

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2021

Bc. Lucie Byrtusová



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

MODUL TRASOVÁNÍ ŠARŽÍ PRO SYSTÉM SCADA V PROCESNÍ VÝROBĚ

TRACEABILITY MODULE FOR SCADA USED IN PROCESS INDUSTRY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Lucie Byrtusová

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jakub Arm, Ph.D.

BRNO 2021



Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Studentka: Bc. Lucie Byrtusová

ID: 195289

Ročník: 2

Akademický rok: 2020/21

NÁZEV TÉMATU:

Modul trasování šarží pro systém SCADA v procesní výrobě

POKYNY PRO VYPRACOVÁNÍ:

Tato práce má za úkol navrhnout a implementovat softwarový modul trasování (traceability) šarží do již existujícího SCADA systému MagmaView, přičemž je kladen důraz na implementaci aktuálních standardů platných v procesním průmyslu.

- 1) Proveďte literární rešerši standardu ANSI/ISA-95.
- 2) Proveďte literární rešerši v problematice traceability v procesní výrobě.
- 3) Definujte požadavky na softwarový modul.
- 4) Navrhněte datový a interakční model vytvářeného modulu.
- 5) Navrhněte GUI modulu a specifikujte interakci s uživatelem.
- 6) Realizujte softwarový modul.
- 7) Otestujte funkčnost a vyhodnoťte míru souladu se standardem.

DOPORUČENÁ LITERATURA:

Scholten, B. The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing. ISA, 2007. ISBN: 978-0979234385.

Termín zadání: 8.2.2021

Termín odevzdání: 17.5.2021

Vedoucí práce: Ing. Jakub Arm, Ph.D.

doc. Ing. Petr Fiedler, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Diplomová práce se zabývá implementací modulu trasování šarží do stávajícího SCADA systému MagmaView. První, teoretická část je věnována rešerši standardu ANSI/ISA-95 a rešerši problematiky trasování v procesní výrobě. V praktické části je nejprve rozebrán návrh datového modelu a modelu pro interakci s uživatelem. Dále je popsána implementace zmíněných návrhů, zahrnující několik softwarových nástrojů jako grafické prostředí MagmaView, Groovy skriptování nebo komunikace s SQL databází. Součástí implementace je také vzorový projekt simulující jednoduchý výrobní proces a potřeby operátora při práci se šaržemi.

KLÍČOVÁ SLOVA

MagmaView, SCADA, šarže, Groovy, skripty, SQL, traceabilita, trasování, standard ANSI/ISA-95, B2MML

ABSTRACT

The master thesis is focused on implementing a module for batch traceability into the existing SCADA system MagmaView. First, the theoretical part is dedicated to the research of ANSI/ISA-95 standard and traceability problematics in the process industry. First, the practical part analyses the design of the data model and model for user interaction. The next topic is the implementation of the mentioned models using several software tools such as the graphical environment of MagmaView, Groovy scripting, and communication with SQL database. Part of the implementation is also a testing project simulating a simple production process and the needs of the operator to work with the batches.

KEYWORDS

MagmaView, SCADA, batches, Groovy, scripts, SQL, traceability, tracing, standard ANSI/ISA-95, B2MML

BYRTUSOVÁ, Lucie. *Modul trasování šarží pro systém SCADA v procesní výrobě*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2021, 93 s. Diplomová práce. Vedoucí práce: Ing. Jakub Arm

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Bc. Lucie Byrtusová
VUT ID autora: 195289
Typ práce: Diplomová práce
Akademický rok: 2020/21
Téma závěrečné práce: Modul trasování šarží pro systém SCADA
v procesní výrobě

Prohlašuji, že svou závěrečnou práci jsem vypracovala samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autorky*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Ráda bych poděkovala vedoucímu diplomové práce panu Ing. Jakobovi Armovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	14
1 Standard ISA-95	15
1.1 Hierarchické modely	15
1.1.1 Model plánování a řízení	15
1.1.2 Model hierarchie zařízení	17
1.2 Model datových toků	18
1.3 Objektový model	20
1.3.1 B2MML	22
2 Trasování ve výrobě	24
2.1 Význam trasování	24
2.1.1 Trasování materiálu	25
2.1.2 Trasování výrobního postupu	25
2.1.3 Trasování - ostatní	26
2.2 Možnosti implementace traceability systému	26
2.2.1 Automatizované systémy	26
2.2.2 Budoucnost traceability systémů	28
2.3 Příklady dostupných řešení	28
2.3.1 COMES® Traceability	29
2.3.2 Act-in Traceability	30
2.3.3 Katana Product Traceability	31
3 Modul trasování šarží	32
3.1 Vizualizace MagmaView	32
3.2 Požadavky na modul trasování šarží	32
3.3 Fáze vývoje modulu	33
4 Návrh datového modelu	34
4.1 Využití standardu ISA-95 při návrhu databáze	34
4.1.1 Porovnání přístupů k návrhu	34
4.2 Databázové tabulky pro číselníky	35
4.2.1 Obecný popis tabulky	36
4.2.2 Tabulka Person	36
4.2.3 Tabulka Equipment	37
4.2.4 Tabulka ResourceNetwork	37
4.2.5 Tabulka MaterialClass	38
4.2.6 Tabulka MaterialDefinition	38

4.3	Databázové tabulky pro zaznamenávání množství materiálu	39
4.3.1	Tabulka MaterialLotQuantity	39
4.3.2	Tabulka EngineeringUnits	39
4.4	Databázové tabulky pro trasování šarží	40
4.4.1	Tabulka MaterialLot	40
4.4.2	Tabulka MaterialRatio	41
4.4.3	Tabulka ProductLot	42
4.4.4	Tabulka JobResponse	43
5	Návrh GUI modulu a popis interakce s uživatelem	44
5.1	Správa šarží	44
5.1.1	Menu pro správu šarží	45
5.1.2	Založení nové šarže	46
5.1.3	Editace existující šarže	48
5.1.4	Kontrola oprávnění	49
5.1.5	Zobrazení existujících šarží	50
5.1.6	Historie šarže	52
5.1.7	Detail šarže	53
5.2	Editace číselníků	55
5.2.1	Menu editace číselníků	55
5.2.2	Formulář pro editaci číselníků	56
6	Implementace softwarového modulu	58
6.1	SCADA MagmaView	58
6.2	Implementace databáze	59
6.3	Objekt Formulář	59
6.4	Objekt Tabulka	60
6.5	Groovy skripty	62
6.5.1	Editace číselníků	62
6.5.2	Nová šarže	63
6.6	Popis procesu	63
6.7	Řízení procesu a zaznamenávání procesních událostí	64
6.8	Zpracování procesních událostí jako historii šarže	67
6.9	Vyčtení historie šarže	70
6.10	Shrnutí procesu trasování šarží	73
6.11	Vyhodnocení běhu výrobního procesu	74
7	Vyhodnocení souladu se standardem	76
	Závěr	77

Literatura	79
Seznam symbolů a zkratk	81
Seznam příloh	82
A Datový model	83
B SCADA Vzorového projektu	85
C Obsah přiloženého CD	93

Seznam obrázků

1.1	Hierarchie úrovní podniku dle ISA-95 [1]	16
1.2	Hierarchie zařízení dle ISA-95 [1]	18
1.3	Datový tok rozhraní podnik-řízení [1]	19
1.4	Objektový model Možnosti výroby [1]	21
2.1	Dopředné a zpětné dohledávání zdrojů a produktů	24
2.2	Ukázka traceability systému společnosti COMES® [10]	29
2.3	Ukázka traceability systému společnosti Act-in [9]	30
2.4	Ukázka traceability systému společnosti Katana [12]	31
3.1	Blokový diagram systému MagmaView [13]	32
4.1	"Úplný"databázový model	34
4.2	"Redukovaný"databázový model	35
4.3	Databázová tabulka s uživateli	36
4.4	Databázová tabulka výrobních zařízení (umístění)	37
4.5	Databázová tabulka dodavatelů/zákazníků	37
4.6	Databázová tabulka tříd materiálu	38
4.7	Databázová tabulka definice materiálu	38
4.8	Databázová tabulka množství materiálu	39
4.9	Databázová tabulka inženýrských jednotek	40
4.10	Databázová tabulka výrobní šarže	41
4.11	Databázová tabulka poměru materiálu výrobní šarže	42
4.12	Databázová tabulka produktové šarže	42
4.13	Databázová tabulka historie umístění šarže	43
5.1	Diagram užití Správa šarže	45
5.2	Návrh menu pro správu šarží	46
5.3	Popup okno pro založení nové šarže	46
5.4	Diagram aktivity funkce Založit šarži	47
5.5	Diagram aktivity funkce Upravit šarži	48
5.6	Systémové hlášení o nedostatečném oprávnění uživatele	49
5.7	Popup okno pro správu existujících šarží	50
5.8	Diagram aktivity Zobrazit existující šarže	51
5.9	Popup okno zobrazující historii zvolené šarže	52
5.10	Diagram aktivity Historie šarže	53
5.11	Popup okno s detailními informacemi o dané šarži	54
5.12	Diagram aktivity Detail šarže	54
5.13	Diagram užití Editace	55
5.14	Menu pro editaci	55
5.15	Popup okno pro editaci číselníků a uživatelů	56

5.16	Diagram aktivity pro editaci číselníků a uživatelů	57
6.1	Realizace grafického objektu Tabulka v programovacím jazyce JavaScript	61
6.2	Přehledová obrazovka vzorového projektu pro trasování šarží	64
6.3	Datový blok pro ukládání záznamů o událostech	64
6.4	Tok materiálu v technologickém procesu vyjádřený pomocí Petriho sítě	65
6.5	Uživatelsky definovaný datový typ pro záznam o události	66
6.6	Varianty zobrazení historie šarže na podnět uživatele	70
6.7	Zobrazení historie výrobní šarže	71
6.8	Zobrazení historie materiálových šarží	72
6.9	Zobrazení historie výrobní šarže	72
6.10	Diagram zpracovávání dat o šaržích	73
B.1	Ukázka Runtime aplikace MagmaView	85
B.2	Přehledová obrazovka vzorového projektu pro trasování šarží	86
B.3	Základní menu modulu šarží	86
B.4	Formulář pro založení nové šarže	87
B.5	Hlášení o úspěšném provedení akce	87
B.6	Menu pro zobrazení historických dat o šaržích	87
B.7	Okno pro zobrazení historie procesních jednotek	88
B.8	Okno pro zobrazení historie materiálových šarží	88
B.9	Okno pro zobrazení historie materiálových šarží	89
B.10	Okno pro zobrazení historie materiálových šarží	89
B.11	Okno pro zobrazení historie výrobních šarží	89
B.12	Formulář pro editaci existující šarže	90
B.13	Formulář pro editaci třídy materiálu	90
B.14	Formulář pro editaci definice materiálu	91
B.15	Formulář pro editaci inženýrských jednotek	91
B.16	Formulář pro editaci uživatelů	92
B.17	Formulář pro editaci údajů o dodavateli/odběrateli	92

Seznam tabulek

- 6.1 Kombinace šarží jednotlivých komponent testovacích průběhů výroby 74
- 6.2 Počet záznamů v jednotlivých tabulkách pro dva výrobní procesy . . 74

Seznam výpisů

1.1	Úryvek z XML schématu B2MML-ProductionCapability [3]	23
6.1	Ukázka kódu MagmaView editoru - objekt Formulář	60
6.2	Ukázka předávání dat tabulce v Groovy skriptu	61
6.3	Ukázka kódu skriptu pro zpracování událostí z PLC	68

Úvod

Traceabilita, neboli schopnost trasovat výrobu, je váženou součástí podnikových systémů a pro některá odvětví průmyslu nepostradatelný nástroj, který pomáhá optimalizovat finanční ztráty podniku při problémech s produktem, vzniklých při výrobě, a také chránit své dobré jméno. Díky postupnému přechodu podniků od papírových reportů k automatizovaným systémům je proces trasování výroby značně usnadněn. Tento postupný vývoj se týká také vizualizačního systému MagmaView, pro který má být navržen a implementován modul pro trasování šarží, čemuž se věnuje tato diplomová práce.

První teoretická kapitola se věnuje popisu mezinárodního standardu ISA-95 a modelům, které jsou ve standardu zadefinovány. Popsána je také B2MML implementace standardu, která je stěžejní pro návrh modulu.

Teoretická část končí s druhou kapitolou, zaměřenou na traceabilitu výroby. V kapitole jsou nejprve vyzdviženy hlavní přínosy implementace této funkcionality do podnikových systémů a také jsou zde uvedeny různé pohledy na to, co lze do oblasti trasování zahrnout. Druhá část kapitoly se věnuje konkrétním příkladům dostupných řešení traceability na českém i zahraničním trhu.

Třetí kapitola je krátkým úvodem k modulu trasování. Nejprve je představena vizualizace MagmaView, do níž má být modul implementován. Dále jsou shrnuty požadavky na modul, co se od něj očekává a v jakém rozsahu, a jakým způsobem vývoj probíhá.

Prvním krokem návrhu modulu je sestavení datového modelu, kterému je věnována kapitola čtvrtá. Navržená databázová struktura je zde popsána jednotlivě po tabulkách, je vysvětlen jejich význam, souvislost se standardem a také jakým způsobem jsou tabulky vzájemně propojeny.

Druhým krokem návrhu modulu trasování je popis interakce s uživatelem a návrh GUI modulu, čemuž je věnována kapitola pátá. Návrh obsahuje sadu vizualizačních oken, skrze která bude obsluha systému spravovat šarže. Dále se kapitola soustředí na popis procesů, které souvisejí s uživatelskou interakcí.

Implementace navržených modelů je popsána v kapitole šest, kde je na vzorovém projektu vysvětleno, jakým způsobem je modul šarží realizován pomocí nástrojů MagmaView, Groovy skriptů a SQL dotazů. Kapitola je zakončena shrnutím popsaných postupů pro zpracování dat šarží a jakým způsobem se na databázových tabulkách projeví běh výrobního procesu.

Poslední kapitola práce obsahuje zhodnocení souladu modulu s platnými standardy.

1 Standard ISA-95

Pro účely modulu trasování šarží byl využit mezinárodní standard ISA-95, plným názvem „ANSI/ISA-95 Enterprise-Control System Integration“. Standard vznikl s cílem sjednotit do té doby různorodá integrační řešení. Slouží tedy jako nástroj pro propojení podnikových a řídicích systémů a definici rozhraní mezi nimi, nezávisle na výrobním odvětví nebo typu výrobního procesu.

ISA-95 obsahuje množství termínů a modelů, které určují, jaké informace je nutné předat mezi systémy podnikových úrovní a jak mají být tato data strukturována. Využitím standardu lze tedy usnadnit komunikaci mezi různými systémy a lidmi napříč odděleními a dosáhnout tak zvýšení efektivity a snížení nákladů.

Přestože je standard spojován zejména s integrací systému MES¹, který má za cíl být překladacem mezi řečí obchodní a řečí výrobní, je sestaven tak, aby byl průvodcem pro datový tok skrz všechny úrovně podniku od fyzických procesů, přes řídicí a ovládací systémy, až po administrativní doménu.

V kapitole jsou popsány zejména ty části ISA-95, které jsou stěžejní pro tuto práci.

1.1 Hierarchické modely

Jak již bylo zmíněno, standard se dotýká všech úrovní podniku, a to tak, že tyto úrovně definuje, popisuje, jaké funkce jsou pro jednotlivé úrovně typické a také jaká zařízení úrovně zahrnují. Popis je proveden na dvou modelech a to **Model plánování a řízení** a **Model hierarchie zařízení**.

1.1.1 Model plánování a řízení

Model plánování a řízení definuje hierarchii úrovní, ve kterých probíhá určitá forma rozhodování. Rozdělení je znázorněno na obrázku 1.1.

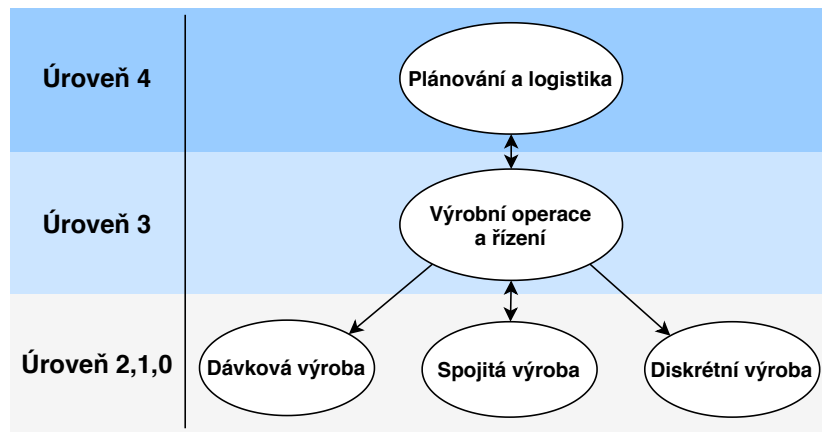
Úroveň 4 - Plánování a logistika

Nejvyšší úroveň, čtvrtá, představuje řízení podniku z hlediska plánování, logistiky, administrativy a obchodu.

V této sekci probíhají následující aktivity [1]:

- **Správa materiálu, energií a produktu** - Zajišťování materiálu a jeho uskladňování. Zajištění energií a sledování spotřeby. Uskladňování produktu.
- **Správa informací o kvalitě produktu** - Důležité pro zákazníka jako doklad o splnění požadavků. Dlouhodobé sledování vývoje kvality.

¹Manufacturing Execution Systems - Systém pro řízení operací výroby a přenos dat mezi procesní a administrativní vrstvou podniku



Obr. 1.1: Hierarchie úrovní podniku dle ISA-95 [1]

- **Správa zařízení** - Pro splnění plánu musí být všechna zařízení provozuschopná. Uchovávání dat o životě zařízení slouží k plánování údržby.
- **Personalistika** - Zajišťování lidských zdrojů v rámci celého podniku.
- **Základní plánování výroby** - Dlouhodobé plánování a stanovení cílů. Plánování v časovém rámci měsíců, dnů, směn.
- **Plánování kapacity** - S využitím výše uvedených informací je možné zkontrolovat, zda-li se podnik schopný naplnit plán a má k dispozici veškeré potřebné zdroje.

Úroveň 3 - Výrobní operace a řízení

Následuje třetí úroveň, na kterou je směřována podstatná část standardu. Jedná se o řízení výroby a zahrnuje tyto funkce [1]:

- **Přidělování a řízení zdrojů** - Na rozdíl od předchozí úrovně se zde zdroje, tj. materiál, nástroje, obsluha, přidělují již v konkrétních jednotkách konkrétním zařízením, které musí být rovněž připraveny v řádném stavu.
- **Dispečerské řízení** - Zpracovávání příkazů, objednávek, dávek a úkonů.
- **Sběr a archivace dat** - Ukládání dat a parametrů z procesu a ze zařízení. Funkce rovněž zodpovídá za zaznamenání nestandardních událostí a alarmů.
- **Řízení kvality** - Kontrola parametrů a měření v reálném čase pro zajištění splnění požadavků na kvalitu produktu.
- **Řízení procesu** - Monitorování produkce. Podpora pro údržbu a řešení nečekaných událostí.
- **Plánování výroby a trasování** - Funkce poskytuje přehled o aktuálních stavech a podmínkách produkce, včetně přiřazené obsluhy, použitého materiálu apod. Funkce by rovněž měla umožňovat vysledovat kompletní historii produktu.

- **Analýza výkonnosti** - Funkce zpracovává procesní data pro sestavení reportů o výsledcích produkce v porovnání s požadavky nebo předchozími reporty.
- **Operativní plánování** - Výsledkem operativního plánování je sekvence úkonů sestavených na základě zadaných priorit, specifík a produkčních pravidel.
- **Správa dokumentace** - Zahrnuje správu instrukcí, receptur, procedur, výkresů, záznamů, komunikace mezi směnami apod.
- **Management lidských zdrojů** - Správa pracovníků a jejich pracovních rozvrhů, informace o pracovnících, kvalifikace apod.
- **Plánování údržby** - Udržování nástrojů a zařízení v provozuschopném stavu. Plánování pravidelných nebo preventivních odstávek. Sběr dat o chodu zařízení a případných problémech.

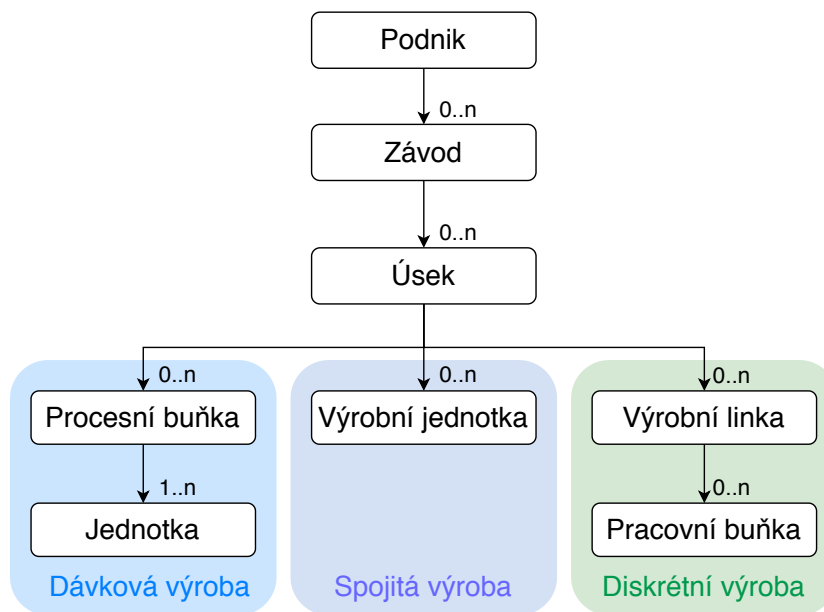
Úrovně 2,1,0

Nejnižší úrovně 2,1 a 0 se týkají procesního řízení, funkcí operací a supervizních funkcí a mají své vlastní modely v závislosti na typu výroby [1]. Zde navazuje standard ISA-88, který se věnuje dávkovým procesům a je podrobněji rozebrán v kapitole 2.

1.1.2 Model hierarchie zařízení

Dalším hierarchickým modelem je model zařízení, který líčí typickou organizaci fyzického vybavení podniku s dávkovou, diskrétní nebo spojitou výrobou viz. obrázek 1.2. Každý stupeň modelu představuje zodpovědnost za určitá rozhodnutí a funkcionalitu charakteristickou pro konkrétní úroveň podniku. Zařízení si mohou vyměňovat data se svými hierarchickými sousedy nebo mohou tvořit různá seskupení a sdílená data kumulovat. Seskupení zařízení na nižších úrovních tvoří prvky na vyšších úrovních. Seskupení se mohou i vzájemně překrývat [1].

Nejvyšším prvkem hierarchie zařízení je **podnik**. Podnik rozhoduje, jaké produkty se budou vyrábět a kde. Podnik může být seskupením jednoho a více závodů. **Závod**, stejně jako podnik spadá pod **úroveň 4 v modelu řízení a plánování** viz. 1.1.1. Závod představuje fyzickou nebo logickou buňku, nejčastěji popisovanou svou geografickou polohou, která se stará o lokální management a optimalizaci. Cíle vycházející z této úrovně jsou výsledkem hrubého plánování výroby jednoho nebo více úseků.



