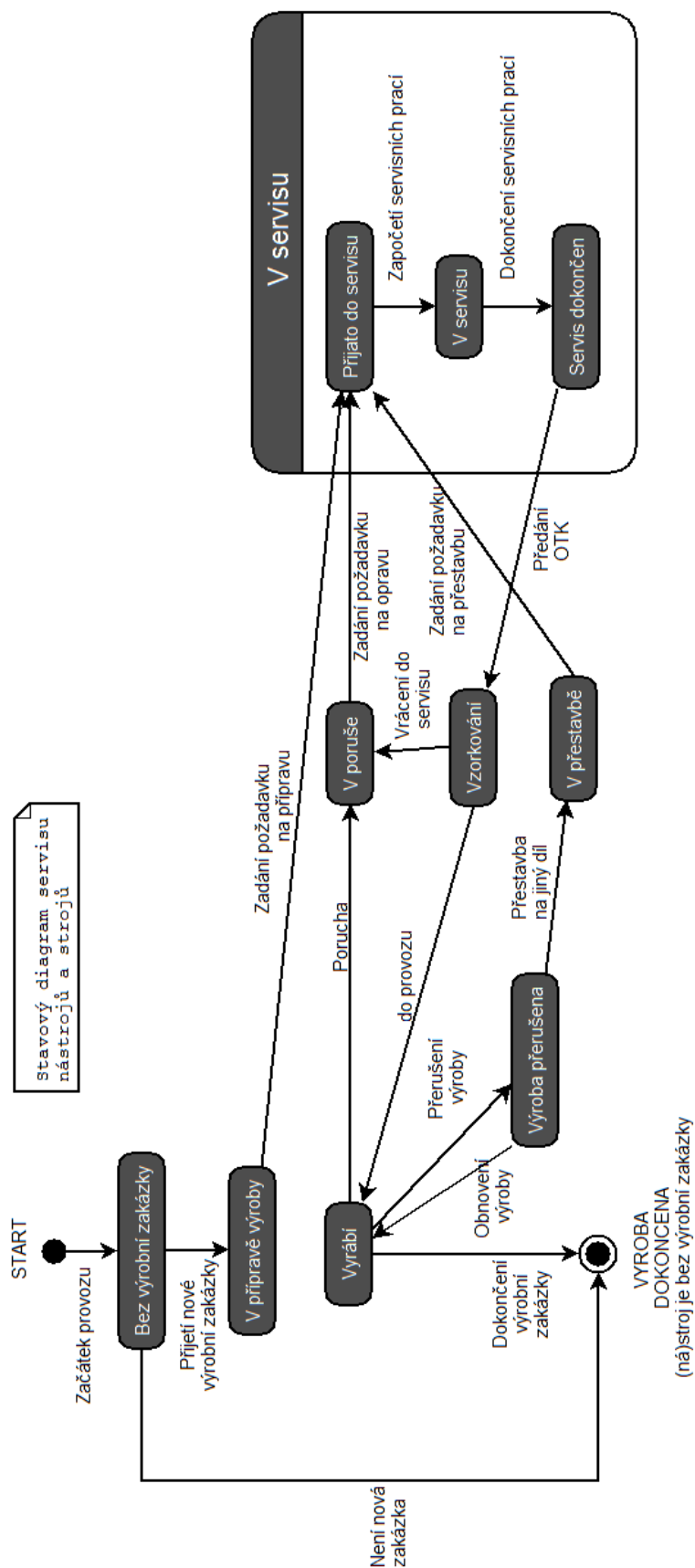


1.6.Administrátor

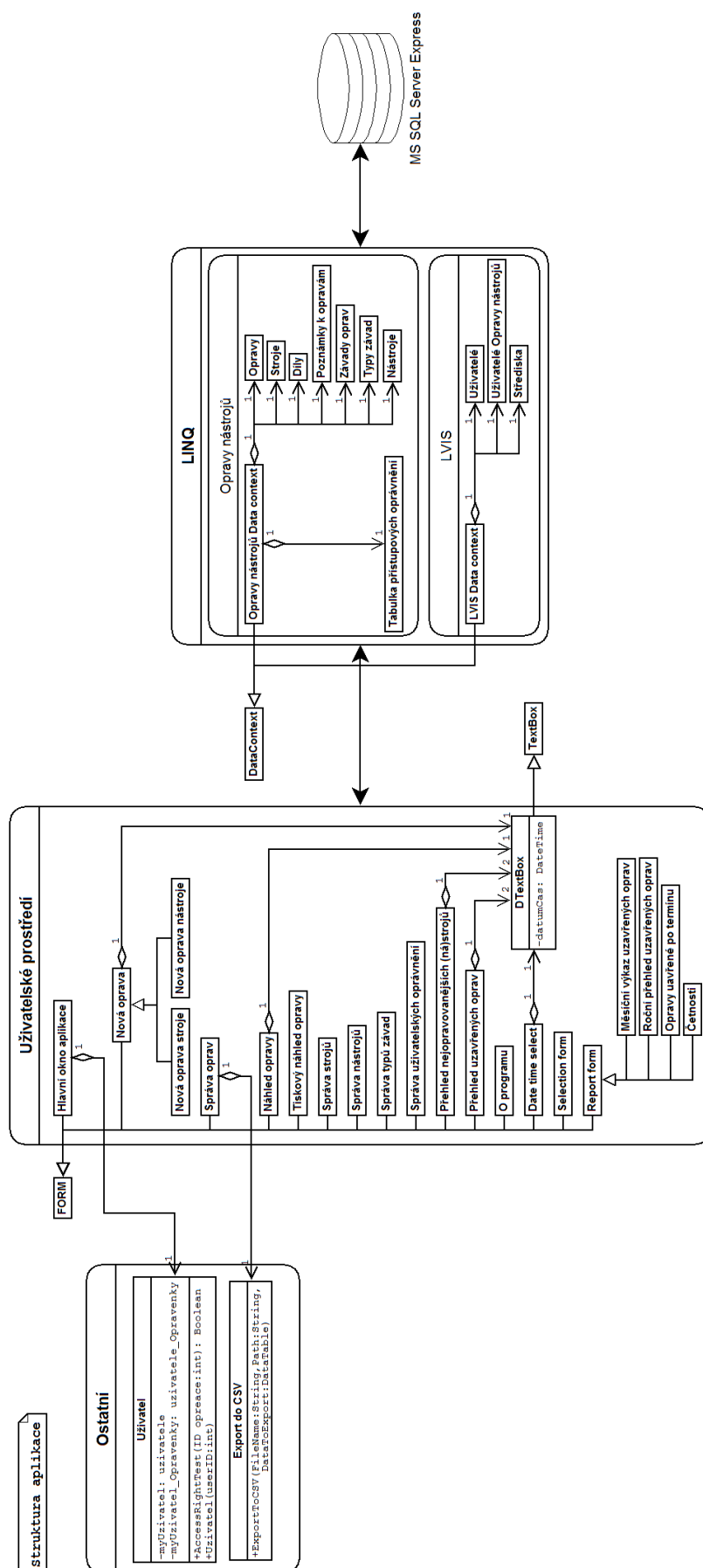
2. Diagram aktivit



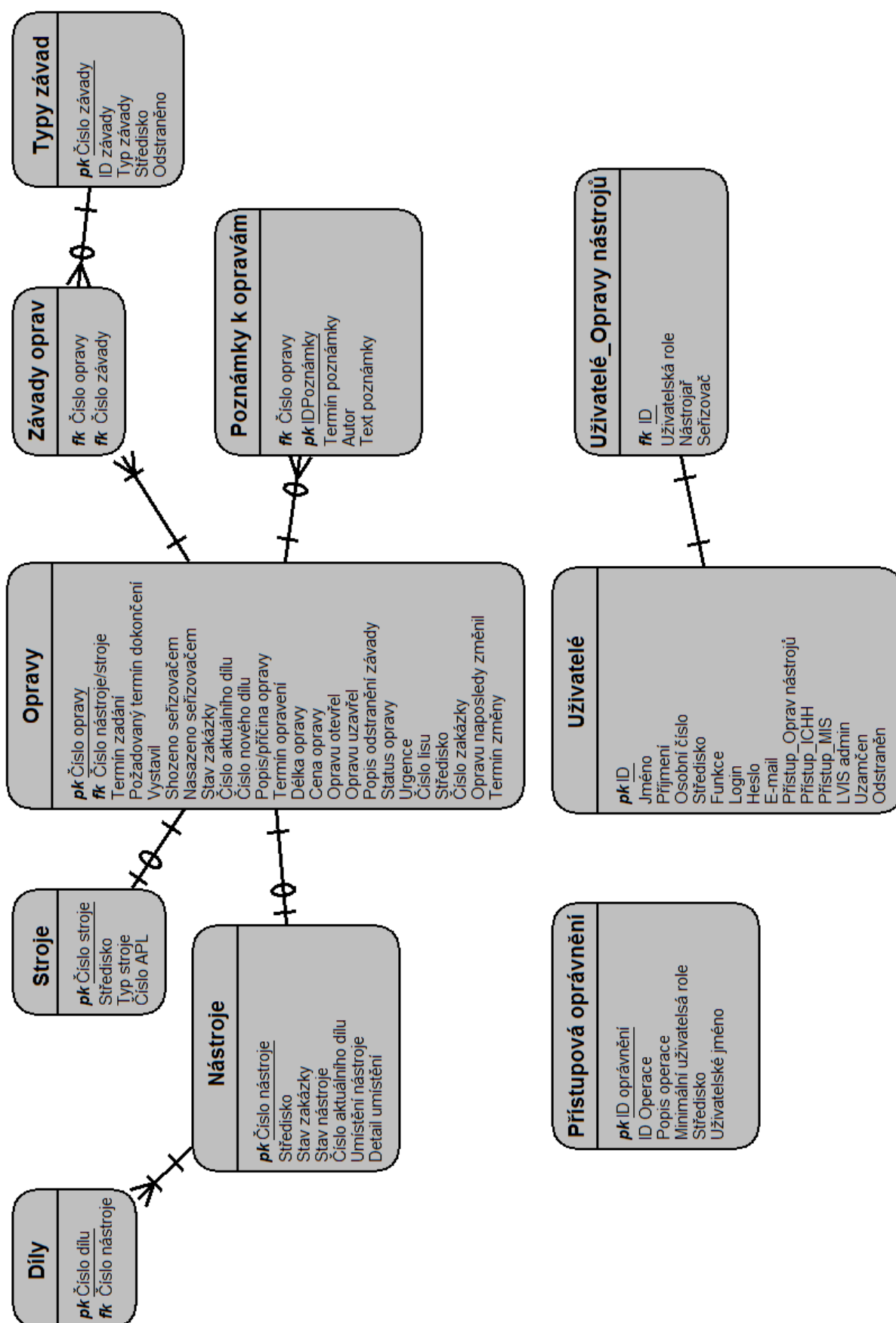
3. Stavový diagram nástrojů a strojů



4. Struktura aplikace Opravy nástrojů 2.0



Příloha číslo 4 - Datový model



Příloha číslo 5 – Tabulka přístupových oprávnění

Přehled požadovaných uživatelských oprávnění pro přístup k autorizovaným akcím systému	Uživatel			OTK			Nástroj / Seřizovač			Mistr			Hlavní mistr			Administrátor
	220	230	250	220	230	250	220	230	250	220	230	250	220	230	250	xxx
Přidat opravu stroje (220)	•			•			•			•			•			
Přidat opravu nástroje (230)		•			•			•			•			•		