Obr. 1.2: Hierarchie zařízení dle ISA-95 [1]

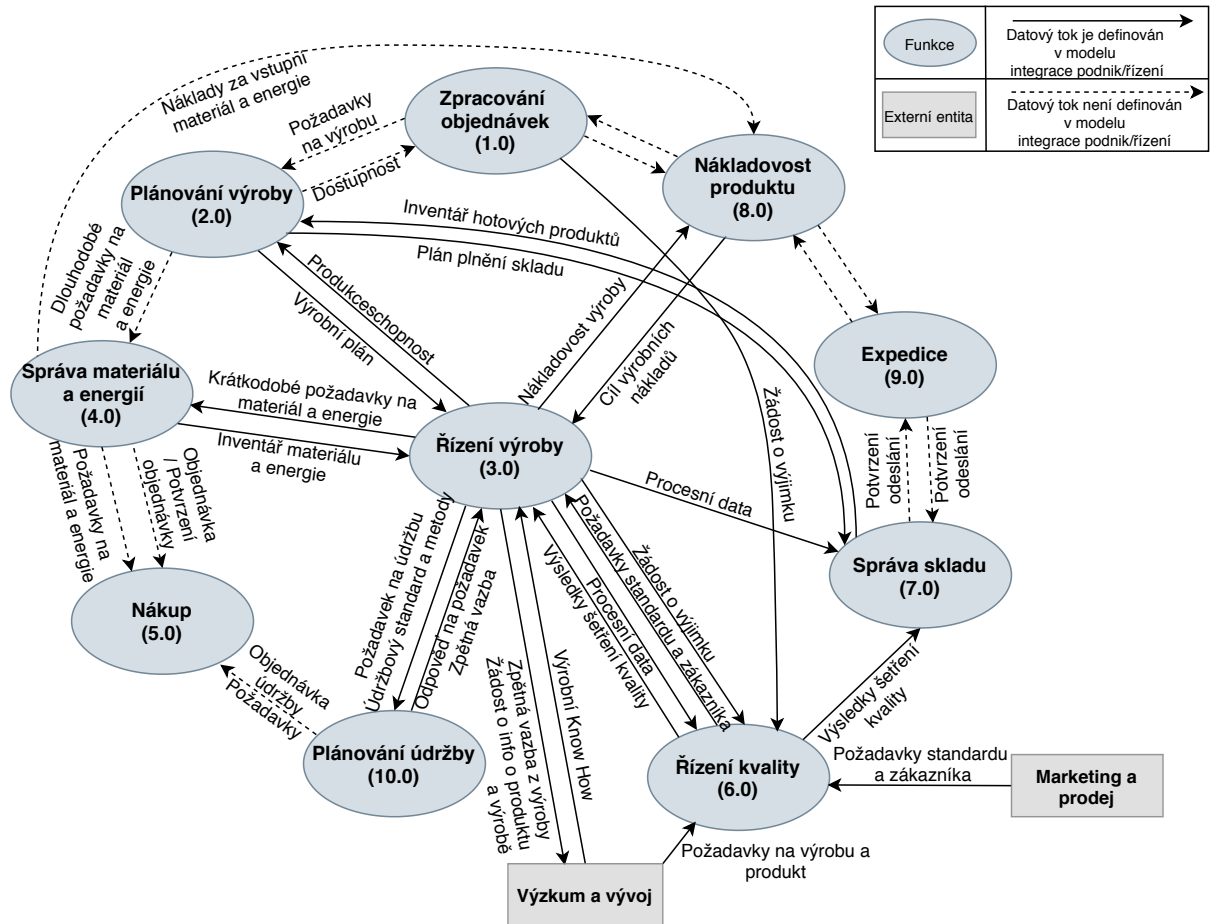
Úsek je obvykle nazýván dle svého účelu nebo koncového produktu, který z něj vychází. Na úsecích zpravidla končí úroveň 4 a začíná úroveň 3, tzn. hrubé plánování se promítá do detailních rozvrhů s ohledem na kapacitu a možnosti úseku. Úsek se skládá z elementů nižší úrovně odpovídajícího druhu výrobu nebo jejich kombinace. Obecně se tyto elementy váží již ke konkrétním fyzickým zařízením a jejich přímým řízením. Pro dávkovou výrobu to jsou elementy jako **procesní buňka** a **jednotka**, pro spojitou výrobu je oním elementem **výrobní jednotka** a pro diskrétní výrobu mluvíme o **výrobní lince** a **pracovní buňce** [1].

Výše uvedená hierarchie zařízení je podmíněna rozsahem podniku. Počty nižších elementů mohou být v řádech desítek nebo také nemusí existovat.

1.2 Model datových toků

Rozhraní podnik-řízení, neboli rozhraní mezi úrovní 3 a úrovní 4, je ve standardu popsáno pomocí modelu datových toků. Tento model definuje podnikové funkce zabývající se výrobou a výměnou dat mezi nimi. Model je interpretován na obrázku 1.3, skládá se z desíti funkcí a dvou entit, které mohou spadat pod úroveň řízení podniku nebo řízení výroby, což je rozdílné podnik od podniku. Jednotlivé funkce jsou vzájemně propojeny dvěma typy čar, opatřeny šipkou indikující směr toku dat. Plná černá čára představuje data, která jsou jasně definována v modelu integrace podnik-řízení a jsou nezbytná pro řízení výroby. Všechny tyto čáry nesou označení. Přerušovaná čára naznačuje toky informací nedefinovaných modelem integrace

podnik-řízení a nemusí nutně nést označení. Jejich smyslem je doplnit představu o datovém toku na rozhraní [1].



Obr. 1.3: Datový tok rozhraní podnik-řízení [1]

Následující výčet funkcí stručně přibližuje vykonávanou činnost, jaká data funkce zpracovává a s jakými funkcemi data sdílí [1]

1. **Zpracování objednávek** - Funkce přijímá, zpracovává a potvrzuje zákaznickou objednávku a definuje z ní plynoucí požadavky na výrobu s ohledem na dostupnost. Předvídá vývoj prodeje a hlásí hrubou marži.
2. **Plánování výroby** - Funkce zpracovává plán výroby na základě schopnosti výroby plnit plán a na základě počtu již hotových naskladněných produktů. Výstupem je také informace pro správu skladu, kolik místa je třeba rezervovat pro novou várku. Dále definuje dlouhodobé požadavky na materiál a energie.
3. **Řízení výroby** - Zde ústí většina dat spojených s řízením výroby. Soustřeďují se zde informace o materiálu, o nákladech, kvalitě i údržbě. Výstupem z funkce jsou informace o kapacitě výroby a procesní data. Obousměrná komunikace zde probíhá i s externí entitou Výzkum a vývoj, jejímž produktem je know how

- potřebné pro výrobu, sestavené a upravené na základě zpětné vazby z výroby.
4. **Správa materiálu a energií** - Vytváření objednávek materiálu v reakci na požadavky z plánování a řízení výroby.
 5. **Nákup** - Vyřizuje objednávky materiálu a údržby.
 6. **Řízení kvality** - Zpracovává procesní data za účelem analýzy kvality koncového produktu a plnění požadavků na kvalitu. Funkce zpracovává rovněž požadavky z obou externích entit Výzkumu a Prodeje.
 7. **Správa skladu** - Funkce zahrnuje správu materiálu i hotových produktů, včetně analýzy jejich kvality. Odpovídá na požadavky expedice a plánování výroby.
 8. **Nákladovost** - Počítá celkové náklady na výrobu produktu. Výstupem je vytyčení hranic nákladů, se kterými dále pracuje funkce Řízení výroby.
 9. **Expedice** - Organizace transportu produktu a komunikace s transportními společnostmi. Příprava dokumentů potřebných k expedici. Výstupem jsou náklady na expedici a potvrzení odeslání.
 10. **Plánování údržby** - Funkce zahrnuje monitoring stavu a provozu zařízení a zajištění údržby a nákup náhradních dílů.

1.3 Objektový model

Informace definované v modelu datových toků (1.2) lze obecně rozdělit do tří oblastí:

- Informace o schopnosti vyrobit produkt
- Informace specifikující produkt a jeho výrobu
- Informace o skutečné výrobě produktu

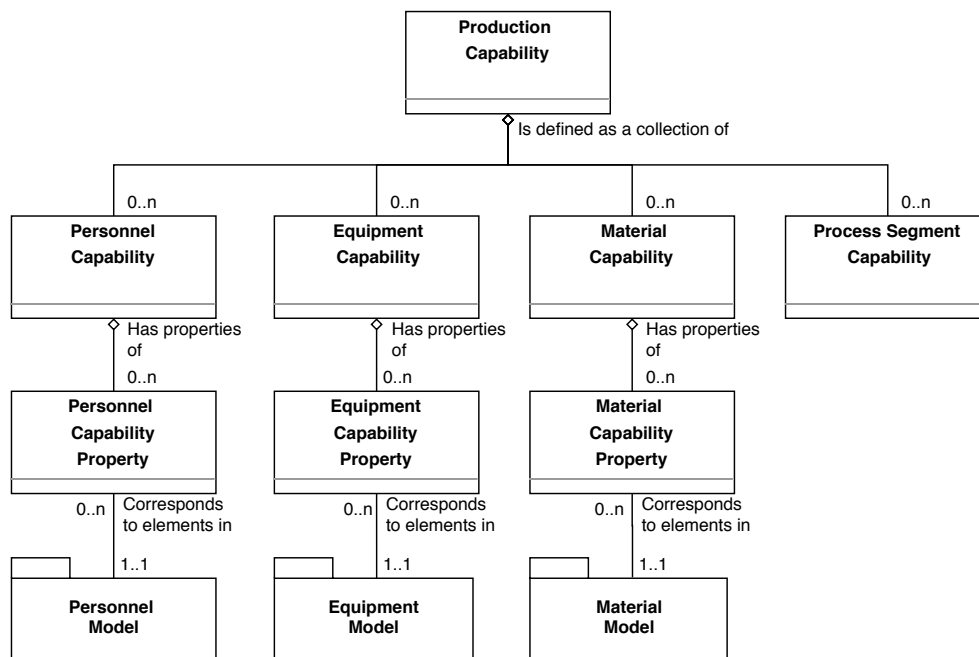
Pro všechny tři oblasti je nezbytné si část informací předávat s dalšími podnikovými systémy. Každá z těchto překlápěných informací je součástí jednoho nebo více objektových modelů definovaných standardem, které slouží k ilustraci struktury dat. Dané struktury byly navrženy s ohledem na skutečnost, že systémy sdílející data mohou být rozdílné a obě strany musí být schopné si data přeložit se stejným významem.

Zmiňované objektové modely jsou sada diagramů vytvořených pomocí jazyka UML, což je jednoduchý grafický jazyk používaný v softwarovém inženýrství. Pro příklad bude uveden a popsán objektový model Možnosti výroby viz. obrázek 1.4.

Informace o schopnosti vyrobit produkt

Jedná se o soubor informací o všech zdrojích potřebných k výrobě, o jejich dostupnosti, stavu, množství apod. Nezbytné jsou rovněž jedinečné názvy jednotek těchto zdrojů, které budou využity pro plánování výroby a údržby.

Příklad struktury těchto předávaných informací dle standardu je znázorněn na obrázku 1.4. Obrázek je uveden v originálním jazyce, aby nedošlo k zavádějícímu výkladu standardu. Informace o schopnostech výroby (Production Capability) jsou složeny z údajů následujících tříd: možnosti personálu (Personnel Capability), možnosti zařízení (Equipment Capability), možnosti materiálu (Material Capability) a možnosti výrobního segmentu (Process Segment Capability). Specifika těchto tříd jsou dále uvedena v třídách typu vlastnost (Property). Data, včetně specifik, z uvedených tříd jsou reference, nebo set referencí, na konkrétní jednotky zdrojů definovaných v dalších modelech (Personnel Model, Equipment Model, Material Model). Obdobný model by byl aplikovatelný také na segment výroby [1].



Obr. 1.4: Objektový model Možnosti výroby [1]

Informace specifikující produkt a jeho výrobu

Skupina informací specifikujících produkt je sdílena mezi externími modely jako kusuovníky a soupiska zdrojů, definované mimo standard ISA-95, nebo výrobní postupy. Třídy modelu Definice produktu se podobají modelu předchozího souboru informací. Ale na rozdíl od dostupnosti je zde relevantní přímá specifikace vlastností materiálu, zdrojů a personálu. Každá specifikace má výčet vlastností, které jsou opět zdefino-vány ve stejných, již zmiňovaných modelech jako Personnel Model nebo Equipment Model [1].

Informace o skutečné výrobě produktu

Informace o skutečné výrobě produktu jsou definovány dvěma modely, které odpovídají žádostem o výrobu a odpověďmi na tyto žádosti.

Žádost o výrobu produktu vyplývá z kroků výrobního postupu a lze ji rozložit na parciální žádosti o přidělení segmentu výroby. Tyto parciální žádosti lze zase rozložit na mikrožádosti o všechny prvky nutné pro výrobu jako žádost o přidělení zdrojů, žádost o materiál nebo žádost o pracovní sílu. Každá ze žádostí je blíže specifikovaná třídami typu *Vlastnost*, které jsou definovány, obdobně jako pro předchozí dvě skupiny informací, v dalších modelech jako *Personnel Model* nebo *Equipment model*. V reakci na žádosti jsou obchodnímu systému předávána zpět aktuální data o stavu od nejnižších výrobních jednotek a zdrojů až po celé výrobní segmenty [1].

1.3.1 B2MML

Objektové modely sice poskytují dostatečný přehled o koloběhu dat mezi podnikovými systémy, nicméně jejich implementace by vyžadovala nemalé úsilí a nic by nezaručovalo zajištění jednotnosti řešení pro všechny systémy. V tohoto důvodu vznikla B2MML implementace, která objektové modely standardu ISA-95 překládá do XML schémat. Smysl zůstává stejný. Stejně jako objektové modely, i XML schémata vyjadřují, jakým způsobem musí být data strukturována, aby mohla být předávána mezi různými aplikacemi. V tomto případě jsou schémata cílena již na softwarovou implementaci.

Využívání schémat umožňuje aplikacím, nebo programátorům, rozumět tomu, jaká data přijímá nebo v jakém formátu musí data odeslat. Každému schématu B2MML přísluší dokumentace, která obsahuje vysvětlení významu jednotlivých tříd včetně diagramu elementů ve schématu a návaznosti na příslušný model z ISA-95.

Pro ukázkou je opět použit objektový model *Možnosti výroby* z obrázku 1.4. V této struktuře je uvedena třída **Možnosti personálu Vlastnost**. Tato třída odpovídá elementu "**PersonnelCapabilityPropertyType**" jehož šablona je specifikována v XML schématu B2MML - *Production Capability*. Jak můžeme vidět v ukázce kódu, viz. výpis 1.1, každý prvek této třídy by měl obsahovat identifikátor "ID", popis "Description", hodnotu "Value" a množství "Quantity"[3].

Výpis 1.1: Úryvek z XML schématu B2MML-ProductionCapability [3]

```
<xsd:complexType name = "PersonnelCapabilityPropertyType">
  <xsd:sequence>
    <xsd:element name = "ID" type = "IdentifierType" />
    <xsd:element name = "Description" type = "DescriptionType"
    <xsd:element name = "Value" type = "ValueType"
    <xsd:element name = "Quantity" type = "QuantityValueType"
    <xsd:group ref = "Extended:PersonnelCapabilityProperty"
  </xsd:sequence>
</xsd:complexType>
```

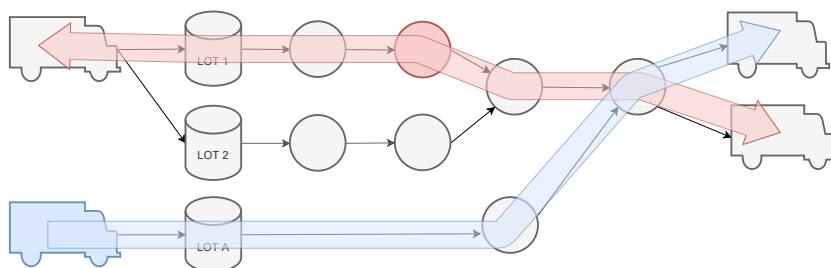
Šablona každé další třídy je specifikována obdobným způsobem. Součástí schémat jsou také šablony transakcí, tedy softwarové dotazy a žádosti na element příslušné třídy.

2 Trasování ve výrobě

Trasování výroby, označováno také jako Traceability, je jedna z klíčových funkcionalit pro podnikové systémy, jejíž cílem je využít dostupná data pro indikaci aktuálního stavu produktu nebo vytvoření jeho kompletní historie. Vytvořenou historii je možné využít k dopřednému dohledávání zdrojů (od vstupního materiálu po konečný produkt), tak i k zpětnému dohledávání zdrojů (od konečného produktu ke vstupnímu materiálu). Trasovat lze nejen pohyb materiálu, ale také další zdroje jako vybavení a obsluhu, nebo průběh výroby produktu.

2.1 Význam trasování

Nejčastějším důvodem k využití informací získaných trasováním výroby je řešení příčin a následků nekvality. Nehledě na výrobní odvětví, při výrobě kdejakého produktu mohou vznikat chyby. S některými se počítá, např. zmetky vzniklé v důsledku chyby opakování nebo jiných nepřesností, jiné mohou vzniknout nepozorovaně a čím pozdější odhalení, tím vyšší ztráty. Při odhalení chyby je prvním krokem vypořádat se s nečekanou situací. Může to například znamenat stáhnout vadnou šarži od všech distributorů nebo zastavit další zpracovávání vadné dávky. Dalším krokem je zjistit příčinu chyby. Zde se dohledávání komplikuje, neboť příčinou může být celá řada zdrojů a dostupnost velkého množství informací automaticky negarantuje rychlé a přesné odhalení chyby. Zajisté však usnadní alespoň zúžení oblasti pátrání například díky kontrole dodržení kroků výrobního postupu nebo kontrole parametrů a hodnot během průběhu výroby.[7]



Obr. 2.1: Dopředné a zpětné dohledávání zdrojů a produktů

Dalším přínosem traceability je možnost optimalizovat výrobu. Z dostupných dat lze studovat vliv změny hodnot a úpravy výrobních postupů na kvalitu, optimalizovat ztráty sledováním bilancí materiálů nebo také výkon na směnách. Zde ovšem kromě traceability vstupují do rovnice i jiné nástroje podnikových systémů. Posledním, zde zmíněným, důvodem k trasování výroby jsou legislativní podmínky,

kteře chrání jak spotřebitele, tak obchodníka. Z nich plyne povinnost prokázat splnění požadavků na jakost výrobků v oblastech jako je farmacie nebo automobilový průmysl. Speciální důraz je kladen v oblasti potravinářského průmyslu, kde je povinnost rozšířena také o tzv. dohledatelnost materiálu a jeho původu.

Obecně by bylo možné do trasování ve výrobě zahrnout celou škálu dat a ukazatelů. V rámci této práce je traceabilita rozdělena do tří oblastí získávání informací a to informace o materiálu, informace o výrobním postupu a doplňující informace.

2.1.1 Trasování materiálu

Z označení výsledného produktu musí být vždy možné dohledat, jaké vstupní suroviny byly použity pro jeho výrobu. Ukládání informací o kvalitě a původu materiálu je tak pro výrobce zásadní, ať už má rozsáhlou síť dodavatelů nebo pouze jednoho dodavatele.

Při trasování materiálu je důležité mít k dispozici informace jako jsou údaje o dodavateli, množství dovezeného a zpracovaného materiálu a informace o vlastnostech materiálu, které jsou sloučením základních, předpokládaných vlastností a podrobnějších výsledků z laboratoře výrobce.

Typickou situací vyžadující trasování materiálu je pozdě odhalená vada nebo nekvalita, která vyžaduje stažení produktu z trhu. Obdobně může nastat situace, kdy není chyba na straně výrobce, ale na straně dodavatele. Ten v případě svého pochybení díky svému logistickému systému zjistí, komu vadný materiál dodal a výrobce zase zjistí, jak s vadným materiálem naložil, trasování se tedy propojuje i napříč různými podniky. [7]

2.1.2 Trasování výrobního postupu

Další oblastí traceability je zaznamenávání průběhu výroby produktu. Zde se velmi prohlubují rozdíly mezi implementací v různých podnicích. Některé podniky nepotřebují zaznamenávat výrobní postup téměř vůbec, jiné jej naopak potřebují zaznamenat velmi podrobně. Podrobnější zaznamenávání je důležité zejména pro chemický a potravinářský průmysl, kde dochází k různým fyzickým a chemickým přeměnám a je vyžadována stálost podmínek (např. proces pasterizace vyžadující udržení dané teploty po určitou dobu). Přestože jsou tyto procesy ve většině případů řízené automaticky, působí zde řada vlivů (např. lidský faktor, selhání techniky), kvůli kterým se nelze spoléhat, že vše proběhlo dle doporučeného postupu.

Při trasování výroby je tedy nutné zaznamenat informace jako je zvolená receptura, nastavené požadované hodnoty, trendy sledovaných parametrů nebo také alarmní stavy zařízení, které během výroby šarže nastaly.[7]

2.1.3 Trasování - ostatní

Do ostatních údajů traceability jsou zahrnuty veškeré informace specifické pro daný podnik. Řadí se zde například trasování obsluhy dle směn nebo uživatelů přihlášených do supervizních aplikací, kteří nesou zodpovědnost za průběh výroby nebo její části. Co se tedy týče personálu, je vhodné zaznamenat identifikační údaje osoby, kontaktní údaje, jejich pozici v podniku a také jaká oprávnění k řízení výroby osoba má.

Důležité také může být zaznamenávání údajů o zařízeních. Podstatnými informacemi jsou zde např. jejich označení, parametry zařízení nebo kdy a kým byla provedena poslední údržba.[7]

2.2 Možnosti implementace traceability systému

Implementace systému na trasování výroby vyžaduje individuální přístup, neboť podnik od podniku se liší důležitost dat a požadavky na systém. Návrh takového systému na míru je ovšem finančně velmi nákladný a pro menší provozy má příliš malou návratnost. Z tohoto důvodu řada podniků využívá pro trasování materiálů a výroby papírové zprávy, vyplňované ručně operátory, vedoucími provozu apod. Toto řešení má řadu negativ, počínaje časem stráveným nad vyplňováním zprávy, přes skladování a organizaci nadbytečného množství dokumentů, po komplikované dohledávání chyb a jejich příčin. Připočte-li se ke všemu lidská chyba, je výsledkem strmý růst nákladů při řešení problémů a snížení konkurenceschopnosti.

Alternativou může být využití dostupných komerčních řešení nebo pořízení vlastního automatizovaného systému.

2.2.1 Automatizované systémy

Systém pro sledování výroby se většinou pořizuje jako součást MES systémů. Pro mnohé podniky je pořízení kompletního MES systému příliš nákladné, prodejci proto častěji nabízí modulární řešení, kdy si zákazník vybere, které funkcionality MES systému nejvíce potřebuje. Pro rozšíření systému sledování výroby by bylo možné pořídit např. skladový a logistický systém, systém pro automatické objednávání materiálu nebo takové plánování údržby.

Vzhledem k většímu toku dat musí výrobci těchto systémů optimalizovat jejich zpracovávání a ukládání, obzvláště, pokud má sběr dat probíhat v reálném čase. Dále je nutné zajistit dostatečný výkon hardwaru, který bude systém používat a poskytnout také dostatečně velký datový prostor pro databázi.

Zabezpečení

Při implementaci jakéhokoliv systému, který má přístup k informacím o výrobě, je nutné zvážit míru zabezpečení a zabránit tak možnému zneužití údajů nebo krádeži výrobního tajemství. Základním opatřením je udělování práv uživatelům systému a jejich kontrola při provádění různých úkonů. Stejně tak je nutné zabezpečit počítač, na kterém běží server s databází. Přídavnými opatřeními může být šifrování citlivých dat nebo použití bezpečnějšího komunikačního protokolu (např. OPC UA).

Značení produktů

Aby byl systém schopný sledovat pohyb materiálu a produktu, je důležité nějakým způsobem interpretovat jeho přemístění. V jednodušším případě zadává informace o pohybu obsluha systému ručně. Pro provoz operující s větším množstvím materiálu je na místě uvažovat o efektivnějších způsobu a to např. použití čárových kódů nebo jiných forem značení, které jsou při přemístování skenovány a automaticky zaznamenány do systému.

Modelování dat

V praxi jsou veškerá získaná data odesílána do databázového serveru. Uživatele systému trasování nemusí zajímat, jakým způsobem jsou data ukládána do databáze. Pro vývojáře systému je to ovšem zásadní, neboť struktura dat musí být navržena s ohledem na jejich očekávané množství a také s ohledem na formu, v jaké se data budou v případě potřeby číst.

Množství dat je důležité pro návrh modelu databáze, který by měl být tvořen z pohledu logického, fyzického i konceptuálního. Struktura tabulek by měla odpovídat aktuálním potřebám aplikace, ale také by měla počítat s možným rozšířením výroby v budoucnu.

Požadavky je nutné stanovit dopředu i pro čtení záznamů operátory v aplikaci. Některá data spolu mohou logicky souviset, ale pokud nebudou v databázi správně provázaná, nebude možné je v aplikaci správně zobrazit. Čtení záznamů může mít podobu tabulky nebo reportů v různých formátech (pdf, csv). Pokud by byla struktura databáze příliš komplikovaná, při vyčítání dat by mohlo dojít k zbytečnému zahlcení výkonu kvůli vykonávání složitých příkazů.

Data nemusí být čtena pouze operátory, ale mohou být předávána mezi dalšími podnikovými systémy. Aby spolu mohly správně komunikovat, je nutné formátovat data tak, aby jim rozuměli obě strany. Z tohoto důvodu je vhodné držet se při návrhu platných norem a standardů, které obsahují doporučení, jak by měly modely vypadat a jak by měly být vzájemně provázané. [8]

2.2.2 Budoucnost traceability systémů

Představu o blízké budoucnosti industriálních systémů máme díky iniciativě Industrie 4.0. Tato iniciativa slouží jako předpis pro podniky, čeho by měly ve svých provozech dosáhnout, aby si zajistily dlouhodobou konkurenceschopnost a dosáhly maximální možné efektivity výroby.

Jednou z úvah o budoucnosti průmyslu je opatření výrobku komunikačním zařízením, čímž řídí svůj životní cyklus sám tak, že komunikuje s výrobními stanovišti a vyjednává si nejefektivnější cestu skrze výrobní proces. Informace o výrobě produktu tak bude možné vyčíst přímo z produktu samotného. Své komunikační zařízení má stejně tak i výrobní jednotka. Díky tomu je možné sbírat data z jakékoliv perspektivy výroby. Pro systémy traceability to tedy znamená výrazné zvýšení objemu informací, které mohou být využity ke sledování procesů výroby nebo vyhledávání příčin nekvality.

Veškerá data by v budoucnu měla být sdílena přes cloudové servery, což by mohlo výrazně urychlit předávání informací mezi podnikovými systémy. Použitím cloudového řešení pro ukládání dat je možné se posunout i v oblasti bezpečnosti. Data nejsou uložena na fyzickém zařízení v provozu, ke kterému by se kdokoli mohl přihlásit. Data jsou také lépe chráněna proti jejich ztrátě, neboť cloudové servery standardně operují redundantně.

Stručně řečeno, budoucnost traceability systémů spočívá ve sběru většího množství dat a stejně tak v jejich efektivnějším využití nejen pro účely trasování výroby, ale obecně k optimalizaci výroby po všech stránkách.

2.3 Příklady dostupných řešení

Na českém i světovém trhu se nabízí velké množství MES systémů, z nichž většina zahrnuje i aplikaci nebo modul pro trasování výroby, materiálu a procesů. Tato řešení jsou ve většině případů škálovatelná a upravují se na základě parametrů zákazníka, aby se mu dostalo všech požadovaných funkcionalit, ale zároveň neplatil velké peníze za nadbytečné možnosti systému.

Jako příklad dostupných řešení jsou zde uvedeny aplikace COMES® Traceability, Act-in Traceability a Katana Product Traceability.

2.3.1 COMES® Traceability

Jedním z nejznámějších výrobních informačních systémů na českém trhu je MES systém COMES® vyvíjeným společností COMPAS Automatizace s.r.o.

Systém COMES vychází z mezinárodních standardů ANSI/ISA-95, ANSI/ISA-88, díky čemuž splňuje požadavky regulačních orgánů a může být použit ve farmacii a potravinářství. COMES je modulární systém a kromě modulu Traceability nabízí také modul Historian pro archivaci, Modeller pro modifikaci funkcí MES, Batch pro řízení dávkové výroby a Logon pro správu uživatelů.

COMES je vyvíjen pro operační systém Windows ve frameworku Visual Studio .NET a jako databázovou platformu využívá MS SQL server a webový server IIS. COMES je koncipován jako webový systém postavený na architektuře klient/server. Díky této technologii není nutné pro přístup do systému instalovat náročný software.

Modul Traceability byl vyvinut pro trasování výroby z hlediska vstupního materiálu, meziproductů a konečných šarží. Trasováním se tvoří rodokmen výrobku, díky kterému je možné zpětně dohledat potřebné informace v případě odhalení problému. Modul umožňuje také dopředné vyhledávání historie pro nalezení všech možných šarží a produktů vyrobených z poškozeného nebo jinak nevhodného materiálu.

Důležitou vlastností modulu je také sledování veškerých výrobních operací s materiálem a ve spojení s modulem Modeller je možné vytvořit detailní výrobní protokoly, včetně bilancí materiálu [10]. Příklad podobného protokolu z výroby je uveden na obrázku 2.2.

Protokol traceability						1.8.2006 8:00:00
						1.8.2006 12:00:00
			Cíl	Množství	Start	Stop
			N873	721.800	01.08.2006 07:14:09	01.08.2006 16:09:09
			N873	468.000	01.08.2006 08:01:09	01.08.2006 11:56:09
Lot míchané		Zdroj	Cíl	Množství	Start	Stop
054_A10601102		H740	S803	200.278	01.08.2006 04:11:41	01.08.2006 04:21:18
	Lot suroviny	Zdroj	Cíl	Množství	Start	Stop
	11701_A_344/06	S700	H740	76.351	01.08.2006 04:13:32	01.08.2006 04:14:25
	11702_A_337	N702	H740	21.599	01.08.2006 04:11:41	01.08.2006 04:12:18
	11703_A_336	N704	H740	100.424	01.08.2006 04:15:01	01.08.2006 04:19:36
	11721_A10601102_2316		H740	1.000	01.08.2006 04:21:18	01.08.2006 04:21:18
	11725_A10601102_2315		H740	0.600	01.08.2006 04:20:57	01.08.2006 04:20:57
Lot míchané		Zdroj	Cíl	Množství	Start	Stop
054_A10601103		H740	S822	200.082	01.08.2006 04:27:04	01.08.2006 04:37:45
	Lot suroviny	Zdroj	Cíl	Množství	Start	Stop
	11701_A_344/06	S700	H740	76.098	01.08.2006 04:28:50	01.08.2006 04:29:39
	11702_A_337	N702	H740	21.758	01.08.2006 04:27:04	01.08.2006 04:27:41
	11703_A_336	N704	H740	100.254	01.08.2006 04:30:14	01.08.2006 04:34:41
	11721_A10601103_2318		H740	1.000	01.08.2006 04:37:45	01.08.2006 04:37:45
	11725_A10601103_2317		H740	0.600	01.08.2006 04:37:25	01.08.2006 04:37:25
Lot míchané		Zdroj	Cíl	Množství	Start	Stop
054_A10601104		H740	S818	199.943	01.08.2006 05:42:17	01.08.2006 05:52:12
	Lot suroviny	Zdroj	Cíl	Množství	Start	Stop
	11701_A_344/06	S700	H740	76.203	01.08.2006 05:44:07	01.08.2006 05:45:02
	11702_A_337	N702	H740	21.345	01.08.2006 05:42:17	01.08.2006 05:42:53
	11703_A_336	N704	H740	100.258	01.08.2006 05:45:36	01.08.2006 05:50:12
	11721_A10601104_2320		H740	1.000	01.08.2006 05:52:12	01.08.2006 05:52:12
	11725_A10601104_2319		H740	0.600	01.08.2006 05:51:51	01.08.2006 05:51:51

Obr. 2.2: Ukázka traceability systému společnosti COMES® [10]

2.3.2 Act-in Traceability

Dalším ze silných zástupců MES systémů na českém trhu je systém společnosti Act-in. Součástí tohoto MES systému je aplikace Traceability pro monitorování kompletního průběhu výroby produktu. Aplikace je modulární, může tedy fungovat i samostatně.

Hlavními funkcemi aplikace Traceability je například evidence pracovníků, technologických parametrů a vstupního materiálu, sledování výrobku a výrobních dávek, identifikace výrobků a zpětná dohledatelnost výrobních kroků a parametrů. To vše tvoří rodokmen výrobku. Dalšími významnými vlastnostmi je kontrola dodržení předepsaných postupů a provedených operací. Ze zmíněných dat aplikace tvoří základ pro BigData analýzy a Product LifeCycle Management (funkce řízení životního cyklu produktu).

Pro vzájemnou komunikaci prvků systému a jejich rozpoznání slouží řada implementovaných technologií jako QR kódy, RFID, NFC a jiná IoT řešení. Z komunikačních protokolů podporuje např. RS232/422/485, sběrnice Fieldbus až po nejmodernější řešení OPC UA.

Oproti konkurenčním aplikacím je tato zajímavá díky možnosti napojení na moderní technologie typu chytré brýle nebo rozšířená realita [9].

The screenshot displays the Act-in Traceability software interface. The main window is titled 'Bestand Batch Product Tabellen' and shows a search bar with the ID '510101010000170124104202'. Below the search bar, there are four columns: Retainer, Duct, Shell, and Eindproduct, each with a corresponding ID. A table below lists the production steps with their names and timestamps:

Name	Timestamp
IM Retainer	24-1-2017 10:42:00
IM Duct	1-2-2017 23:22:00
Skinalush	2-2-2017 16:27:00
Skisscore	2-2-2017 16:34:53
Flaming	2-2-2017 16:26:31
Foam	2-2-2017 16:54:30
Miling	2-2-2017 18:05:56
Welding	2-2-2017 18:13:15

Below this table, there are sections for 'Process parameters' and 'Batch Info'. The 'Batch Info' section shows a table with 'Geldig van' and 'Geldig tot' columns. At the bottom, there is a table with 'Geldig van', 'Geldig tot', 'Batch No.', and 'Article No.' columns.

On the right side of the interface, there is a photograph of a factory floor. A blue machine with a 'LABEL' sign is visible, along with a printer producing labels. The background shows industrial equipment and a worker.

Obr. 2.3: Ukázka traceability systému společnosti Act-in [9]

2.3.3 Katana Product Traceability

Velmi mladou společností je Katana, založená v Estonsku, jejichž rychle se rozvíjející produkt Katana Smart Manufacturing Software působí jako univerzální nástroj nabízející nejnútnejší funkcionalitu z různých podnikových úrovní jako plánování výroby, nákup, nákladovost produktu, vyřizování objednávek, přidělování úkolů pracovníkům a nebo správa inventáře, pod kterou spadá sledování šarže. Sledování šarže není modulární, pořizuje se jako doplněk celého systému.

Šarže je trasována z hlediska dávky materiálu a konečného produktu, software do trasování nezahrnuje proces výroby a výrobní parametry, kromě celkové doby výroby, z čehož lze vydedukovat, že je aplikace vhodná zejména diskretní výrobu a malovýrobu, kde jsou důležitějšími parametry počty kusů na skladě. Zajímavou vlastností je upozornění na blížící se dobu spotřeby, což umožňuje upřednostnit spotřebu materiálu nebo urychleného prodeje dané dávky.

Software stojí na cloudovém řešení a pro přístup využívá webový prohlížeč. Pořízení systému funguje formou paušálních plateb odvíjejících se od počtu uživatelů a rozsahu výroby [11] [12].

Inventory Intel
[P-101] Lasersaber / blue Main warehouse

In stock	Expected	Committed	Reorder	Missing/Excess
20	6	12	6	8

Movement	Caused	Batch #	Change	Cost per unit	Balance after	Value in stock
14. June	#SO-125	B-1020	-4	20 \$	3	60 \$
10. June	#MO-52	B-1020	5	20 \$	7	140 \$
8. June	#SO-123	B-1018	-1	20 \$	2	40 \$
7. June	#SO-120	B-1018	-2	20 \$	3	60 \$
2. June	#MO-51	B-1018	5	20 \$	5	100 \$

Obr. 2.4: Ukázka traceability systému společnosti Katana [12]

3 Modul trasování šarží

Cílem práce je navrhnout a implementovat modul trasování šarží do systému MagmaView, který odpovídá řízení výroby na úrovni SCADA. Smyslem modulu je usnadnit vyhledávání vazeb mezi materiálem a konečným produktem v obou směrech a také poskytnout lepší přehled o průběhu výroby.

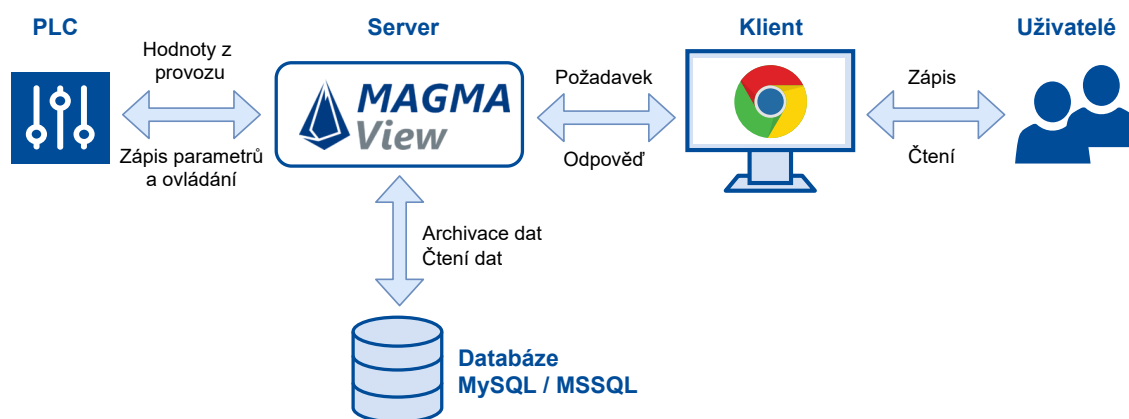
3.1 Vizualizace MagmaView

Systém MagmaView je mladá, moderní vizualizace postavená na architektuře klient/server, kde MagmaView zastupuje webserver poskytující data klientovi, zpravidla webovému prohlížeči Google Chrome. Díky tomu je umožněn přístup více uživatelů současně a to v jedné síti nebo vzdáleně např. prostřednictvím VPN.

Vzhledem k využití webserveru je aplikace multiplatformní, lze tedy spustit nezávisle na použitém hardwaru nebo operačním systému.

Vizualizace využívá pro sběr a archivaci dat SQL databázi a je přizpůsobena pro použití s MySQL nebo MSSQL databázovým systémem. Ukládaná data slouží pro přehled o alarmních stavech nebo zobrazení trendů parametrů.

Systém komunikuje s PLC systémy Siemens prostřednictvím protokolu S7 a dokáže také komunikovat s jakýmkoliv jiným řídicím systémem schopným využívat rozhraní OPC UA [13].



Obr. 3.1: Blokový diagram systému MagmaView [13]

3.2 Požadavky na modul trasování šarží

Přestože se pojem "šarže" obvykle používá ve spojení s dávkovou výrobou, v praxi to nemusí nutně znamenat, že výroba, splňující teoretické představy o dávkové výrobě,

je opravdu jako dávková řízena. V rámci této práce se jako pojem šarže chápe určité množství materiálu ve výrobě nebo množství konečného produktu.

Vzhledem k tomu, že se materiál může různě slučovat nebo rozdělovat na menší dávky, je obtížné vést jeden cyklus výrobního procesu jako jednu šarži. Pomyslně je možné šarže rozdělit na dodavatelskou, výrobní a expediční.

Dodavatelská šarže se používá pro označení dávek surového materiálu. Tento typ šarže vyžaduje zaznamenat především informace o dodavateli a o kvalitě materiálu. Expediční šarže představuje dávku hotového produktu, který je připraven k expedici nebo již expedován. V tomto případě se zaznamenávají především informace o odběrateli. Požadavkem na modul je schopnost propojit tyto dva druhy šarží a dopátrat se od jednoho k druhému.

Další požadavek se týká posledního typu šarže, šarže výrobní, a tím je mapování umístění šarže ve výrobním procesu. Umístěním je chápáno výrobní zařízení nebo jiná jednoznačně daná pozice ve výrobě, kde se může zpracovávaný materiál nacházet. Požadované jsou také časové údaje odkdy dokdy se šarže v daném umístění nacházela. Historii umístění je nutné archivovat pro případ potřeby.

Co se týče potřeb operátorů, pro správu šarží musí být modul přístupný z vizualizace a měl by umožnit jak vytváření nových šarží, tak editovat ty existující a zobrazit si jejich detail a historii.

Posledním požadavkem na modul je využít pro návrh aktuální standardy platné v procesním průmyslu.

3.3 Fáze vývoje modulu

Návrh jakéhokoliv softwaru je vhodné podrobně promyslet a zahrnout do něj všechny možné situace, které by v aplikaci mohly nastat, čímž se předchází pozdnímu odhalení nedostatků, nesouladů a mnohdy pracným zásahům do kódu. Vývoj modulu pro trasování šarží lze rozdělit do několika fází:

- **Návrh datového modelu** - Cílem návrhu je ujasnit, s jakými informacemi bude modul pracovat, odkud tyto informace budou pocházet a v jaké formě budou uloženy. Návrhu datového modelu se věnuje kapitola 4.
- **Návrh GUI modelu a definice interakce s uživatelem** - Popis interakce s uživatelem pomáhá odhalit všechny uživatelské situace, které je nutné ošetřit. Součástí je i návrh oken ve vizualizaci. Fáze je popsána v kapitole 5.
- **Implementace** - Z návrhu datového a GUI modelu lze vytušit, jaké funkcionality bude v kódu potřeba zahrnout. Návrhy tedy slouží jako předpis pro vývoj. Implementace není zahrnuta do této semestrální práce.
- **Testování** - Testování má za cíl odhalit neošetřené situace a také zhodnotit uživatelskou zkušenost. Testování není zahrnuto do této semestrální práce.

4 Návrh datového modelu

Před softwarovou implementací je nutné znát, jaké informace bude nutné ukládat do databáze, odkud se tyto informace budou čerpat a v jaké podobě. Tyto znalosti poskytuje datový model, který byl navržen formou vzájemně propojených databázových tabulek. V kapitole jsou postupně popsány jednotlivé tabulky a jejich vzájemné vztahy. Kompletní model, včetně vazeb mezi tabulkami, je součástí přílohy A.

4.1 Využití standardu ISA-95 při návrhu databáze

Pro návrh datového modelu byla využita B2MML implementace standardu ISA-95, která se skládá ze sady XSD schémat odpovídající objektovým modelům, které jsou součástí standardu ISA-95 (více v kapitole 1). Objektové modely a schémata by měly být schopné pokrýt i ty nejrozsáhlejší provozy. V případě tohoto modulu šarží je vyžadováno použití pouhého zlomku dostupné inspirace, neboť se neočekává použití modulu pro větší provozy. Použití standardu by však mělo výrazně usnadnit další rozšiřování modulu.

Jakým způsobem a jaká část schémat byla použita pro návrh modelu databáze je popsáno pro každou tabulku zvlášť. Nestandardní a doplňující tabulky se odlišují červeným záhlavím na obrázku.

4.1.1 Porovnání přístupů k návrhu

Existuje více způsobů jak aplikovat standard při návrhu databáze. Ten první lze nazvat "úplný", kdy se pro návrh využijí všechny modely a schémata, ačkoliv by mohly zůstat nepoužité, tedy bez vyplněných hodnot při provozu modulu. Týká se to všech pevně daných vlastností v modelech např. jedinečné identifikátory nebo názvy.

Hlavní tabulka				Doplňující tabulka		
ID	Sloupec 1	Sloupec 2	Vlastnosti	ID	Vlastnost 1	Vlastnost 2
			A			
			B			
			C	A		
			A			
				B		
				C		

Obr. 4.1: "Úplný" databázový model

V modelech se však vyskytují i dynamické entity např. typu Property nebo Description. Počet instancí těchto entit je neomezený, není tedy možné pro ně v databázi předem připravit sloupce v tabulkách. Tyto dynamické entity by měly mít vlastní tabulky bez definovaných sloupců, kde hodnoty představují jednotlivé záznamy (řádky) a není tím nijak omezen jejich počet (znázorněno na obrázku 4.1). Na druhou stranu může model obsahovat zbytečně velký počet tabulek a v záznamech již není možné orientovat se jinak, než s použitím vizualizace.

Další princip návrhu by bylo možné nazývat "**redukovaný**". Zde už bereme ohled na budoucí použití aplikace a reálné potřeby. Z datového modelu se tedy vynechají entity, které by zůstaly nevyužity. Co se týče dynamických entit, místo neomezeného počtu záznamů ve vlastní tabulce se předem stanoví jasný počet entit, realizovaných jako sloupce v nadřazené tabulce (znázorněno na obrázku 4.2). Redukuje se tedy počet potřebných tabulek a databáze je přehlednější. Nevýhodou tohoto přístupu je náročnější zpracovávání databáze v případě rozšiřování modulu.

Hlavní tabulka					Eliminovaná tabulka		
ID	Sloupec 1	Sloupec 2	Vlastnost 1	Vlastnost 2	ID	Vlastnost 1	Vlastnost 2

Obr. 4.2: "Redukovaný" databázový model

Poslední zvažovaný přístup k návrhu modelu lze nazvat "**kombinovaný**". Tento přístup byl použit pro návrh níže popisovaného datového modelu. Vzhledem k tomu, že se dá přibližně odhadnout jaká možná rozšíření by mohla být v budoucnu požadována a stejně tak, kolik informací bude v tabulkách ukládáno, je pro návrh každé tabulky zváženo, jaký přístup bude vhodnější.

4.2 Databázové tabulky pro číselníky

Na nejnižší úrovni databázového modelu jsou tabulky pro entity, které ve vizualizaci uživatel používá jako číselníky. Jedná se o umístění šarže ve výrobě (tabulka Equipment), typy materiálu (tabulka MaterialClass), definice materiálu (tabulka MaterialDefinition) a inženýrské jednotky (tabulka Eng_Units). Tabulka Person, představující uživatele, není určena pro použití s číselníky, neboť se přihlášený uživatel zaznamenává automaticky, ale platí pro ni a stejně tak pro ostatní tabulky, že se u nich neočekává velký počet záznamů, ani jejich časté úpravy.

4.2.1 Obecný popis tabulky

Pro zjednodušení budou všechny obrázky databázových tabulek popsány nejprve obecně. Jelikož byly tabulky inspirovány standardem, jsou pro zvýraznění podobnosti s modely uvedeny všechny názvy v anglickém jazyce.

Každá tabulka má v záhlaví uveden svůj název. Budoucí sloupce v tabulce představují jednotlivé řádky. První řádek tabulky je vždy identifikátor. V každém řádku jsou nejprve uvedeny typické vlastnosti (legenda je součástí přílohy A) pro danou entitu. Následuje, modrou barvou zvýrazněný, skutečný název sloupce v tabulce, mající větší vypovídající hodnotu o elementu, a jeho datový typ. Poslední informací, uvedenou nad názvem sloupce, je název atributu z B2MML schématu, kterým byl sloupec inspirován.

4.2.2 Tabulka Person

V tabulce jsou zaznamenány informace o uživateli, kteří budou mít přístup k modulu trasování šarží. Každému uživateli je přiděleny přihlašovací údaje (login a heslo) pro přihlášení do systému. Dále má každý uživatel přidělena přístupová práva, dle kterých mu jsou povoleny zásahy do systému. Doplňujícími údaji je jméno a příjmení uživatele a také stav.

Záznamy v tabulce se vytvářejí ručně, obsluhou ve vizualizaci nebo na žádost zákazníka ještě před uvedením do provozu.

Person	
PK,NN	<u>Person_ID: int</u>
	PersonName Person_Name: VARCHAR (50)
NN, UQ	PersonProperty Person_Login: VARCHAR (50)
NN	PersonProperty Person_Password: VARCHAR (50)
NN	PersonellClassID Access_ID: int
NN	PersonProperty Person_State: VARCHAR(50)

Obr. 4.3: Databázová tabulka s uživateli

Pro návrh tabulky bylo použito B2MML schéma Personnel, třída Person. Na obrázku 4.3 lze vidět, jaké atributy třídy byly použity pro jednotlivé sloupce tabulky.

4.2.3 Tabulka Equipment

Tabulka Equipment slouží k definici možných umístění šarže ve výrobě. Obsahem tabulky jsou pouze identifikátory záznamů a název zařízení/umístění.

Záznamy v tabulce se vytvářejí ručně, obsluhou ve vizualizaci nebo na žádost zákazníka ještě před uvedením do provozu.

Equipment	
PK,NN	<u>Equipment_ID: int</u>
NN	EquipmentProperty Equipment_Name: VARCHAR (50)

Obr. 4.4: Databázová tabulka výrobních zařízení (umístění)

4.2.4 Tabulka ResourceNetwork

Pro zaznamenání údajů o dodavateli nebo odběrateli slouží tabulka ResourceNetwork. V tabulce jsou obsaženy převážně administrativní a kontaktní informace jako např. IČO, adresa, telefon nebo web. Dále je zde uveden povaha vzájemného vztahu (odběratel/dodavatel) a stav vztahu (aktivní/neaktivní).

ResourceNetworkConnection	
PK,NN	<u>ResourceNetworkConnection_ID: int</u>
NN	ResourceNetworkConnectionProperty Resource_Name: VARCHAR (50) ResourceNetworkConnectionProperty Resource_ICO: VARCHAR (50) ResourceNetworkConnectionProperty Resource_DIC: VARCHAR (50) ResourceNetworkConnectionProperty Resource_Adress: VARCHAR (100) ResourceNetworkConnectionProperty Resource_Phone: VARCHAR (50) ResourceNetworkConnectionProperty Resource_WEB: VARCHAR (100) ResourceNetworkConnectionProperty Resource_State: VARCHAR (100) ResourceNetworkConnectionProperty Resource_Relationship: VARCHAR (100)

Obr. 4.5: Databázová tabulka dodavatelů/zákazníků

Této oblasti informací odpovídá standardní B2MML schéma ResourceRelationshipNetwork, třída ResourceNetworkConnection. Kromě identifikátoru není vyžadována žádná pevně daná informace, proto jsou všechny sloupce tabulky typu Property. Záznamy v tabulce se vytvářejí ručně, obsluhou ve vizualizaci nebo na žádost zákazníka ještě před uvedením do provozu.

4.2.5 Tabulka MaterialClass

Tabulka MaterialClass obsahuje definice třídy materiálu (např. cukr, pomeranč). Záznam typu materiálu obsahuje identifikátor a název třídy. Tabulce MaterialClass odpovídá v B2MML schématu Material třída MaterialClass. Záznamy v tabulce se vytvářejí ručně, obsluhou ve vizualizaci nebo na žádost zákazníka ještě před uvedením do provozu.