Přidat opravu nástroje (250)			•			•			•			•			•	
Přidat poznámku k opravě (220)				•			•			•			•			
Přidat poznámku k opravě (230)					•			•			•			•		
Přidat poznámku k opravě (250)						•			•			•			•	
Přidat seřizovače (220)				•			•			•			•			
Přidat seřizovače (250)						•			•			•			•	
Otevřít/uzavřít opravu (220)							•			•			•			
Otevřít/uzavřít opravu (230)								•			•			•		
Otevřít/uzavřít opravu (250)									•			•			•	
Přidat čas/cenu (220)							•			•			•			
Přidat čas/cenu (230)								•			•			•		
Přidat čas/cenu (250)									•			•			•	
Upravit opravu (220)										•			•			
Upravit opravu (230)											•			•		
Upravit opravu (250)												•			•	
Správa nástrojů - úpravy (220)										•			•			
Správa nástrojů - úpravy (250)												•			•	
Správa nástrojů - úpravy (220,250)																•
Správa strojů - úpravy (220)										•			•			
Správa strojů - úpravy (230)											•			•		
Správa strojů - úpravy (250)												•			•	
Správa strojů - úpravy (220, 230, 250)																•
Odstranit opravu (220)													•			
Odstranit opravu (230)														•		
Odstranit opravu (250)															•	
Správa typů závad (220)													•			
Správa typů závad (230)														•		
Správa typů závad (250)															•	
Správa typů závad (220,230,250)																•
Správa uživatelských oprávnění																•