MaterialClass	
PK,NN	MaterialClass_ID: int
NN	MaterialClassProperty MaterialClass_Name: VARCHAR (50)

Obr. 4.6: Databázová tabulka tříd materiálu

4.2.6 Tabulka MaterialDefinition

Pro definici materiálu (např. třtinový cukr, stéviový cukr) se využívá tabulka MaterialDefinition, která ve standardním B2MML schématu Material odpovídá třídě MaterialDefinition. Kromě identifikátoru je v tabulce název definice, odpovídající třídě Property a odkaz na třídu materiálu MaterialClassID, viz. předchozí odstavec.

MaterialDefinition	
PK,NN	MaterialDefiniton ID: int
NN	MaterialDefinitionProperty Material_Name: VARCHAR (50) MaterialClassID MaterialClass_ID: int

Obr. 4.7: Databázová tabulka definice materiálu

Po uvedení systému do provozu budou součástí tabulky další elementy typu Property, které budou blíže specifikovat vlastnosti materiálu, např. hodnoty z laboratoře.

Záznamy v tabulce se vytvářejí ručně, obsluhou ve vizualizaci nebo na žádost zákazníka ještě před uvedením do provozu.

4.3 Databázové tabulky pro zaznamenávání množství materiálu

Následující skupina tabulek slouží pro zaznamenávání množství materiálu. Zaznamenání množství probíhá dvěma způsoby. Ten první je při zakládání šarže operátorem ve vizualizaci, kde zadává množství ručně. Druhý způsob je automatický, kdy se množství materiálu čte z předepsaných receptur nebo z řídicích systémů. Záznamy množství však vypadají shodně a jsou společně uloženy v tabulce MaterialLot_Quantity, s odkazem na inženýrskou jednotku v tabulce EngineeringUnits. Na záznam množství se odkazuje záznam v MaterialRatio nebo záznam šarže.

4.3.1 Tabulka MaterialLotQuantity

Tabulka slouží k zaznamenání množství materiálu a byla navržena dle B2MML schématu Common, třída Quantity. Záznam se skládá z identifikátoru, hodnoty reprezentující množství a odkazu na inženýrskou jednotku.

MaterialLot_Quantity	
PK,NN	<u>LotQuantity_ID: int</u>
NN	Quantity Quantity: int
NN	UnitOfMeasure Eng_Unit_ID: int

Obr. 4.8: Databázová tabulka množství materiálu

4.3.2 Tabulka EngineeringUnits

Nestandardní tabulka EngineeringUnits slouží k předdefinování možných inženýrských jednotek. Důvod k vytvoření tabulky je zachování jednotného způsobu zápisu jednotky operátory (např. kus/ks). Tabulka obsahuje identifikátor záznamu, na který se odkazuje v tabulce MaterialLotQuantity, název inženýrské jednotky,

který bude zobrazen v číselníku ve vizualizaci a případně popis jednotky. Záznamy v tabulce se vytvářejí ručně, obsluhou ve vizualizaci nebo na žádost zákazníka ještě před uvedením do provozu.

EngineeringUnits	
PK,NN	<u>Eng_Unit_ID: int</u>
NN	Unit: VARCHAR(50)
NN	Unit_description: VARCHAR(50)

Obr. 4.9: Databázová tabulka inženýrských jednotek

4.4 Databázové tabulky pro trasování šarží

Poslední skupina z návrhu datového modelu je jádro trasovacího systému. Jedná se o tabulky, ve kterých jsou zaznamenány šarže (tabulky MaterialLot, ProductLot), jejich historie ve výrobě (tabulka JobResponse) a jejich vzájemné propojení (tabulka MaterialRatio).

4.4.1 Tabulka MaterialLot

Nejsložitější tabulkou šaržové části databáze je tabulka MaterialLot, která slouží pro zaznamenání všech šarží nezpracovaného materiálu a pomocných výrobních šarží. Pro každý záznam platí, že má jméno, číslo, čas a uživatele k založení šarže, čas a uživatele k poslední změně šarže a stav šarže. Ostatní údaje však závisí na tom, kde se v hierarchii výroby šarže nachází.

V případě šarže, kdy se jedná o dovezenou dávku materiálu, se ukládá i dodavatel a množství materiálu. Pole JobResponse_ID zůstává nevyplněno.

V případě šarže, která je dávkou putující ke zpracování a může být složena z více druhů materiálu, zůstává pole pro dodavatele a množství nevyplněno a naopak se vyplňuje pouze pole JobResponse_ID.

JobResponse_ID je tzv. cizí klíč¹, který si lze představit jako spojení hodnot v tabulkách. V této tabulce funguje cizí klíč jako rodič (výrobní dávka), v tabulce MaterialRatio může být k tomuto klíči přidruženo několik hodnot (složky dávky). Výrobní dávka (A) se poté může smíchat s jinou výrobní dávkou (B).

¹Cizí klíč (Foreign key) - Pole v tabulce, které jednoznačně identifikuje řádek jiné nebo stejné tabulky[?]

Vznikne další záznam v tabulce MaterialLot (dávka C) a cizí klíč bude ukazovat do tabulky MaterialRatio, kde k onomu klíči bude přiřazena dávka A a dávka B. Tabulka byla navržena dle třídy MaterialLot v B2MML šabloně Material.

MaterialLot	
PK,NN	<u>MaterialLot_ID: int</u>
NN	MaterialLotProperty Lot_Number: int
NN	MaterialLotProperty Lot_Name: VARCHAR(50)
	MaterialDefinitionID MaterialDefinition_ID: int
	MaterialLotProperty Supplier_ID: int
	MaterialLotProperty JobResponse_ID: int
NN	MaterialLotStatus Lot_State: VARCHAR(50)
	MaterialLotQuantity LotQuantity_ID: int
NN	MaterialLotProperty Time_Insert: DateTime
	MaterialLotProperty Time_Update: DateTime
	MaterialLotProperty Insert_Person_ID: int
	MaterialLotProperty Update_Person_ID: int
	Description Note: VARCHAR (200)

Obr. 4.10: Databázová tabulka výrobní šarže

4.4.2 Tabulka MaterialRatio

Pomocná tabulka MaterialRatio slouží jako navigační tabulka pro propojení souvisejících šarží. Ukládají se zde dávky materiálu, ze kterých je složená výrobní šarže. S nadřazenou šarží jsou dávky spojeny pomocí cizího klíče, jak bylo popsáno výše. Ke každé dávce materiálu je kromě cizího klíče přiřazen ukazatel umístění v tabulce JobResponse a ukazatel množství v tabulce MaterialQuantity.

MaterialRatio	
PK,NN	MaterialRatio_ID: int
FK,NN	MaterialLot_ID: int
FK,NN	JobResponse_ID: int
	LotQuantity_ID: int

Obr. 4.11: Databázová tabulka poměru materiálu výrobní šarže

4.4.3 Tabulka ProductLot

Hotové dávky produktu, které jsou uloženy na skladě nebo již odevzdány odběrateli, se zaznamenávají do tabulky ProductLot. Důvod pro oddělení od zbylých záznamů šarží je čistě praktický, neboť je potřeba méně údajů a také nemusíme rozlišovat mezi dodavatelem a zákazníkem.

Tabulka vzešla z tabulky MaterialLot, ze které se odstranily nepotřebné sloupce.

ProductLot	
PK,NN	ProductLot_ID: int
NN	MaterialLotProperty Lot_Number: int
NN	MaterialLotProperty Lot_Name: VARCHAR(50)
NN	MaterialDefinitionID MaterialLot_ID: int
NN	MaterialLotProperty Customer_ID: int
NN	MaterialLotStatus Lot_State: VARCHAR(50)
	MaterialLotQuantity LotQuantity_ID: int
	MaterialLotProperty TimeStamp: DateTime
	MaterialLotProperty Person_ID: int

Obr. 4.12: Databázová tabulka produktové šarže

4.4.4 Tabulka JobResponse

Historie umístění šarže je zaznamenána v tabulce JobResponse. Tabulka byla navržena na základě největší podobnosti s očekávanými informacemi dle třídy JobResponse v B2MML schématu WorkPerformance. Záznam se skládá z identifikátoru, odkazu na šarži, odkazu na umístění, dvou časových údajů odkdy dokdy se šarže v daném umístění nacházela a z poznámky.

Očekává se, že záznamy budou přidávány především automaticky, dle událostí v řídicím systému.

JobResponse	
PK,NN	<u>JobResponse_ID: int</u>
NN	MaterialActual MaterialLot_ID: int
NN	EquipmentActual Equipment_ID: int
	StartTime Time_START: DateTime
	StopTime Time_STOP: DateTime
	Description Note: VARCHAR (200)

Obr. 4.13: Databázová tabulka historie umístění šarže

5 Návrh GUI modulu a popis interakce s uživatelem

Po navržení datového modelu je další fází navrhnout GUI modul a specifikovat, jakým způsobem bude probíhat interakce s obsluhou systému. Před vytvářením samotného designu vizualizace je důležité prvně specifikovat, jaké aktivity se od uživatele očekávají a jaký je rozsah uživatelských možností. Pro specifikaci a oddělení funkcionalit modulu trasování byly vytvořeny dva diagramy užití pro **správu šarží** a **editaci číselníků**.

Před rozбором návrhu bude vysvětleno pojetí několika pojmů používaných při popisu:

- **Diagram užití** - Používá se pro zjištění všech možných případů užití systému z pohledu uživatele. Diagram se skládá z případů užití, aktérů (postav provádějící akce) a vztahy mezi nimi [5].
- **Diagram aktivit** - Používá se pro modelování procesu. Sestává z posloupnosti úkonů, které se vykonají za účelem splnění uživatelské požadavky [5].
- **Číselník** - List s předdefinovanými možnostmi, ze kterého uživatel vybere vždy pouze jednu možnost.
- **Formulář** - Okno ve vizualizaci, které obsahuje prvky pro vyplnění údajů uživatelem (číselníky, zadání textu).
- **Layout** - Předem stanovené grafické rozložení prvků, které se při určité události načte ze systému. Prvky v rozvržení mají přidružené dynamické proměnné, jejichž hodnoty se nastavují při načítání layoutu.

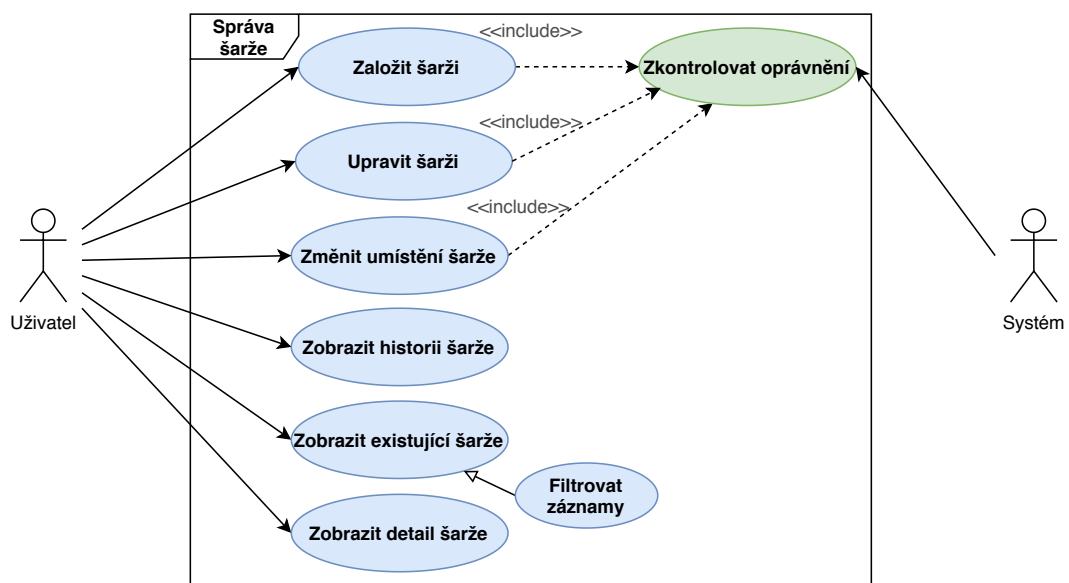
5.1 Správa šarží

Prvním z diagramů užití je diagram pro správu šarží a je znázorněn na obrázku 5.1. Na obrázku lze vidět dva aktéry, uživatele a systém. Správou šarží jsou myšleny jakékoliv akce zpřístupněné uživateli, které se týkají sledování dávky ve výrobě. Význam akcí je následující:

- **Založit šarži** - Uživatel chce zaznamenat dovoz materiálu. Akce vyžaduje kontrolu oprávnění systémem.
- **Upravit šarži** - Uživatel chce upravit údaje o již existující šarži. Vyžaduje kontrolu oprávnění systémem.
- **Změnit umístění šarže** - Hlášení o změně pozice šarže ve výrobě. Aktér závisí na povaze provozu, akce bude prováděna buď uživatelem nebo automaticky dle událostí v PLC, z tohoto důvodu není dále specifikováno uživatelské

prostředí pro tuto funkcionalitu. V případě zadávání přesunu operátorem je vyžadována kontrola oprávnění systémem.

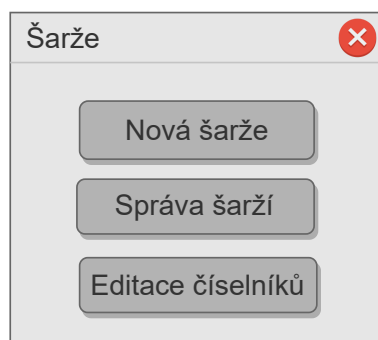
- **Zobrazit historii šarže** - Uživatel chce zobrazit historii operací s konkrétní šarží nebo její umístění ve výrobních úsecích.
- **Zobrazit existující šarže** - Uživatel chce zobrazit seznam existujících šarží. Rozšířením je funkce **Filtrovat záznamy**.
- **Zobrazit detail šarže** - Uživatel chce zobrazit detailní informace o zvolené šarži.
- **Zkontrolovat oprávnění** - Každý registrovaný uživatel má přiřazená určitá práva. Při úpravách a zásahu do systému je v rámci bezpečnosti nutné zkontrolovat, že má uživatel potřebné oprávnění pro provádění změn.



Obr. 5.1: Diagram užití Správa šarže

5.1.1 Menu pro správu šarží

Pro přístup k modulu šarží z vizualizace bylo navrženo menu viz. obrázek 5.2, které by mělo být přístupné z hlavní technologické obrazovky. Menu je realizováno jako popup okno a obsahuje tři tlačítka: Nová šarže, Správa šarží a Editace číselníků. Tlačítko **Nová šarže** otevře popup okno Založení nové šarže (viz.5.1.2). Tlačítko **Správa šarží** otevře okno zobrazující všechny existující šarže (viz. 5.1.5). Tlačítko **Editace číselníků** otevře menu pro editaci číselníků, čemuž se blíže věnuje kapitola 5.2.



Obr. 5.2: Návrh menu pro správu šarží

5.1.2 Založení nové šarže

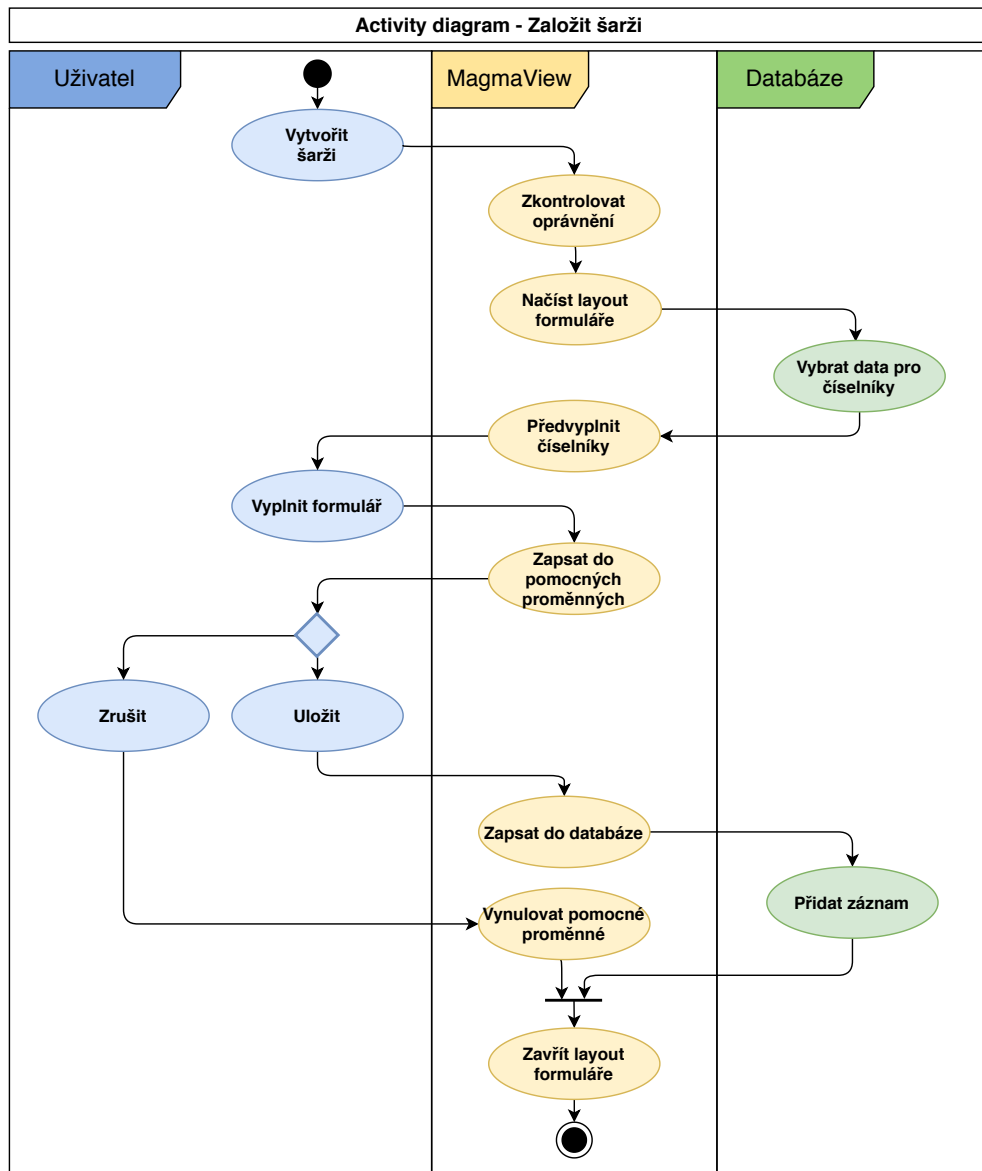
Při stisknutí tlačítka Nová šarže z menu se otevře popup okno znázorněné na obrázku 5.3. V okně se nachází několik uživatelsky vyplnitelných prvků jako název šarže nebo číslo šarže, a několik číselníků, např. výběr dodavatele nebo výběr typu zboží. V záhlaví okna se nachází dvě klikatelné ikony, křížek, pro zrušení zakládání nové šarže a disketa, symbolizující uložení nového záznamu. Ze systému se automaticky načítá aktuální datum a přihlášený uživatel. Tyto informace budou uloženy spolu s ostatními uživatelskými vstupy. K nové šarži je možné připojit libovolnou poznámku.

A screenshot of a "Nová šarže" (New order) popup window. The window has a title bar with a save icon and a close button. The main area contains several form elements: "Název šarže" (Order name) and "Číslo šarže" (Order number) are text input fields; "Datum" (Date) is a button showing "19.07.2020"; "Přihlášený uživatel" (Logged-in user) is a button showing "admin"; "Dodavatel" (Supplier), "Zboží" (Goods), and "Typ zboží" (Goods type) are dropdown menus; "Množství" (Quantity) and "Měrná jednotka" (Unit of measure) are text input fields; and "Poznámka" (Note) is a large text area.

Obr. 5.3: Popup okno pro založení nové šarže

Proces Založení nové šarže je znázorněn pomocí diagramu aktivity na obrázku 5.4. Uživatel vytvoří požadavek na založení nové šarže (stiskne tlačítko v menu). Aplikace nejprve zkontroluje uživatelské oprávnění k založení šarže. Pokud má dostatečná práva, načte se předpřipravený layout formuláře k vyplnění. Při načítání

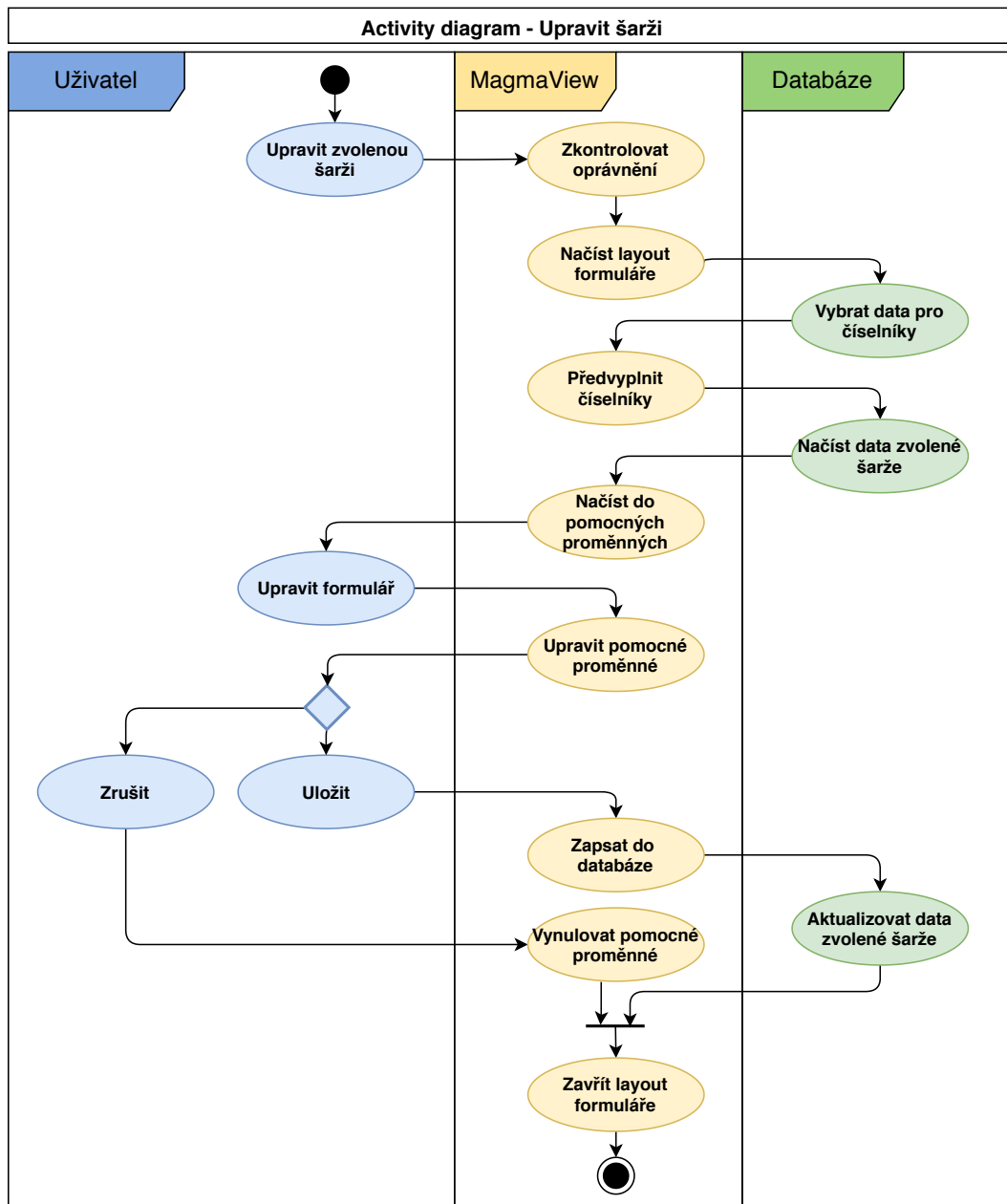
layoutu se rovněž vyplní dříve popsané číselníky, jejichž entity se vyčtou z databázových tabulek. Uživatel vyplní prvky ve formuláři a tyto uživatelské vstupy se zapíší do pomocných proměnných. Po vyplnění má uživatel možnost zrušit založení nové šarže (v záhlaví okna), čímž se vymažou hodnoty v pomocných proměnných a formulář se zavře. Druhou možností uživatele je záznam uložit. Uložení záznamu systém připraví data pro zápis do databáze a následně se uloží do příslušných databázových tabulek. Po uložení záznamu se formulář automaticky zavře.



Obr. 5.4: Diagram aktivity funkce Založit šarži

5.1.3 Editace existující šarže

Pokud uživatel vyžaduje upravit existující šarži, musí se ve vizualizaci dostat k popup oknu Detail šarže (kap. 5.1.7), kde se nachází tlačítko Upravit šarži. Po stisknutí tlačítka se uživateli otevře formulář shodný s formulářem pro založení nové šarže, s jediným rozdílem a to, že prvky pro uživatelský vstup mají předvyplněné hodnoty, vyčtené z databáze. Další části procesu jsou shodné s procesem založení nové šarže a jsou znázorněny v diagramu aktivity na obrázku 5.5.

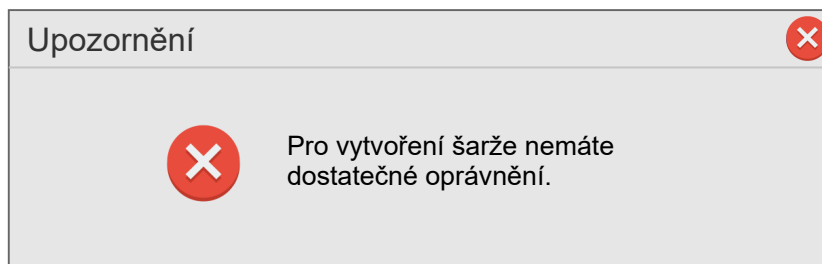


Obr. 5.5: Diagram aktivity funkce Upravit šarži

5.1.4 Kontrola oprávnění

Obě funkcionality Založení šarže a Upravení šarže vyžadují kontrolu oprávnění uživatele k provádění změn. O oprávnění uživatelů rozhodují vedoucí provozů, kteří mohou práva uživatelů editovat přímo ve vizualizaci.

V případě, že uživatel nemá dostatečná oprávnění ke změnám, objeví se systémové hlášení, jak je znázorněno na obrázku 5.6 a neotevře se žádné požadované popup okno.

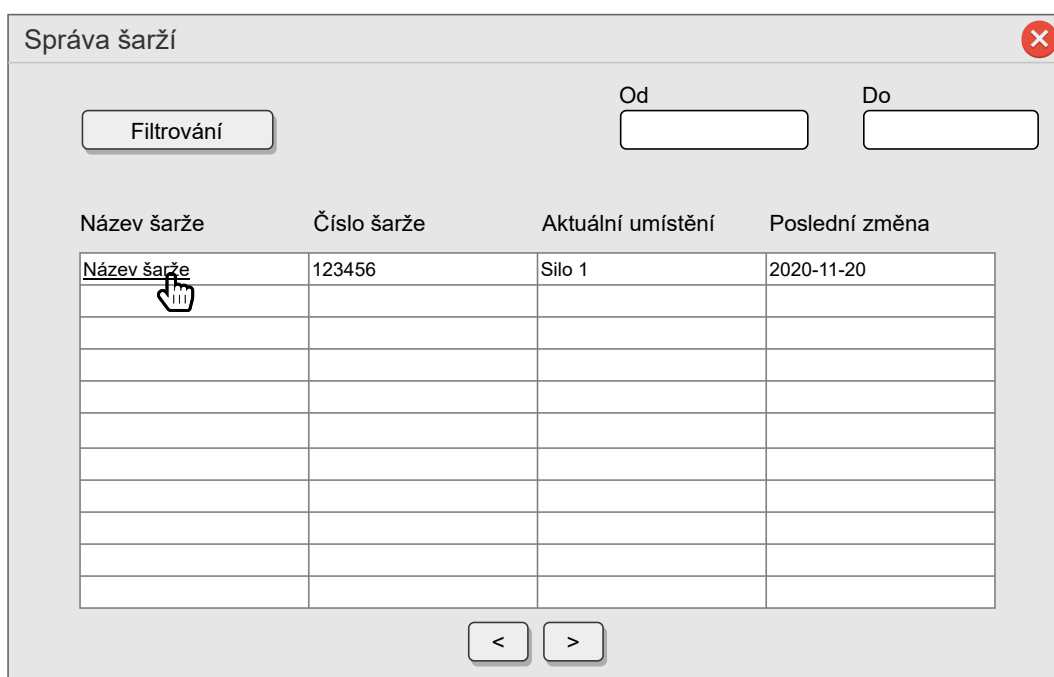


Obr. 5.6: Systémové hlášení o nedostatečném oprávnění uživatele

5.1.5 Zobrazení existujících šarží

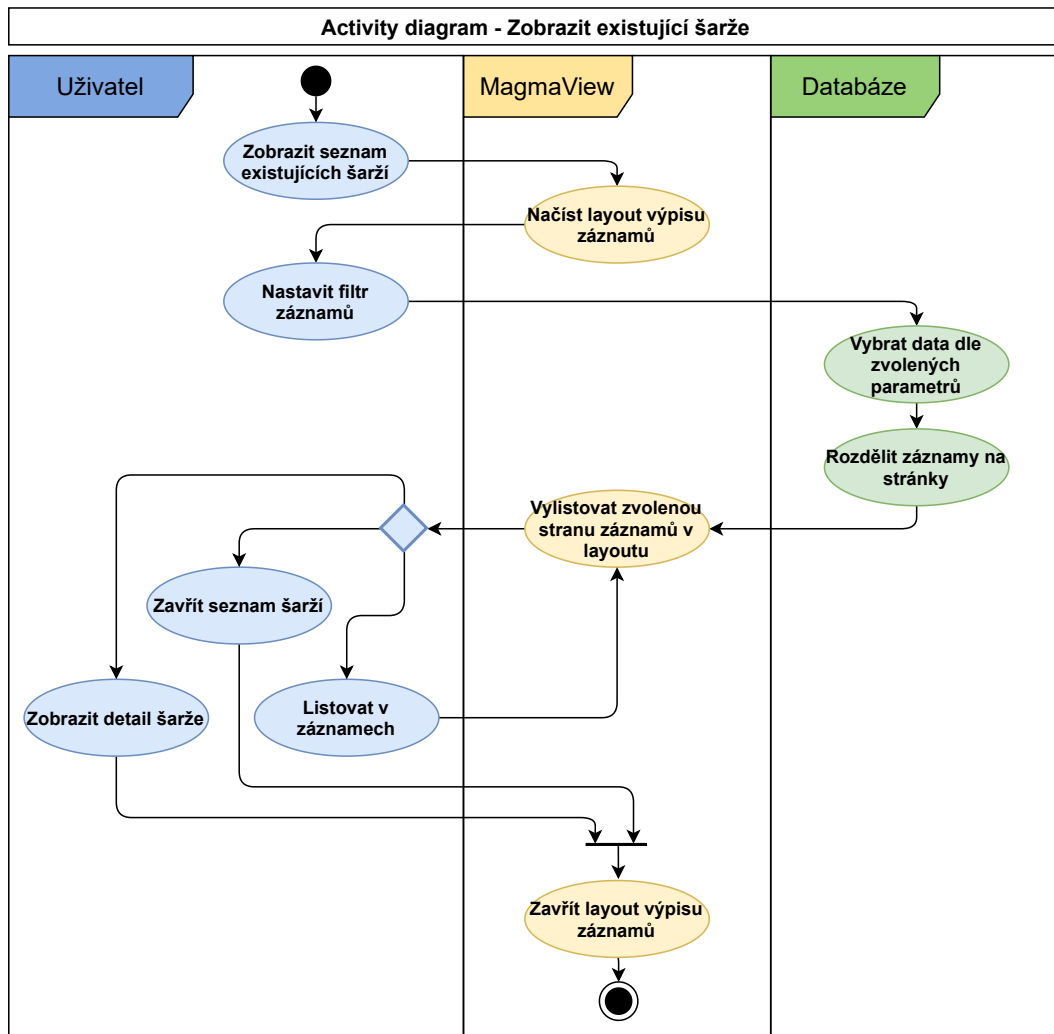
Pro zobrazení seznamu všech existujících šarží slouží obrazovka Správa šarží (obrázek 5.7). Pro každý záznam je zobrazen název šarže, číslo šarže, aktuální umístění ve výrobě a čas poslední změny. Při prvotním otevření okna se zobrazí záznamy seřazené dle času poslední změny šarže. Uživateli je poté umožněno zvolit si časové pásmo, ze kterého chce šarže zobrazit nebo aplikovat jiné filtry, které budou blíže specifikovány na základě požadavků zákazníka.

Na každý název šarže je možné kliknout, čímž se otevře okno s detailem vybrané šarže. V případě většího množství databází vrácených záznamů jsou záznamy rozděleny na stránky a k navigaci se zobrazí tlačítka pro listování.



Obr. 5.7: Pop-up okno pro správu existujících šarží

Na obrázku 5.8 je znázorněn průběh odpovědi na požadavek zobrazení existujících šarží, kromě průběhu prvotního otevření okna, kdy se seznam naplní automaticky. Po vzniku události od uživatele se načte layout pro výpis záznamů. Dále se zpracuje uživatelem nastavené filtrování dat a na základě toho systém sestaví dotazy na databázi. Záznamy vrácené z databáze se rozdělí na optimální počet stran podle maximálního počtu řádků v okně, a do načteného layoutu výpisu se nastaví první strana těchto záznamů. Uživatel má poté tři možnosti a to listovat v záznamech, zobrazit detail vybrané šarže kliknutím na její název nebo zavřít seznam.



Obr. 5.8: Diagram aktivity Zobrazit existující šarže

5.1.6 Historie šarže

Pro podrobnější informace o pohybu šarže přes výrobní proces slouží obrazovka Historie umístění šarže (obrázek 5.9). K této obrazovce se uživatel může dostat z okna detailu konkrétní šarže. Jelikož se může výrobní šarže skládat z několika materiálových šarží, je vždy nejprve uvedena výrobní šarže, reprezentující vyšší úroveň, spolu s umístěním ve výrobě a intervalem odkdy dokdy se v daném umístění šarže nacházela.

Nižší úrovně, prezentované jako Subšarže, představují složky materiálu, ze kterých se výrobní šarže skládá. Jak název šarže, tak subšarže fungují jako odkaz a kliknutí na název otevře okno s detailními informacemi o zvolené šarži.

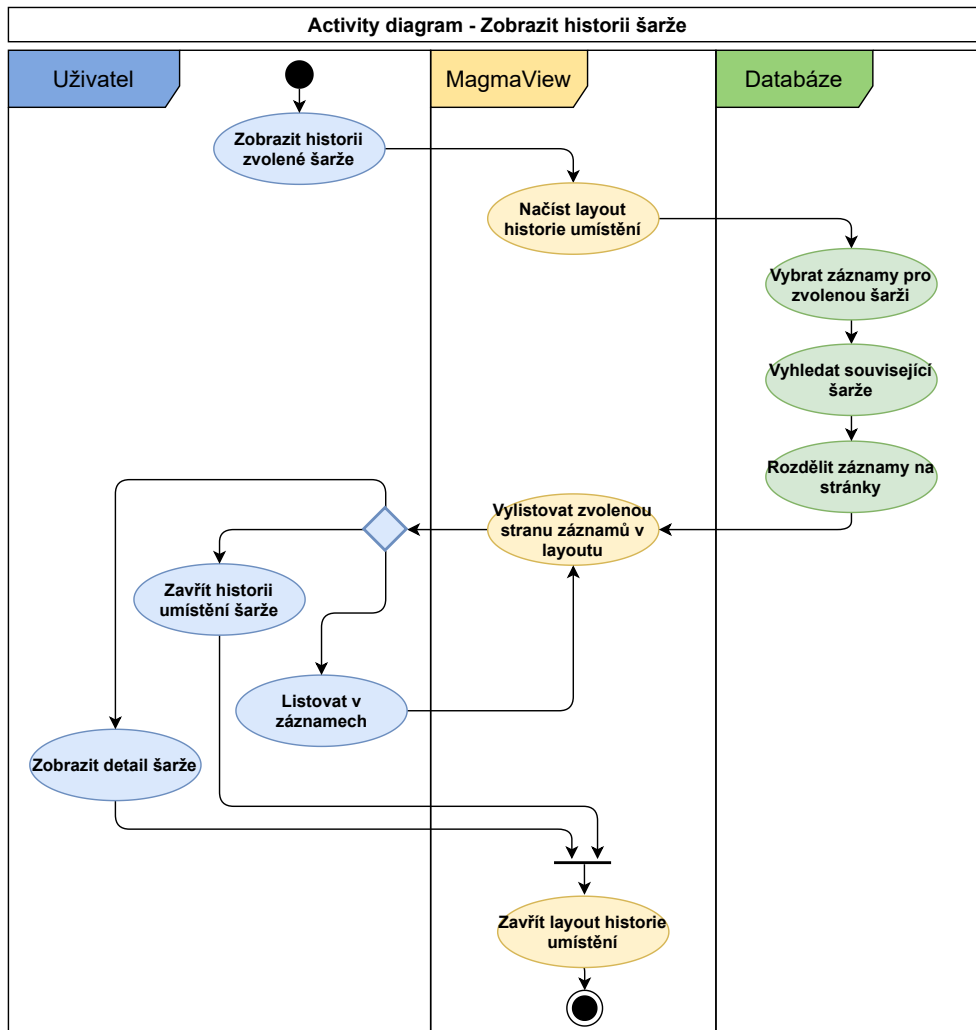
V případě většího množství záznamů lze, stejně jako v seznamu existujících šarží, listovat stranami pomocí tlačítek se šipkami.



Šarže	Subšarže	Umístění	Od	Do
<u>ABC</u>		MIX1	2020-11-20 11:00	2020-11-20 14:30
	<u>123_A</u>			
	<u>123_B</u>			
	<u>123_C</u>			

Obr. 5.9: Popup okno zobrazující historii zvolené šarže

Komunikace se systémem probíhá tak, že po vzniku požadavku na zobrazení historie šarže se nejprve načte grafické rozložení okna. Poté se sestaví dotaz na databázi, který vrátí všechny záznamy týkající se vybrané šarže v dopředném i zpětném směru (vyhledá surový materiál a stejně tak i všechny hotové produkty). Výsledný list záznamů se rozdělí na stránky a (šipkami v okně) volená strana se vypíše v popup okně. Uživatel může listovat v záznamech, zobrazit si detail šarže nebo zavřít okno. Proces zobrazení historie šarže je znázorněn na obrázku 5.10.



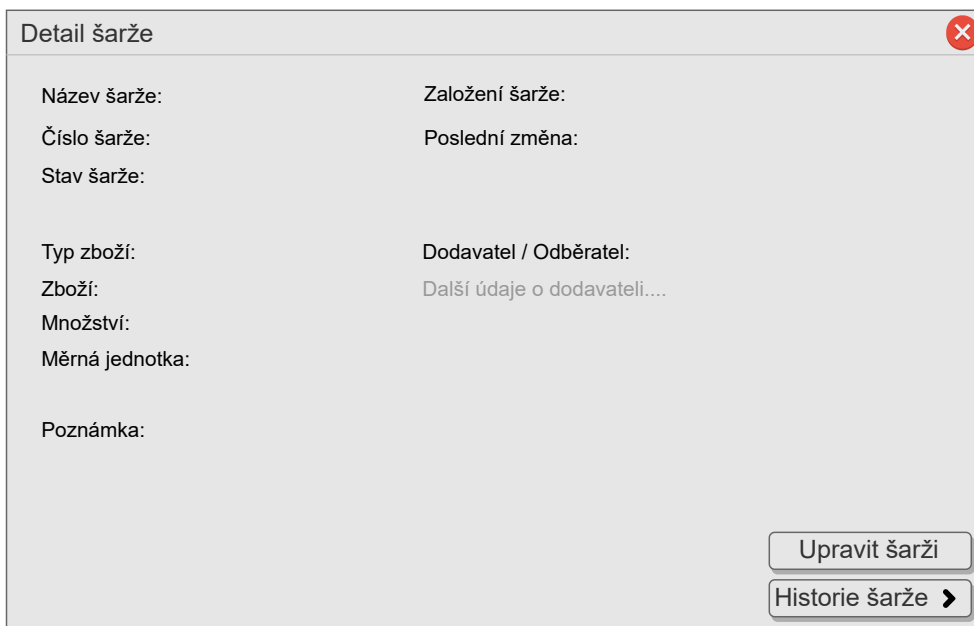
Obr. 5.10: Diagram aktivity Historie šarže

5.1.7 Detail šarže

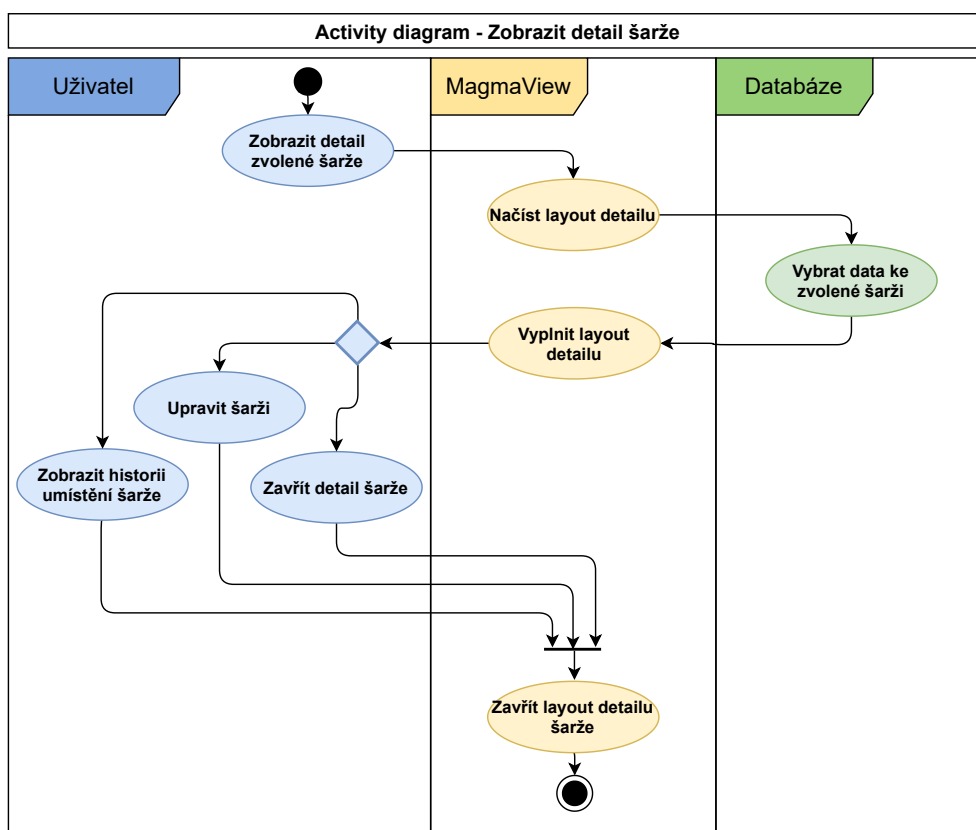
Poslední z funkcionalit pro správu šarží je zobrazení detailu šarže (obrázek 5.11) . Na tuto obrazovku se uživatel může dostat kliknutím na název šarže buď v okně Historie šarže, nebo v okně Správa šarží.

V okně jsou všechny informace týkající se dané šarže, které byly zadány uživatelem při zakládání nové šarže nebo byly systémově aktualizovány. Z této obrazovky může uživatel dále pokračovat na historii šarže nebo informace upravit skrze tlačítko Upravit šarži.

Proces zobrazení detailu šarže, viz. obrázek 5.12, spočívá v reakci systému na požadavek uživatele tím, že načte layout okna, který obsahuje předpřipravené prvky. Poté je sestaven dotaz na databázi, která vrátí dostupné informace o zvolené šarži. Informace se poté namapují na příslušné prvky v okně.



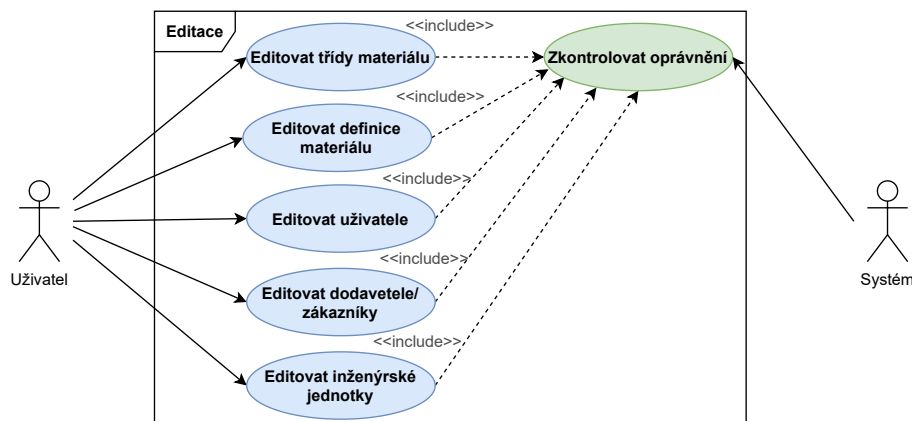
Obr. 5.11: Popup okno s detailními informacemi o dané šarži



Obr. 5.12: Diagram aktivity Detail šarže

5.2 Editace číselníků

Druhý diagram užití, viz. obrázek 5.13, se týká editace uživatelů a editace číselníků, různě používaných ke správě šarží. Pro obecný návrh se počítá s editací tříd materiálu, editací uživatelů, editací údajů o dodavatelích/zákaznících a editací inženýrských jednotek. Každá z akcí vyžaduje kontrolu oprávnění uživatele k provádění změn. V případě nedostatečného oprávnění se zobrazí systémové hlášení, shodné s hlášením na obrázku 5.6.



Obr. 5.13: Diagram užití Editace

5.2.1 Menu editace číselníků

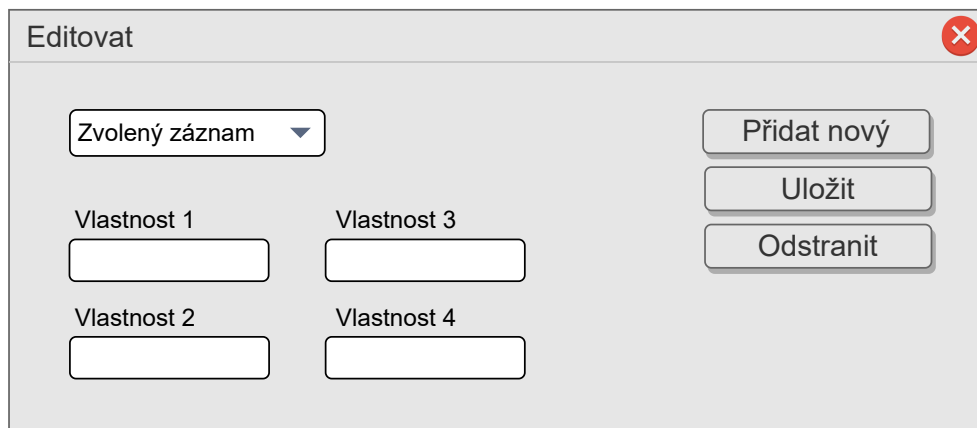
Při stisknutí tlačítka Editace číselníků v menu modulu šarží, které bylo popsán v kapitole 5.1.1, se uživateli objeví nové menu, jak je znázorněno na obrázku 5.14. Zde si může uživatel zvolit, která z entit bude editována. V závislosti na zvolené entitě se uživateli otevře příslušný formulář pro editaci.



Obr. 5.14: Menu pro editaci

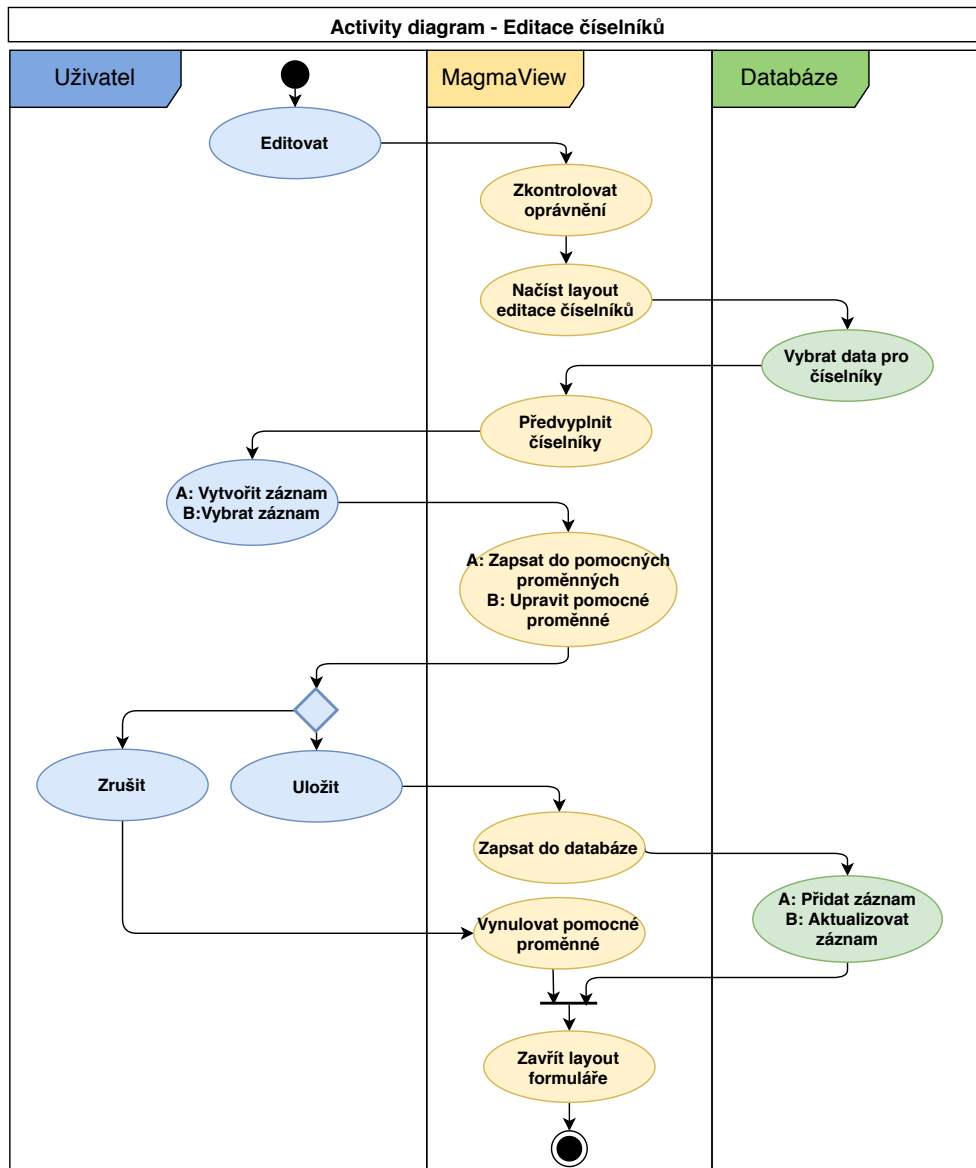
5.2.2 Formulář pro editaci číselníků

Na obrázku 5.15 je pro zjednodušení uveden obecný formulář pro editaci, bez konkrétních prvků. Pro všechny různé entity je shodná volba editovaného záznamu a to prostřednictvím číselníku. Shodné jsou také tlačítka Přidat nový, pro vytvoření dalšího záznamu, Uložit, pro uložení změn záznamu a Odstranit, pro odstranění kompletního záznamu.



Obr. 5.15: Popup okno pro editaci číselníků a uživatelů

Na obrázku 5.16 je znázorněn proces editace. Uživatel vytvoří požadavek na editaci entity. Po kontrole oprávnění systémem se načte formulář pro editaci dané entity a z databáze se načtou všechny dostupné instance entity. Pokud chce uživatel vytvořit nový záznam, přepíše se všechny hodnoty pomocných proměnných uživatelskými vstupy. Pokud chce uživatel upravit existující záznam, přepíše se pouze ty proměnné, u kterých došlo ke změnám. Nový záznam nebo úpravu existujícího může uživatel buď uložit, na což bude systém reagovat příkazem databázi, nebo zrušit, čímž dojde k zavření formuláře a vynulování proměnných.



Obr. 5.16: Diagram aktivity pro editaci číselníků a uživatelů

6 Implementace softwarového modulu

V následující kapitole je popsán postup implementace softwarového modulu na vzorovém příkladě pomyslného technologického procesu. Kapitola rovněž popisuje nutné změny oproti návrhu modulu šarží, které vyplynuly z průběhu implementace.

Implementace modulu pro trasování šarží je rozdělena do několika oblastí, které budou jednotlivě popsány:

- Databázové tabulky
- Vizualizace vzorové technologie
- PLC program obsluhující vzorovou technologii
- Groovy skripty pro editace číselníků a zakládání nových šarží
- Groovy skripty pro trasování šarží
- Groovy skripty pro vyčítání dat o šaržích ve vizualizaci

6.1 SCADA MagmaView

Funkční popis MagmaView je stručně uveden v kapitole 3.1. Z hlediska softwarového sestává SCADA ze tří stěžejních aplikací a několika doplňujících modulů. Hlavními aplikacemi jsou:

- **MagmaView-Core** - Obsahuje hlavní služby a veškerou logiku
- **MagmaView-Editor** - Prostředí pro grafickou editaci vizualizace
- **MagmaView-Runtime** - Spuštění aplikace, volání jednotlivých služeb Core v daném pořadí
- **Ostatní moduly** - License pro správu licence zařízení, DocGenerator pro generování dokumentace k vizualizaci, PassMigrate pro hashování uživatelských hesel

Součástí MagmaView-Editor je možnost definovat pro některé objekty (např. tlačítko) různé akce jako zápis hodnoty, otevření popup oken nebo volání skriptů, čímž se rozšiřují možnosti vizualizace pro zkušenější uživatele. SCADA je optimalizována pro překlad skriptů v jazyce Groovy.

Groovy

Groovy je objektově-orientovaný programovací jazyk pro Java platformu, jenž se překládá ve formě skriptů. Výhodou Groovy skriptů je zjednodušený syntax, včetně oblasti datových typů, které nemusejí být explicitně definovány. V MagmaView je implementována vlastní interpretace Groovy syntaxe pro operace s objekty SCADA.

Vzhledem k rané fázi vývoje modulu pro trasování šarží a vzhledem k proměnlivé povaze technologických procesů nejsou Groovy skripty pro práci se šaržemi implementovány do jádra MagmaView aplikace, ale jsou volány externě z kořenového adresáře. Volání skriptů může probíhat v reakci na akce ve vizualizaci nebo periodicky např. ve skriptu `Main.groovy`, který se volá automaticky vždy při spuštění vizualizace.

6.2 Implementace databáze

Databázové tabulky pro trasování šarží byly vytvořeny jako doplněk databáze vzorového projektu, která má předdefinovanou formu a její design a obsah je ověřován při kompilaci Runtime aplikace i Editoru.

Vytvořená databáze odpovídá návrhu, jenž je rozebrán v kapitole 4.2, s výjimkou tabulky `Person`, jenž byla navržena dle platných norem. Součástí stávající databáze je totiž tabulka `user`, obsahující obdobná data, včetně hashovaných přihlašovacích hesel a proto jí byla tabulka `Person` nahrazena.

Kromě tabulek, vytvořených čistě pro modul šarží, je nutné editovat také tabulku `variables`, která zahrnuje definice adres pro komunikaci s PLC.

Vytvořené tabulky jsou převážně plněny ze skriptů volaných prostřednictvím grafických objektů typu **Formulář**, jak je popsáno dále v kapitole, kromě tabulky `equipment`, která je naplněna pro každý projekt ručně vývojářem a obsahuje seznam procesních jednotek, potřebných k správnému trasování šarží a historizaci dat.

Obsah tabulek po jednom výrobním cyklu a po naplnění číselníků základními údaji je nastíněn v podkapitole 6.11.

6.3 Objekt Formulář

Pro porozumění propojení databáze, skriptů a vizualizace je nutno popsat grafický objekt Formulář, který zahrnují téměř všechna vizualizační okna v návrhu popsaném v kapitole 5.

Objekt Formulář je jedním z grafických prvků MagmaView editoru a slouží k interakci s uživatelem. Formulář může být sestaven z objektů `Button` (tlačítko), `Input` (uživatelský vstup) nebo `Select` (rozbalovací seznam), které na rozdíl od samostatných objektů vizualizace `Button` nebo `Input`, spadají pod entitu formuláře a lze k nim přistupovat ve skriptech dle jejich názvů. Formulář naopak neumožňuje přistupovat k jakýmkoliv jiným prvkům mimo svou entitu.

V rámci vzorového projektu jsou formuláře využívány pro editaci číselníků a správu šarží. Ukázka kódu formuláře v editoru je uvedena ve výpisu 6.1.

V ukázce lze vidět, že instance formuláře se vytváří voláním funkce **@Form** a pojmenováním instance `form_choose_batch`. Následující příkazy určující pozici (`pos`) nebo velikost objektů (`size`) jsou v souladu s jazykem CSS a lze je použít pro všechny objekty editoru. Každý objekt náležející formuláři musí být definován ve funkci `elements`. Součástí objektu formuláře je speciální akce **onLoad**, která se provede vždy při otevření formuláře ve vizualizaci tj. otevření pop-up okna s formulářem nebo přepnutí na obrazovku, na které se formulář nachází.

Výpis 6.1: Ukázka kódu MagmaView editoru - objekt Formulář

```
@Form form_choose_batch
pos: 40 450
size: 380 420
border: "solid"
bordercolor: "gray"
background: "lightgray"

onLoad: runScript("scripts/form/loadExisting.groovy")

elements: {
  @Button selectBatch (
    pos: 10 70,
    size: 150 30,
    text: "Zvolit šarži",
    onClick: runScript("scripts/form/selectBatch.groovy")
  )
}
```

Dalším důležitým objektem formuláře je již zmíněný element `Select`, který funguje jako rozbalovací seznam a jako reakci na výběr ze seznamu má definovanou speciální akci **onSelect**. Ve většině případů formulářů pro modul šarží volá tato akce stejný skript jako akce `onLoad`.

6.4 Objekt Tabulka

V nabídce grafických objektů MagmaView chyběl objekt pro zobrazení většího množství dat z databáze a pro účely šarží bylo nutné tento prvek do aplikace doprogramovat. Po dohodě se zadavatelem měla být tabulka koncipována jako další funkcionality spadající pod objekt `Formulář`.

Vznikl tedy nový objekt `@Table`, který dědí stejné CSS vlastnosti jako jiné prvky. Co se týče skriptů, v těch lze objektu předat dva argumenty, z nichž první obsahuje pole názvů sloupců tabulky a druhý argument je pole polí, obsahující data tabulky.

Výpis 6.2: Ukázka předávání dat tabulce v Groovy skriptu

```
def nazvy = ["sloupec 1", "sloupec 2"]
def radky = [["radek1", "radek1"] ["radek2", "radek2"]]
form.tableName_table_data(nazvy, radky)
```

Předáním dat tabulce se transformací v MagmaView vytvoří objekt datového typu **JSON**. Na obrázku 6.1 lze vidět kód v jazyce Javascript, který vytvoří instanci tabulky a předá jí JSON objekt. V jsonu najde názvy sloupců a poté cyklicky přidává jednotlivé řádky dat do tabulky. Formátování tabulky je překládáno do jazyka HTML a až po spuštění vizualizace tento kód čte webový procesor.

```
1 OS.form.Table.prototype = Object.create(OS.form.DataFormElement.prototype)
2 OS.form.Table.prototype.constructor = OS.form.Table
3 OS.form.Table.prototype.update = function (json) {
4   OS.Component.prototype.update.call(this, json)
5   this.$obj.css(json.css)
6   OS.Component.prototype.postUpdate.call(this, json)
7 }
8
9 OS.form.Table.prototype.loadData = function (data) {
10  this.$obj.children().remove()
11  if ('column_names' in data) {
12    var column_names = data.column_names
13    var rows = data.rows
14    var n = Math.min(column_names.length)
15    var m = Math.min(rows[1].length)
16
17    var s = '<thead><tr class = "form_table_row">'
18    for (var i = 0; i < n; i++) {
19      s = s + '<th>' + column_names[i] + '</th>'
20    }
21    s = s + '</tr></thead>'
22    this.$obj.append(s)
23
24    var body = '<tbody class="form_table_body">'
25
26    for (var k = 0; k < m; k++) {
27      body = body + '<tr class=form_table_row>'
28
29      for (var j = 0; j < n; j++) {
30        body = body + '<td>' + rows[j][k] + '</td>'
31      }
32      body = body + '</tr>'
33    }
34    body = body + '</tbody>'
35    this.$obj.append(body)
36  }
37 }
38
```

Obr. 6.1: Realizace grafického objektu Tabulka v programovacím jazyce Javascript

Tabulky s daty implementované ve vzorovém projektu lze vidět na snímcích vizualizace v příloze B.

6.5 Groovy skripty

Jak již bylo zmíněno dříve, interakce uživatele s databází probíhá prostřednictvím formulářů a Groovy skriptů, což sice představuje více práce při tvorbě projektu, ale na druhou stranu umožňuje vyjít vstříc všem potřebám zákazníka. V rámci implementace modulu trasování šarží bylo vytvořeno téměř 50 skriptů, z nichž velká část obsluhuje formuláře pro editaci číselníků a další skripty slouží k založení nové šarže. Složitější skript pro vyčítání historie šarží je popsán v samostatné kapitole 6.9 a skriptu pro zpracovávání procesních událostí je věnována kapitola 6.8.

6.5.1 Editace číselníků

Při zakládání nové šarže materiálu je nutné danou šarží specifikovat, zvolit dodavatele, množství nebo druh zboží a jiné parametry. Pro operátora je rovněž důležité mít možnost spravovat přístupy různých uživatelů a jejich práv. Všechna tato data může editovat v příslušných formulářích, kdy operátor nejprve vybere z rozbalovacího seznamu daný záznam, který ho zajímá, a výběrem se mu vyplní zbývající objekty formuláře. Všechny tyto funkcionality jsou realizovány vždy stejnou strukturou skriptů odlišující se pouze databázovými dotazy. Struktura je tvořena následujícími skripty:

- **Načti existující záznamy** - Po otevření formuláře se skript automaticky přeloží a načte existující záznamy z dané tabulky do rozbalovacích seznamů typu Select. Funkcionalita tohoto skriptu je obsažena ve všech ostatních skriptech, kde vykonává aktualizaci seznamů po provedení změn.
- **Ulož záznam** - Typicky navázáno na objekt typu Button. Skript provádí aktualizaci hodnot uživatelem upraveného záznamu.
- **Ulož záznam jako nový** - Typicky navázáno na objekt typu Button. Skript zkopíruje hodnoty uživatelem vybraného (a editovaného) záznamu a vytvoří v databázi nový záznam. Formulář se vyplní nově vytvořeným záznamem.
- **Smaž záznam** - Typicky navázáno na objekt typu Button. Skript vymaže uživatelem zvolený záznam z databáze. V případě editace uživatelů se záznam nevymaže, pouze se v databázi označí jako neaktivní, nebude zobrazen ve formuláři v seznamu uživatelů, ale nebude umožněno založení nového účtu s duplicitním uživatelským jménem.

Všechny skripty jsou ošetřeny proti neplatným hodnotám a vyžadují přihlášeného uživatele ve vizualizaci.

6.5.2 Nová šarže

Všechny existující záznamy z tabulek číselníků se načítají do rozbalovacích seznamů formuláře pro založení nové šarže materiálu. Skript pro uložení nové šarže je mírně obsáhlejší, neboť zahrnuje vytvoření více, vzájemně propojených, záznamů.

Při překlada skriptu je nejprve uloženo množství materiálu do tabulky `material_lot_quantity`. Následně je ze stejné tabulky zpětně vyčteno id nového záznamu a uloženo do nového záznamu v tabulce `material_lot`. Jelikož se v případě vzorového projektu považuje skladovací tank materiálové šarže pro dané komponenty současně za výrobní jednotku, je vytvořen také historický záznam v tabulce `job_response` s aktuální časovou značkou.

6.6 Popis procesu

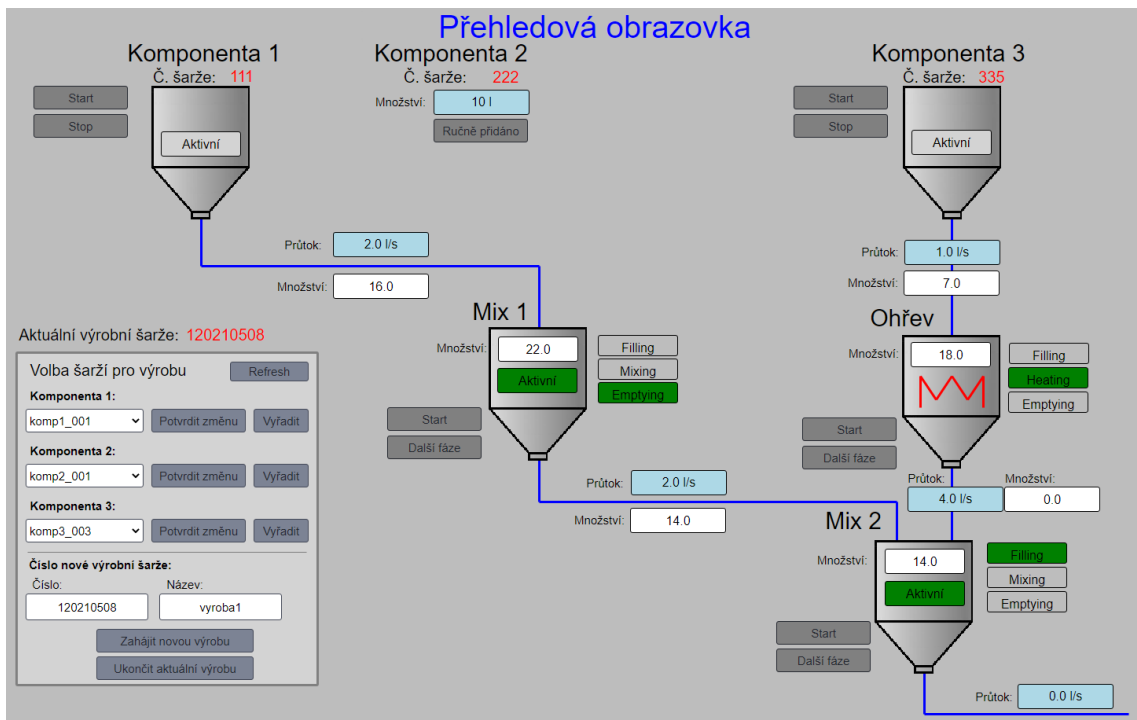
Pro účely vývoje a testování byla vytvořena vzorová SCADA aplikace s jednoduchým technologickým procesem složeným z běžně se vyskytujících technologických bloků. Proces je uvažován jako dávkový a dochází při něm ke zpracování tří druhů materiálu Komponenta 1, 2 a 3. Výroba probíhá ve třech procesních jednotkách Mix 1, Mix 2 a Ohřev, z nich každá operuje ve třech fázích Plnění, Operace s materiálem a Vyprázdnění.

Pro smysluplnost je proces do značné míry realizován tak, jak by tomu bylo u skutečného procesu, tzn. že start procesu je blokován požadavky na stavy procesních jednotek, přechody mezi fázemi jsou také blokovány ukončovacími podmínkami a naprogramována je také simulace toku materiálu včetně možnosti upravovat hodnotu průtoku.

Pro znázornění situace trasování více různě smíchaných šarží se v tanku Mix 1 mísí komponenty 1 a 2 a v další procesní úrovni se v tanku Mix 2 mísí obsah tanků Mix 1 a Ohřev.

Pro každou komponentu může uživatel založit novou materiálovou šarži, kde vyplní veškeré informace o množství, materiálu a dodavateli. Uživatel poté z těchto dostupných šarží vybírá, jakou chce použít pro výrobu. Operátor může šarži změnit také v průběhu výroby, např. v případě doplnění nového materiálu do zásobního tanku. Pokud již některá šarže není k dispozici, operátor ji může ze seznamu vyřadit. Celý jeden výrobní proces má také své vlastní označení šarže, které operátor ručně zadá před zahájením výroby. Označení šarže za ukončenou provádí operátor také ručně z vizualizace.

Přehledová obrazovka vzorové technologie je zachycena na obrázku 6.2.



Obr. 6.2: Přehledová obrazovka vzorového projektu pro trasování šarží

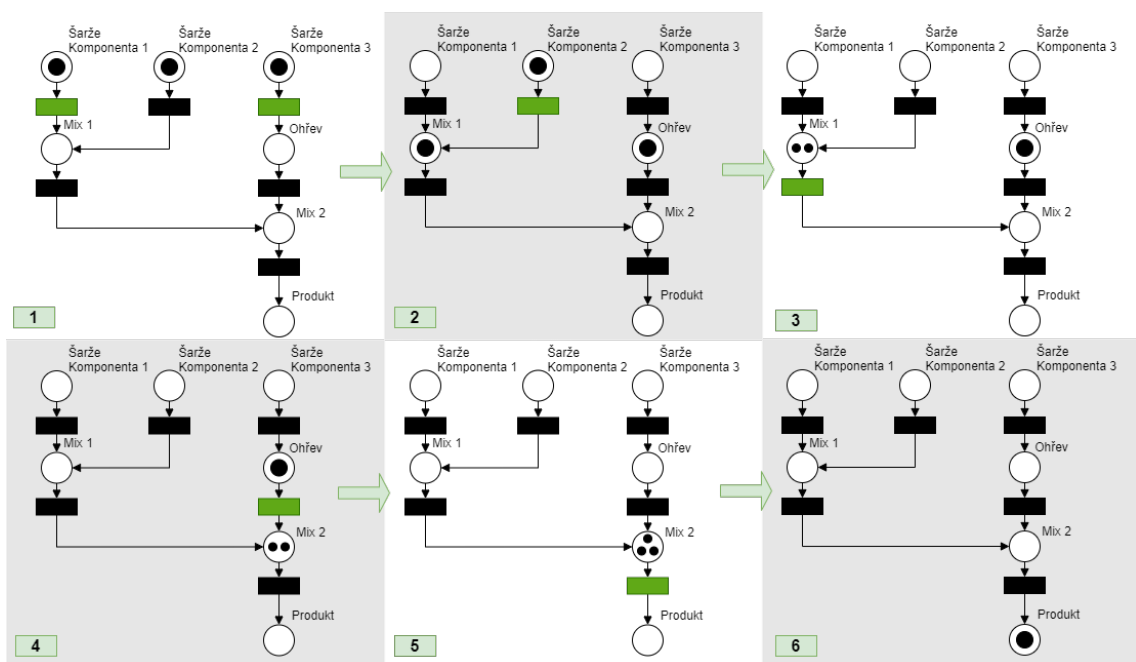
6.7 Řízení procesu a zaznamenávání procesních událostí

Aby bylo možné zpětně trasovat produktovou šarží, je nutné zaznamenat průběh výroby. Výrobní proces ovšem neřídí SCADA aplikace, ale program PLC, který vizualizaci poskytuje data v předdefinované podobě. Řízení výše zmíněného technologického procesu bylo naprogramováno v systému SIMATIC Manager společnosti Siemens. Pro běh procesu byl použit logický kontrolér WinLC, který emuluje funkcionalitu klasického PLC v operačním systému Windows.

Address	Name	Type
0.0		STRUCT
+0.0	Zaznam_pro_SCADA	"UDT_Zaznam"
+40.0	Priprava_Zaznamu	"UDT_Zaznam"
+80.0	Zaznam01	"UDT_Zaznam"
+120.0	Zaznam02	"UDT_Zaznam"
+160.0	Zaznam03	"UDT_Zaznam"
+200.0	Zaznam04	"UDT_Zaznam"
+240.0	Zaznam05	"UDT_Zaznam"

Obr. 6.3: Datový blok pro ukládání záznamů o událostech

Program je rozdělen do tří funkcí. První funkce, **FC_Proces**, obsahuje algoritmus pro řízení technologie a simuluje dávkování materiálu. Průběh technologického procesu je znázorněn pomocí Petriho sítě na obrázku 6.4 Události, které v rámci procesu vznikají, zpracovává funkce **FC_Pridat**, která o nich sestavuje záznam a ukládá je do bufferu událostí. Důvod, proč ukládat události do datového bloku v PLC a ne je rovnou odesílat do MagmaView, je robustnost, neboť u zařízení jako je PLC předpokládáme maximální spolehlivost, kdežto u MagmaView může do běhu aplikace zasáhnout několik faktorů jako operační systém nebo chyba uživatele počítače, na kterém SCADA běží. Při restartování aplikace nebo jiné chybě se tedy neztratí žádný záznam a SCADA naváže ve zpracovávání událostí v místě předchozího přerušení.



Obr. 6.4: Tok materiálu v technologickém procesu vyjádřený pomocí Petriho sítě

Poslední funkce **FC_Udalosti** načítá z emulovaného CPU aktuální čas a kopíruje záznamy z bufferu událostí do záznamu určeném pro SCADA. Součástí funkce je také výpočet CRC za účelem detekce chyby v přenosu dat do MagmaView systému.

Proces vyčtení dat z bufferu je následující. V databázi SCADA aplikace jsou definovány adresy PLC proměnných, kde část proměnných jsou procesní data a část těchto proměnných obsahuje informace o události týkající se šarží. V PLC vzniká procesní událost a data se strádají do přípravného záznamu, který lze vidět na snímku datového bloku 6.3. Hotový připravený záznam se uloží do datového bloku na místo ukazatele (např. Zaznam01) a ukazatel se poté posune na následující pozici.

Po zaplnění bufferu se předchozí záznamy postupně přepisují od začátku bloku. Záznam pro SCADA se nečte přímo z jednotlivých slotů pro události, ale vždy se nakopíruje do slotu Zaznam_pro_SCADA a na tento slot ukazují již zmiňované proměnné definované v databázi.

The screenshot shows a software window titled 'LAD/STL/FBD - [UDT20 -- "UDT_Zaznam" -- Sarze simulace\Sarze\WinLC\...\UDT20]'. Below the title bar is a menu bar (File, Edit, Insert, PLC, Debug, View, Options, Window, Help) and a toolbar. The main area displays a table with the following columns: Address, Name, Type, and Initial value.

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	ID	DINT	L#0
+4.0	Cas	STRUCT	
+0.0	Year	INT	0
+2.0	Month	INT	0
+4.0	Day	INT	0
+6.0	Hour	INT	0
+8.0	Min	INT	0
+10.0	Sec	INT	0
=12.0		END_STRUCT	
+16.0	Operation	INT	0
+18.0	Value	REAL	0.000000e+000
+22.0	SARZE	DINT	L#0

Obr. 6.5: Uživatelsky definovaný datový typ pro záznam o události

Datová struktura záznamu o události, jenž je uvedena na obrázku 6.5, zahrnuje identifikátor záznamu, časovou značku, číslo operace reprezentující druh události, hodnotu a číslo šarže, které se událost týká.

SCADA čte události cyklicky. Po vyčtení záznamu, který je obsahem výše zmiňovaných proměnných, zapíše SCADA do PLC identifikátor zpracovaného záznamu a PLC může při detekci shody identifikátorů zkopírovat další záznam z bufferu. Vizualizací vyčtený záznam je následně zpracován na základě svého obsahu groovy skriptem. Skript pro zpracování záznamu je popsán v následující kapitole.

6.8 Zpracování procesních událostí jako historii šarže

Groovy skript `loadEvents.groovy`, volaný z hlavního skriptu aplikace `Main.groovy`, cyklicky zpracovává příchozí události z PLC a je jádrem trasování šarží. Na začátku skriptu se nejprve porovnají identifikátor záznamu o události, na který ukazují proměnné PLC a identifikátor události, kterou by měla aktuálně zpracovat SCADA. Pokud se tyto identifikátory liší, pak záznam z PLC ještě nebyl zpracován a skript se vykoná, pokud se neliší, pak v PLC nepřibyl žádný nový záznam do bufferu a skript se překládat nemusí.

Dále je spočten kontrolní součet sečtením velikosti všech proměnných, ze kterých se skládá záznam o události viz. obrázek 6.5. Tento kontrolní součet je proveden i v PLC. Pokud se oba součty nerovnají, znamená to, že při komunikaci SCADA-PLC došlo k nějaké chybě a skript se rovněž neprovede. Po splnění těchto dvou podmínek je vykonáno zpracování záznamu o události.

Každá událost musí obsahovat proměnnou **Operation** označující typ události. Tato proměnná datového typu integer se ve skriptu používá jako argument příkazu pro přepínání mezi větvemi kódu a na základě její hodnoty se vykoná obsah větve vyhovující podmínce. V případě vzorového projektu je větví celkem sedm:

Operace = 1

Operace představuje zahájení dávkování materiálu do procesní jednotky. Do databázové tabulky `job_response` je SQL příkazem přidán záznam s časovou značkou označenou jako `START`.

Operace = 2

Operace představuje ukončení dávkování materiálu, kdy již známe konečné množství dávky. Do tabulky `material_quantity` je uložen záznam s tímto množstvím a inženýrskou jednotkou. Dalším dotazem na databázi je nalezen záznam v `job_response` o předchozí operaci jedna a tento záznam je aktualizován o časovou značku označenou jako `STOP`.

V cílové procesní jednotce pro dávkování materiálu vzniká směs. Proto je vytvořen další záznam v tabulce `material_ratio` s odkazem na subšarži, aby bylo možné později dohledat složení směsi.

Operace = 3

Tato operace se týká softwarově nedávkovaných materiálů, jejich množství zadává na vizualizaci operátor ručně. Stejně jako v předešlém případě se vytvoří nový záznam o množství v tabulce `material_quantity`, vznikne záznam o vzniku směsi v

tabulce material_ratio a nový záznam v tabulce job_response, kde mají časové značky START a STOP stejnou hodnotu.

Výpis 6.3: Ukázka kódu skriptu pro zpracování událostí z PLC

```
1 case [4,14,24]: // Start výroby (Filling fáze)
2
3 def sql_param = new ArrayList<Object>([...])
4 def sql_param1 = new ArrayList<Object>([...])
5 def sql_param2 = new ArrayList<Object>([...])
6
7 try{
8     sql.withTransaction{
9         sql.execute """
10             INSERT INTO material_lot (lot_number,
11             lot_name, equipment_id, lot_state, time_insert)
12             VALUES (?, ?, ?, ?, CAST((CONCAT(?.?.)) AS DATETIME))
13             """ , sql_param
14         sql.execute """
15             INSERT INTO job_response (material_lot_id, equipment_id,
16             time_start, description)
17             VALUES ((SELECT material_lot_id FROM material_lot
18             WHERE lot_number = ? AND equipment_id = ?
19             ORDER BY material_lot_id DESC LIMIT 1), ? ,
20             CAST((CONCAT(?.?.)) AS DATETIME), ?);
21             """ , sql_param1
22         sql.execute """
23             UPDATE material_lot
24             SET job_response_id =
25             (SELECT job_response_id FROM job_response
26             WHERE equipment_id = ?
27             ORDER BY job_response_id DESC LIMIT 1)
28             WHERE material_lot_id =
29             (SELECT material_lot_id FROM job_response
30             WHERE equipment_id = ?
31             ORDER BY job_response_id DESC LIMIT 1)
32             """ , sql_param2 }
33 }catch (Exception ex){
34     println ex
35     exit = true
36     break; }
37 DB21_DINT16 = id;
38 break;
```

Operace = 4

Operací 4 začíná zpracovávání údajů o fázích procesních jednotek, konkrétně iniciační fáze, která předchází dávkování materiálu. Nejprve je SQL příkazy vytvořena subšarže v tabulce `material_lot`, ke které se vztahují směsi materiálu, jak bylo popsáno v operaci 2. Pro ukázkou je ve výpisu 6.3 uveden výňatek ze skriptu, vykonávající tuto operaci.

Následně je vytvořen záznam o aktivitě v `job_response` s časovou značkou `START`. Id tohoto záznamu se zpětně uloží do záznamu o subšarži v `material_lot` pro usnadnění orientace mezi tabulkami.

Operace = 5

Pro všechny fáze mezi iniciační a koncovou fází je určena operace pět, jejíž úkonem je najít v `job_response` záznam o předchozí fázi, aktualizovat jeho časovou značku `STOP` a přidat záznam nový. Aktualizován je stejně jako v případě operace čtyři záznam o subšarži v `material_lot`, kde je nahrazeno id předchozího `job_response` záznamu za id nového záznamu.

Operace = 6

Pro koncové fáze procesních jednotek je k dispozici operace šest. Činnosti v této operaci se teoreticky ničím neliší od činností v operaci pět. To by však nemuselo platit pro všechny typy procesů a proto je operace definovaná separátně.

Operace = 7

Poslední možnou operací je větev sedm, která obsluhuje události značící konec jedné výrobní šarže nebo vzájemné míchání subšarží. V případě vzorového projektu se jedná o smíchání subšarží do procesní jednotky v poslední výrobní úrovni a nebo její úplné vyprázdnění.

Pokud se tedy jedná o smíchání subšarží, je konečné množství směsi zaznamenáno do tabulky `material_lot_quantity` a do tabulky `material_ratio` se vytvoří záznam o složce směsi. Posledním příkazem je aktualizace posledního záznamu o aktivitě v tabulce `job_response`.

Obecné vlastnosti

V každé operaci se nachází několik SQL příkazů, které jsou vždy společně vnořeny do funkce `withTransaction` (viz. řádek 8 výpis 6.3). Tato funkce zajišťuje, že se v případě výskytu výjimky vrácené databází ve kterémkoliv příkazu neuloží žádná změna z předchozích úspěšných SQL příkazů. Díky tomu nezůstanou v databázi

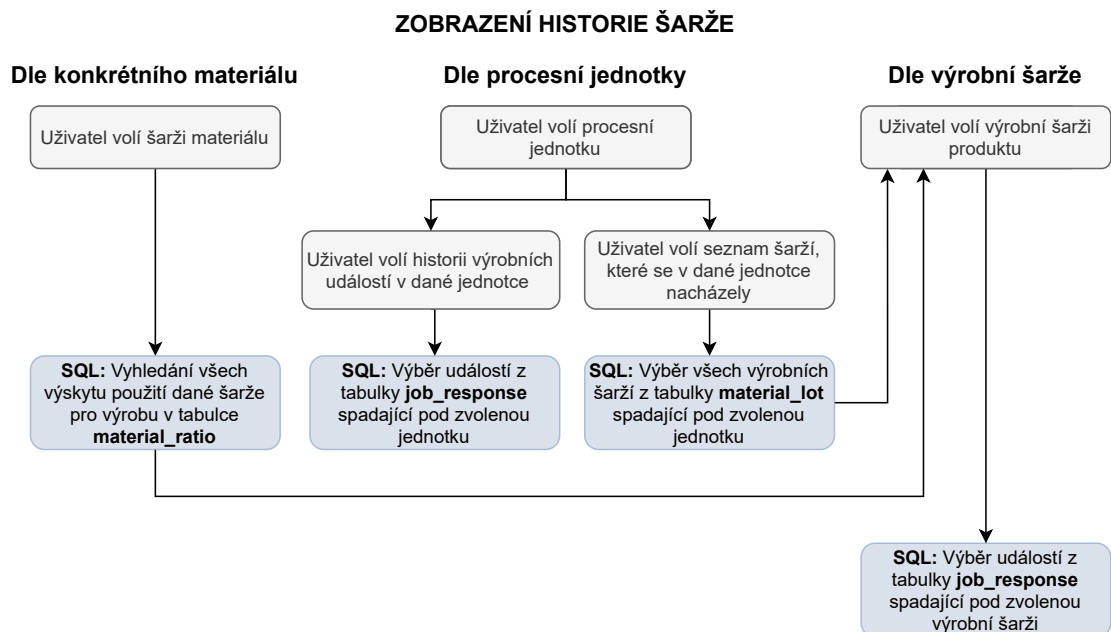
viset neplatná nebo neúplná data a celou operaci je možné provést znovu. Společné pro všechny operace je také zápis identifikátoru události zpět do PLC, v případě, že všechny SQL příkazy proběhly úspěšně a v PLC se tak může ukazatel posunout na další záznam o události. Pokud se vrátí z databáze výjimka, zápis identifikátoru se neprovede a stejná událost bude znovu zpracována při následujícím volání skriptu.

Doporučení pro další vývoj

Jelikož je každý proces jiný a stejně tak se liší požadavky zákazníka, měly by být skripty co nejvíce zobecněny implementací "Query Builderu", funkce, která by dokázala z dodaných dat složit SQL příkazy, aby nemusely být ve skriptu vypsány explicitně.

6.9 Vyčtení historie šarže

Veškeré dosavadně popsané trasování šarže je nyní nutné nějakým způsobem zobrazit uživateli vizualizace. Nejdříve je tedy nutné stanovit, jaké informace by mohl operátor požadovat k zobrazení v případě vzorového projektu. Bylo rozhodnuto, že podstatné by pro operátora měly být tři varianty a to zobrazení historie výrobní šarže, zobrazení historie šarže materiálu a historie procesní jednotky. Diagram těchto variant je uveden na obrázku 6.6.

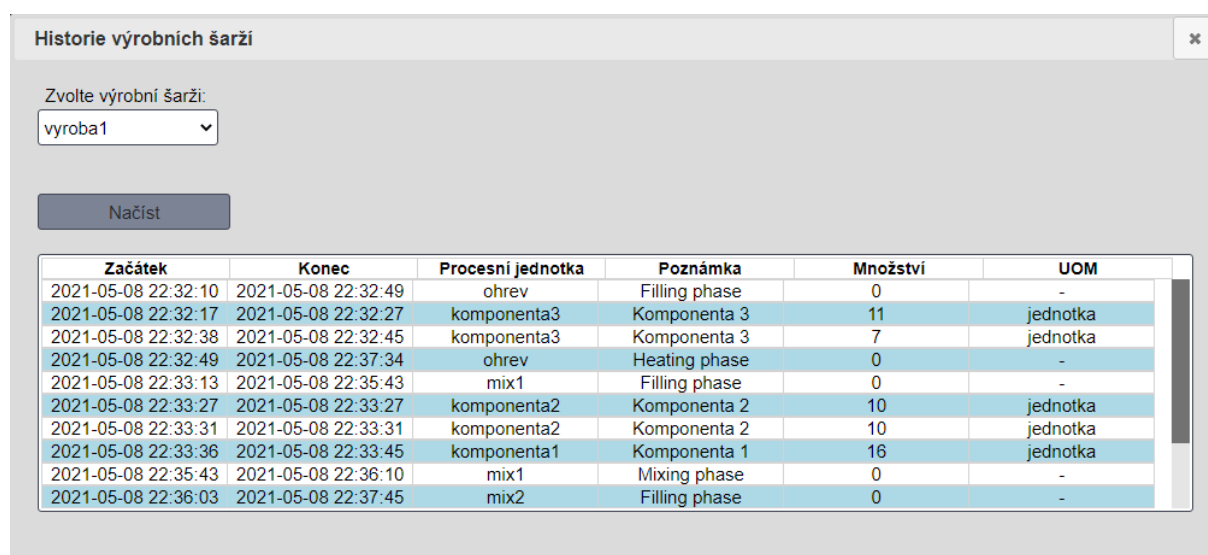


Obr. 6.6: Varianty zobrazení historie šarže na podnět uživatele

Všechny tři varianty jsou opět realizovány Groovy skripty, které se volají dle toho, jaké má uživatel otevřené okno a které tlačítko stiskl. Zavoláním skriptu se načte uživatelova volba z příslušných rozbalovacích seznamů a provedou se SQL příkazy vracející sestavenou databázovou tabulku. Tato tabulka je předána grafickému objektu Tabulka, přičemž názvy sloupců jsou definovány explicitně ve skriptu. Vizualizační okna pro zobrazení historie se liší od původního návrhu, proto zde budou uvedeny snímky jednotlivých variant.

Historie výrobní šarže

Pokud uživatele zajímá průběh celé jedné výrobní šarže, pak si v rozbalovacím seznamu zvolí jednu konkrétní šarži a je mu vrácen seznam historických událostí seřazených podle času. Výsledný seznam je vytvořen spojením různých tabulek, aby byly operátorovi poskytnuty data v co nejsrozumitelnějším tvaru. V tabulce jsou tedy doplněny i názvy procesních jednotek nebo množství použitého materiálu s příslušnou inženýrskou jednotkou. Okno je zobrazeno na snímku 6.7.



The screenshot shows a window titled "Historie výrobních šarží" with a dropdown menu set to "vyroba1" and a "Načíst" button. Below is a table with the following data:

Začátek	Konec	Procesní jednotka	Poznámka	Množství	UOM
2021-05-08 22:32:10	2021-05-08 22:32:49	ohrev	Filling phase	0	-
2021-05-08 22:32:17	2021-05-08 22:32:27	komponenta3	Komponenta 3	11	jednotka
2021-05-08 22:32:38	2021-05-08 22:32:45	komponenta3	Komponenta 3	7	jednotka
2021-05-08 22:32:49	2021-05-08 22:37:34	ohrev	Heating phase	0	-
2021-05-08 22:33:13	2021-05-08 22:35:43	mix1	Filling phase	0	-
2021-05-08 22:33:27	2021-05-08 22:33:27	komponenta2	Komponenta 2	10	jednotka
2021-05-08 22:33:31	2021-05-08 22:33:31	komponenta2	Komponenta 2	10	jednotka
2021-05-08 22:33:36	2021-05-08 22:33:45	komponenta1	Komponenta 1	16	jednotka
2021-05-08 22:35:43	2021-05-08 22:36:10	mix1	Mixing phase	0	-
2021-05-08 22:36:03	2021-05-08 22:37:45	mix2	Filling phase	0	-

Obr. 6.7: Zobrazení historie výrobní šarže

Historie šarže materiálu

Zobrazení historie (viz. obrázek 6.8) materiálové šarže probíhá nejprve výběrem definice materiálu a následně výběrem konkrétní šarže z rozbalovacího seznamu. Po výběru šarže se na pravé straně popup okna zobrazí detaily zadané uživatelem při zakládání nové šarže materiálu. Stisknutím tlačítka Načti se uživateli zobrazí seznam, který obsahuje identifikátory výrobních šarží z tabulky `material_lot`, které se zvolenou materiálovou šarží pracovaly, a k tomu je také přiřazen identifikátor

aktivity z tabulky `job_response`, který poskytuje časový údaj, kdy byl materiál použit.

V okně se nachází také tlačítko **Detail**, po jehož stisknutí se otevře nové okno, podobné oknu pro zakládání nové šarže, a uživatel může editovat informace o šarži.

Obr. 6.8: Zobrazení historie materiálových šarží

Historie procesní jednotky

V případě, že dojde k pozdě odhalené poruše v procesní jednotce (např. nedosažení žádané hodnoty při pasterizaci), mohlo by operátora zajímat, jaké výrobní šarže touto jednotkou prošly. Uživatel tedy na SCADA zvolí danou procesní jednotku a stisknutím tlačítka **Výrobní šarže** je mu v tabulce vrácen seznam šarží, které se v dané jednotce nacházely.

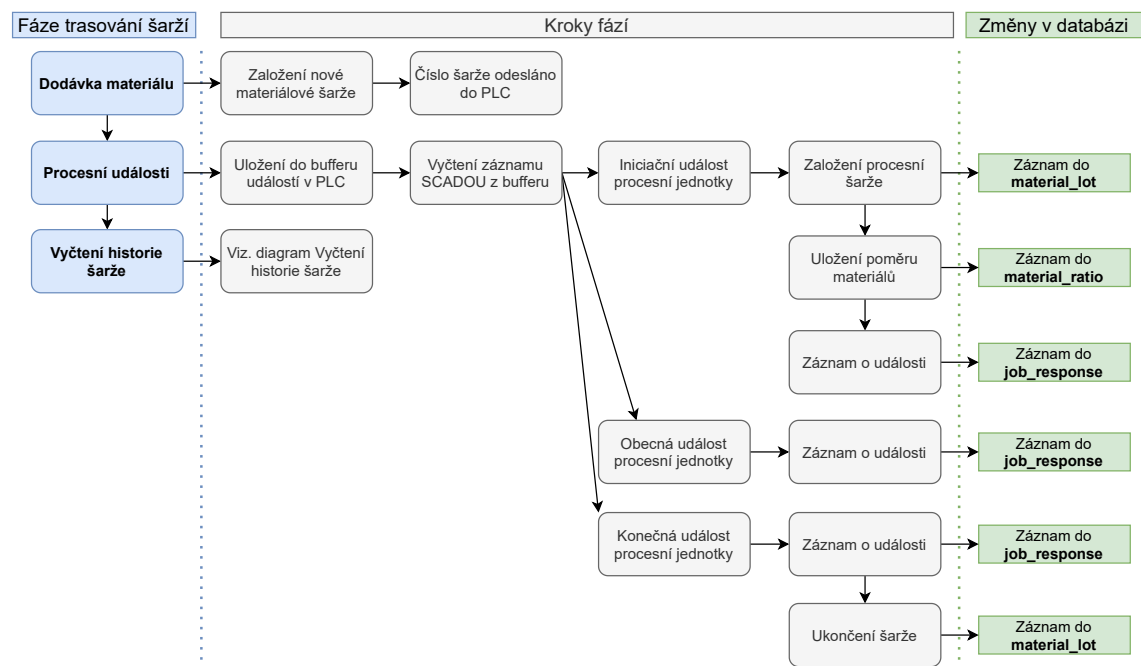
Obr. 6.9: Zobrazení historie výrobní šarže

Obdobně si může uživatel v tabulce zobrazit materiálové šarže, pokud jej zajímá, jaký materiál se v jednotce nacházel.

Druhou variantou je zobrazení historie výrobních událostí v procesní jednotce. Uživatel opět vybere procesní jednotku, ale tentokrát stiskne tlačítko **Výrobní události** a je mu vrácen seznam historických událostí seřazený podle času, nehledě na výrobní šarži. Pro snazší orientaci se při zpracovávání událostí automaticky do poznámky záznamu zapisuje popis události, který lze na obrázku 6.9 vidět ve sloupci Detail.

6.10 Shrnutí procesu trasování šarží

Pro úplné pochopení souvislostí všech částí implementace je opětovně popsán celý proces trasování šarží přehledně krok za krokem z perspektivy uživatele vizualizace a následných odezev na jeho podněty. Vysvětlení je doplněno o diagram na obrázku 6.10, na kterém je také znázorněno, jaké databázové tabulky jsou trasováním ovlivněny.



Obr. 6.10: Diagram zpracovávání dat o šaržích

Před zahájením výroby je nutná příprava materiálu. Dodávku materiálu uživatel zpracuje formou vytvoření nové materiálové šarže v grafickém formuláři. Parametry šarže vybírá z předvyplněných číselníků, které může libovolně editovat.

Zahájení výroby probíhá v dalším formuláři výběrem šarží materiálů, ze kterých se bude vyrábět, a založením nové výrobní šarže. Čísla šarží jsou odeslána do PLC,

neboť později budou součástí záznamu o procesních událostech.

Následuje spuštění výrobního procesu. Události, podstatné pro trasování šarží, se v PLC ukládají do bufferu a SCADA tyto události vyčítá. Na základě typu události se událost zpracuje a vytvoří se nebo aktualizují záznamy v databázových tabulkách. Uložené a historizované údaje o šaržích si operátor může zobrazit v několika variantách a nastavit si filtrování zobrazených dat. Z listu vyfiltrovaného seznamu šarží nebo historických událostí se může uživatel prokliknout na detail libovolné šarže.

6.11 Vyhodnocení běhu výrobního procesu

Jeden průběh výrobního procesu vzorového projektu poskytuje pouze minimum záznamů, vzhledem k jeho rozsahu. Cílem ovšem není testovat optimalizaci řešení, ale korektní provázanost databázových tabulek a schopnost data zpětně zobrazit ve smysluplném formátu. Pro ukázkou byly spuštěny dva výrobní procesy, s různou kombinací materiálových šarží, jak lze vidět v tabulce 6.1.

		Materiálové šarže					
		Komponenta 1		Komponenta 2	Komponenta 3		
Název šarže		Komp1_001	Komp1_002	Komp2_001	Komp3_001	Komp3_002	Komp3_003
Počet dávek	Výroba 1	1	0	2	1	0	1
	Výroba 2	0	1	1	0	0	1

Tab. 6.1: Kombinace šarží jednotlivých komponent testovacích průběhů výroby

V prvním průběhu výroby byly kombinovány dvě šarže komponenty 3, což simuluje situaci, kdy je nutné v průběhu výroby doplnit chybějící materiál. Z komponenty 2 pak byly pro změnu použity dvě dávky stejné šarže, simulující situaci, kdy potřebujeme znát, kolikrát jsme jednu materiálovou šarží použili a např. lépe počítat bilance.

Výrobní šarže	Podšarže	Počet záznamů v databázových tabulkách			
		material_lot	material_ratio	material_lot_quantity	job_response
Výroba 1		1	0	0	0
	Mix 1	1	3	3	6
	Heater	1	2	2	5
	Mix2	1	2	2	3
Výroba 2		1	0	0	0
	Mix1	1	2	2	5
	Heater	1	1	1	4
	Mix2	1	2	2	3

Tab. 6.2: Počet záznamů v jednotlivých tabulkách pro dva výrobní procesy

V tabulce 6.2 lze vidět, jakým způsobem se projeví běh procesů na počtu záznamů v databázových tabulkách.

K optimalizaci řešení lze dodat, že počet záznamů zobrazených v tabulkách by měl být vždy regulovaný editorem skriptů, aby nedocházelo k přílišnému vytížení systému. Počet záznamů v databázi je omezen velikostí volného datového prostoru počítače, na kterém SCADA běží nebo může být omezen nastavením databáze, včetně automatického promazávání nepotřebných záznamů.

7 Vyhodnocení souladu se standardem

Pro vyhodnocení míry souladu se standardem je nutné specifikovat, co je touto frází míněno. Jak již bylo popsáno v teoretické kapitole 1, věnované standardu ISA-95, cílem je sjednotit různorodá řešení a umožnit snazší komunikaci podnikových systémů v různých úrovních. V praxi to mimo jiné znamená přenos dat mezi aplikacemi v podobě unifikovaných datových souborů v jazyce XML. Standardní aplikace by měla být schopna tyto soubory vytvářet i číst. V rámci jedné aplikace standard také vymezuje činnosti, které zmiňovaná data produkují a jaký mají význam pro chod podniku.

V případě modulu pro trasování šarží se jedná o implementaci do aplikace, která obecně není navržena podle platných standardů. Při implementaci tedy bylo nutné dbát na to, aby bylo vše v souladu především se zbývajícími částmi aplikace.

Součástí nového modulu bylo značné množství databázových tabulek, které nijak závažně nezasahují do zbývajících částí databáze MagmaView systému a byl zde prostor pro aplikaci některých oblastí standardu. Konkrétně byly tabulky navrženy dle několika B2MML schémat, ze kterých byly čerpány struktura tabulky, názvy jednotlivých parametrů a vztahy mezi dalšími tabulkami. Ovšem i zde byl nutný kompromis, neboť pokud by byly vytvářeny tabulky bezprostředně podle schémat, bylo by součástí databáze nadbytečné množství tabulek a parametrů, které by vzhledem k rozsahu projektů realizovaných v MagmaView zůstaly nevyužity. Detaily k čerpání ze schémat jsou součástí popisu návrhu datového modelu v kapitole 4.

V případě, že by později byl do aplikace implementován modul pro vytváření a čtení XML souborů, bylo by možné data týkající se šarží s drobnými úpravami přenášet mezi jinými aplikacemi. Zde je ovšem nutno vzít v úvahu, že dosavadně je málokterý výrobní proces řízen standardně a nižší podnikové systémy vždy vyžadují úpravy, aby byly s výrobou v souladu. Pokud by bylo zvýšení míry standardizace cílem dalšího vývoje, doporučuje se začít přepracováním zbývajících částí databáze dle B2MML schémat.

Z hlediska samotné funkcionality modulu trasování šarží je považován modul v souladu se standardem, přestože by měl být dle norem součástí systémů na úrovni MES a zde se jedná o implementaci na úrovni SCADA. Modul je schopen trasování kompletní historie produktu, včetně použitého materiálu, obsluhy nebo stavů a podmínek produkce.

Závěr

Diplomová práce se zabývá trasováním šarží ve výrobě ve vizualizačním systému MagmaView. Cílem práce bylo navrhnout a implementovat softwarový modul, který by spravoval data o šaržích materiálu a produktu ve výrobě a utvářel historii procesního umístění. Vývoj modulu měl podléhat návrhu, který se měl dle požadavků skládat z návrhu databázové struktury a GUI modulu, včetně popisu interakce s uživatelem. Vytvořený návrh měl být realizován, s ohledem na standardy platné v procesním průmyslu, a otestován z hlediska funkčnosti.

V první části byly popsány teoretické aspekty práce, které zahrnovaly rozbor mezinárodního standardu ISA-95 a v něm obsažených modelů, B2MML implementaci onoho standardu. Teoreticky popsána byla také problematika traceability, včetně uvedení příkladů řešení dostupných na trhu. Praktická část práce se skládá ze čtyř kapitol, které pokrývají plánování vývoje, návrh modulu a samotnou implementaci. V kapitole tři byly shrnuty základní požadavky na trasování šarží, ze kterých vyplynulo, že bude nutné k šaržím přistupovat dle jejich povahy a pomyslně je dělit na dodavatelskou, expediční a výrobní šarži. Od tohoto dělení se následně odvíjelo provazování informací o průběhu výroby nebo materiálu.

V následující kapitole čtyři byl navržen datový modul, koncipovaný jako relační databáze sestavená na základě standardních B2MML schémat a předpokladu o tom, jaké informace by obecně mohl požadovat zákazník. Databázový model byl složen z několika tabulek, vzájemně propojených tak, aby bylo možné dohledat spojení mezi materiálem a produktem z jakéhokoli bodu ve výrobním procesu. Dále bylo do databázového modelu zahrnuto zaznamenání historie umístění šarže, tedy na jakých pozicích se šarže při výrobě nacházela a po jakou dobu. Podstatnými ukládanými daty jsou také informace popisující vlastnosti materiálu, množství materiálu, dodavatele a odběratele.

Kapitola pátá doplnila návrh o GUI model, skládající se z několika grafických rozvržení a oken, skrze které by uživatel mohl ovládat modul šarží ve vizualizaci. Ke složitějším oknům byly doplněny diagramy aktivity, specifikující proces interakce uživatele s modulem. GUI modul byl navržen s ohledem na uživatelskou zkušenost a s ohledem na datový model, díky kterému bylo možné specifikovat, jaké informace se očekávají od uživatele a jaké poskytuje systém automaticky.

Kompletní návrhy byly implementovány do MagmaView systému, o čemž pojednávala kapitola šest. Nejprve byly představeny základní nástroje, využívané při realizaci modulu, kterými jsou převážně grafické prostředí MagmaView a Groovy skripty. Do grafického prostředí byl implementován také nový grafický objekt typu tabulka pro zobrazení ukládaných dat. Dále byl vytvořen vzorový projekt, včetně řídicího programu jednoduchého technologického procesu, za účelem přiblížení realizace

skutečnému případu použití v praxi.

Implementace pokračovala různými řešeními pro vytváření dat, předávání dat mezi PLC a systémem SCADA, ukládání těchto informací a následné zobrazení operátorovi prostřednictvím vizualizace.

Posledním cílem práce bylo zhodnocení souladu se standardem, ze kterého vyplynulo, že funkcionality trasování materiálu odpovídá předpokladům standardu. Co se týče softwarového řešení, databáze modulu byla navržena dle standardních schémat a ačkoliv modul není aktuálně schopen předávání dat mezi jinými, standardními systémy, je na to připraven pro budoucí vývoj.

Literatura

- [1] *Enterprise-Control System Integration Part 1: Models and Terminology: ISA-95.00.01-2000*. Research Triangle Park, North Carolina, USA: Instrument Society of America, 2000. ISBN 1-55617-727-5.
- [2] *PÁSEK, Jan a Vlastimil BRAUN. Automatizace procesů II: Úroveň řízení výroby*. Brno, 2015.
- [3] *Business To Manufacturing Markup Language*. 6. Arizona, USA: MESA, 2013.
- [4] *BRAUN, Vlastimil. Systémy sledování výroby a traceability jsou ochranou spotřebitelů i výrobců*. MM Průmyslové spektrum. 2005, , 106. Dostupné také z: <<https://www.mmspektrum.com/systemy-sledovani-vyroby-a-traceability>>
- [5] *KANISOVÁ, Hana a Miroslav MILLER. UML srozumitelně*. 1. Brno: Computer Press, 2004. ISBN 80-251-0231-9.
- [6] *Co je to Cizí klíč?* IT SLOVNÍK.cz [online]. [cit. 2020-12-28]. Dostupné z: <https://it-slovník.cz/pojem/cizi-klic>
- [7] *CHENG, M.J. a J.E.L. SIMMONS. Traceability in Manufacturing Systems*, International Journal of Operations & Production Management. 14. 1994, s. 4-16.
- [8] *M. R. Khabbazi, N. Ismail, M. Y. Ismail and S. A. Mousavi, Data Modeling of Traceability Information for Manufacturing Control System*, 2009 International Conference on Information Management and Engineering, Kuala Lumpur, 2009, pp. 633-637, doi: 10.1109/ICIME.2009.58.
- [9] *Traceability, zpětná dosledovatelnost výrobků*. Act-in.cz [online]. [cit. 2021-01-03]. Dostupné z: <http://www.act-in.cz/traceability>
- [10] *Traceability a sledování výroby potravin*. COMES.EU [online]. [cit. 2021-01-03]. Dostupné z: <https://www.comes.eu/pripadove-studie/250-traceability-a-sledovani-vyroby-potravin>
- [11] *What is Traceability in Manufacturing and How to Implement It?* Katanamrp.com [online]. [cit. 2021-01-03]. Dostupné z: <https://katanamrp.com/blog/product-traceability/>
- [12] *Lot Tracking Software for Product Traceability*. Katanamrp.com [online]. [cit. 2021-01-03]. Dostupné z: <https://katanamrp.com/lot-tracking-software/>

- [13] *MAGMAVIEW Naše vizualizace vašich procesů*. PBControl.cz [online]. [cit. 2020-01-03]. Dostupné z: <https://pbcontrol.cz/magmaview/>

Seznam symbolů a zkratek

ANSI	American National Standards Institute
B2MML	Business To Manufacturing Markup Language
CSS	Cascading Style Sheets
GUI	Graphical User Interface
HTML	Hypertext Markup Language
IIS	Internet Information Services
ISA	International Society of Automation
JSON	JavaScript Object Notation
MES	Manufacturing Execution System
NFC	Near field communication
OPC UA	Open Platform Communications – Unified Architecture
PLC	Programmable Logic Controller
QR	Quick Response
RFID	Radio-Frequency Identification
S7	Komunikační protokol Siemens
SCADA	Supervisory Control And Data Acquisition
SQL	Structured Query Language
UML	Unified Modeling Language
VPN	Virtual Private Network
XML	Extensible Markup Language
XSD	XML Schema Definition

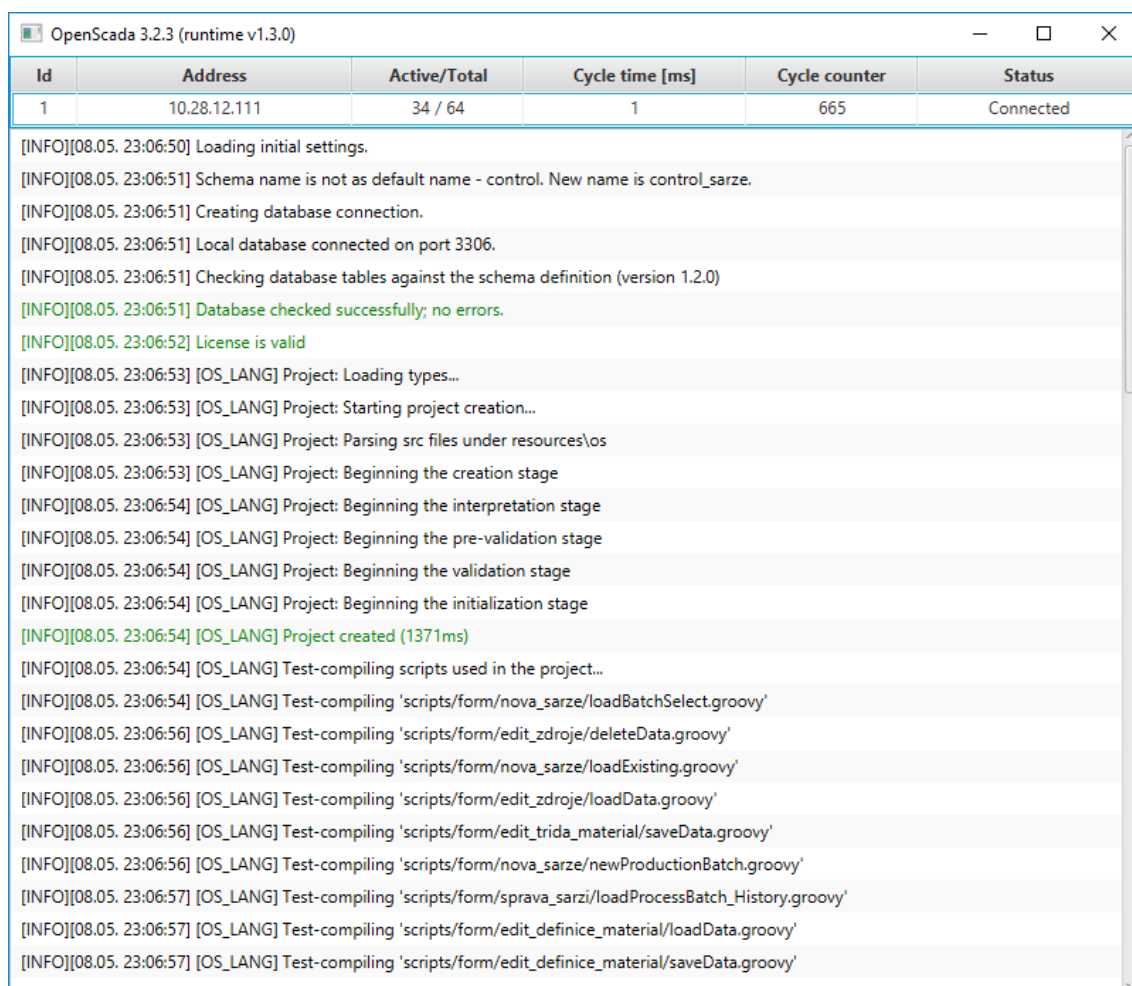
Seznam příloh

A	Datový model	83
B	SCADA Vzorového projektu	85
C	Obsah přiloženého CD	93

A Datový model

B SCADA Vizorového projektu

V příloze jsou pro názornost uvedeny snímky z vizualizace vizorového projektu, včetně vyplněných dat a tabulek, zobrazující data z databáze o proběhlé výrobě.



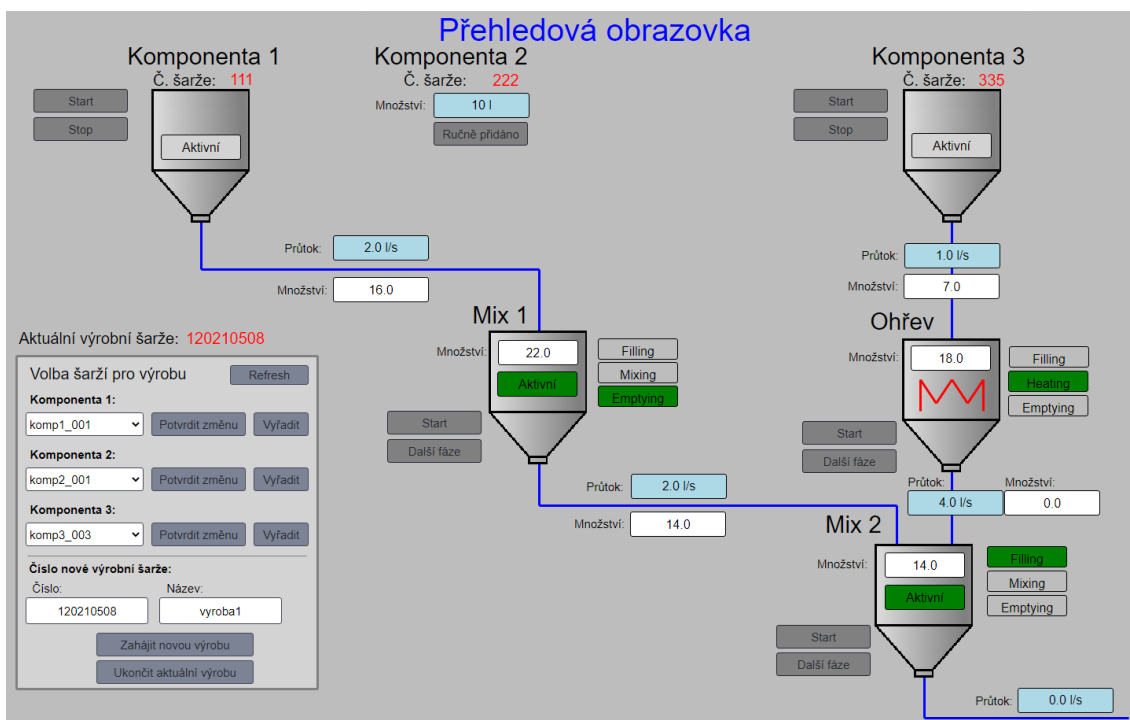
The screenshot displays the OpenScada 3.2.3 runtime interface. At the top, a window title bar reads "OpenScada 3.2.3 (runtime v1.3.0)". Below it is a table with the following data:

Id	Address	Active/Total	Cycle time [ms]	Cycle counter	Status
1	10.28.12.111	34 / 64	1	665	Connected

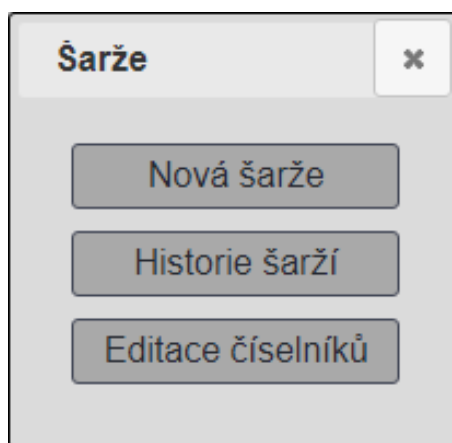
Below the table is a log window showing the following messages:

```
[INFO][08.05. 23:06:50] Loading initial settings.
[INFO][08.05. 23:06:51] Schema name is not as default name - control. New name is control_sarze.
[INFO][08.05. 23:06:51] Creating database connection.
[INFO][08.05. 23:06:51] Local database connected on port 3306.
[INFO][08.05. 23:06:51] Checking database tables against the schema definition (version 1.2.0)
[INFO][08.05. 23:06:51] Database checked successfully; no errors.
[INFO][08.05. 23:06:52] License is valid
[INFO][08.05. 23:06:53] [OS_LANG] Project: Loading types...
[INFO][08.05. 23:06:53] [OS_LANG] Project: Starting project creation...
[INFO][08.05. 23:06:53] [OS_LANG] Project: Parsing src files under resources\os
[INFO][08.05. 23:06:53] [OS_LANG] Project: Beginning the creation stage
[INFO][08.05. 23:06:54] [OS_LANG] Project: Beginning the interpretation stage
[INFO][08.05. 23:06:54] [OS_LANG] Project: Beginning the pre-validation stage
[INFO][08.05. 23:06:54] [OS_LANG] Project: Beginning the validation stage
[INFO][08.05. 23:06:54] [OS_LANG] Project: Beginning the initialization stage
[INFO][08.05. 23:06:54] [OS_LANG] Project created (1371ms)
[INFO][08.05. 23:06:54] [OS_LANG] Test-compiling scripts used in the project...
[INFO][08.05. 23:06:54] [OS_LANG] Test-compiling 'scripts/form/nova_sarze/loadBatchSelect.groovy'
[INFO][08.05. 23:06:56] [OS_LANG] Test-compiling 'scripts/form/edit_zdroje/deleteData.groovy'
[INFO][08.05. 23:06:56] [OS_LANG] Test-compiling 'scripts/form/nova_sarze/loadExisting.groovy'
[INFO][08.05. 23:06:56] [OS_LANG] Test-compiling 'scripts/form/edit_zdroje/loadData.groovy'
[INFO][08.05. 23:06:56] [OS_LANG] Test-compiling 'scripts/form/edit_trida_material/saveData.groovy'
[INFO][08.05. 23:06:56] [OS_LANG] Test-compiling 'scripts/form/nova_sarze/newProductionBatch.groovy'
[INFO][08.05. 23:06:57] [OS_LANG] Test-compiling 'scripts/form/sprava_sarzi/loadProcessBatch_History.groovy'
[INFO][08.05. 23:06:57] [OS_LANG] Test-compiling 'scripts/form/edit_definice_material/loadData.groovy'
[INFO][08.05. 23:06:57] [OS_LANG] Test-compiling 'scripts/form/edit_definice_material/saveData.groovy'
```

Obr. B.1: Ukázka Runtime aplikace MagmaView



Obr. B.2: Přehledová obrazovka vzorového projektu pro trasování šarží



Obr. B.3: Základní menu modulu šarží

Nová šarže [X]

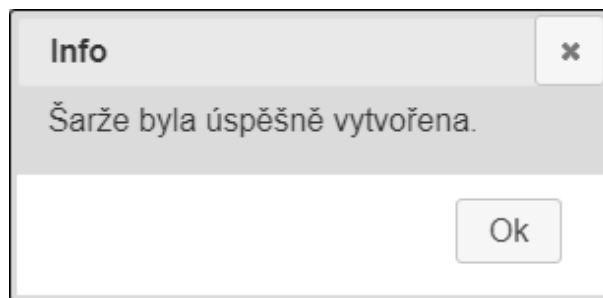
Název šarže: Číslo šarže: Datum: 08.05.2021 22:25
Přihlášený uživatel: admin

Dodavatel: Zboží: Typ zboží:

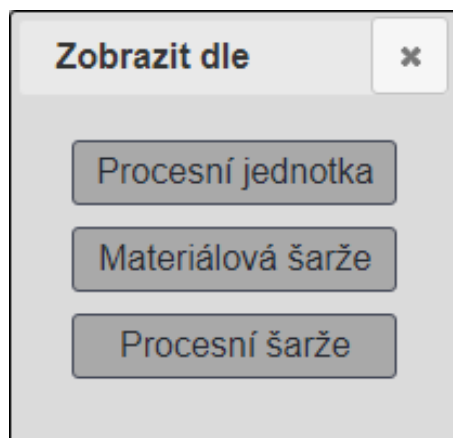
Množství: Měrná jednotka:

Poznámka:

Obr. B.4: Formulář pro založení nové šarže



Obr. B.5: Hlášení o úspěšném provedení akce



Obr. B.6: Menu pro zobrazení historických dat o šaržích

Historie procesních jednotek ✕

Procesní jednotka:

Zvolte, která data zvolené procesní jednotky chcete zobrazit:

Čas vzniku události	Čas ukončení události	Detail
2021-05-08 22:33:13	2021-05-08 22:35:43	Filling phase
2021-05-08 22:33:27	2021-05-08 22:33:27	Komponenta 2
2021-05-08 22:33:31	2021-05-08 22:33:31	Komponenta 2
2021-05-08 22:33:36	2021-05-08 22:33:45	Komponenta 1
2021-05-08 22:35:43	2021-05-08 22:36:10	Mixing phase
2021-05-08 22:36:10	2021-05-08 22:36:28	Emptying phase
2021-05-08 22:41:23	2021-05-08 22:41:59	Filling phase
2021-05-08 22:41:29	2021-05-08 22:41:37	Komponenta 1
2021-05-08 22:41:41	2021-05-08 22:41:41	Komponenta 2
2021-05-08 22:41:59	2021-05-08 22:42:03	Mixing phase

Obr. B.7: Okno pro zobrazení historie procesních jednotek

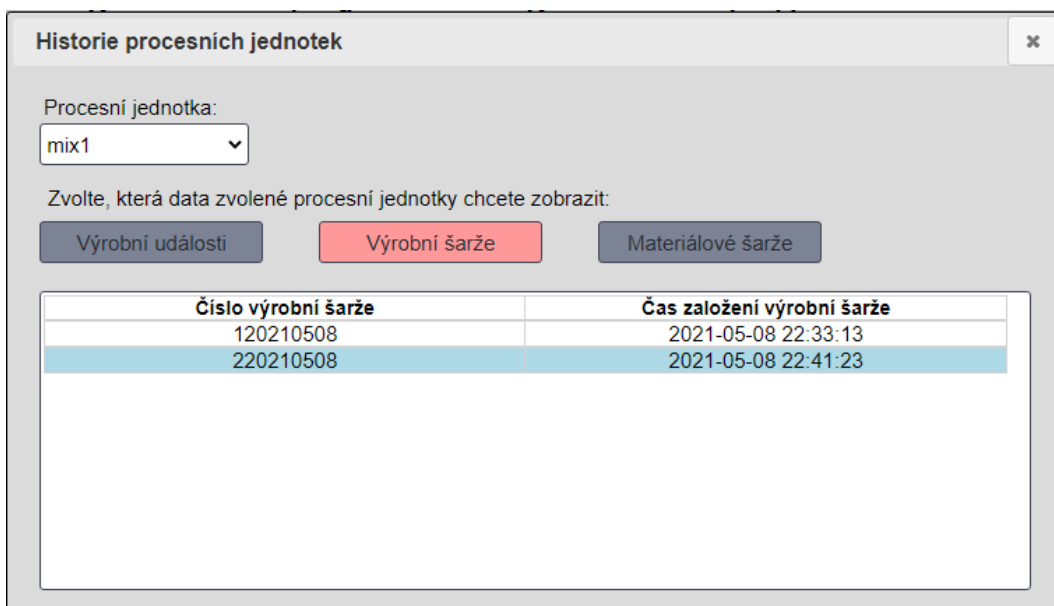
Historie procesních jednotek ✕

Procesní jednotka:

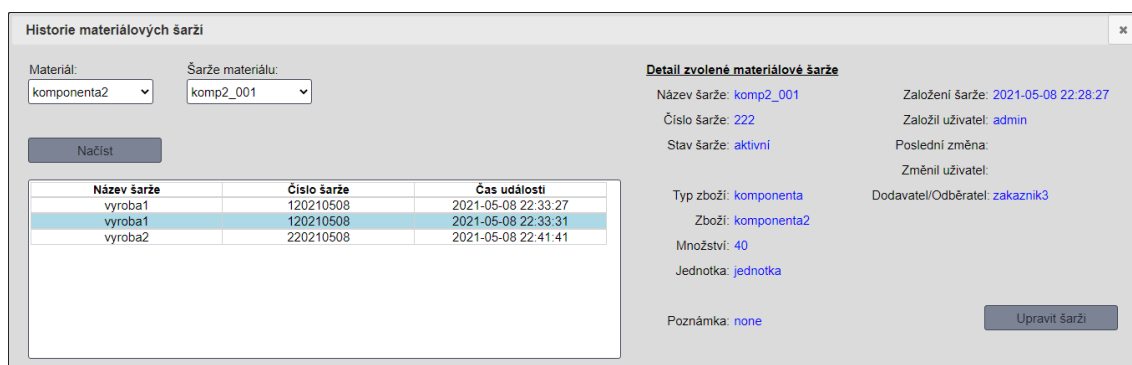
Zvolte, která data zvolené procesní jednotky chcete zobrazit:

Název materiálové šarže	Číslo materiálové šarže	Čas vzniku události
komp2_001	222	2021-05-08 22:33:27
komp2_001	222	2021-05-08 22:33:31
komp1_001	111	2021-05-08 22:33:36
komp1_002	112	2021-05-08 22:41:29
komp2_001	222	2021-05-08 22:41:41

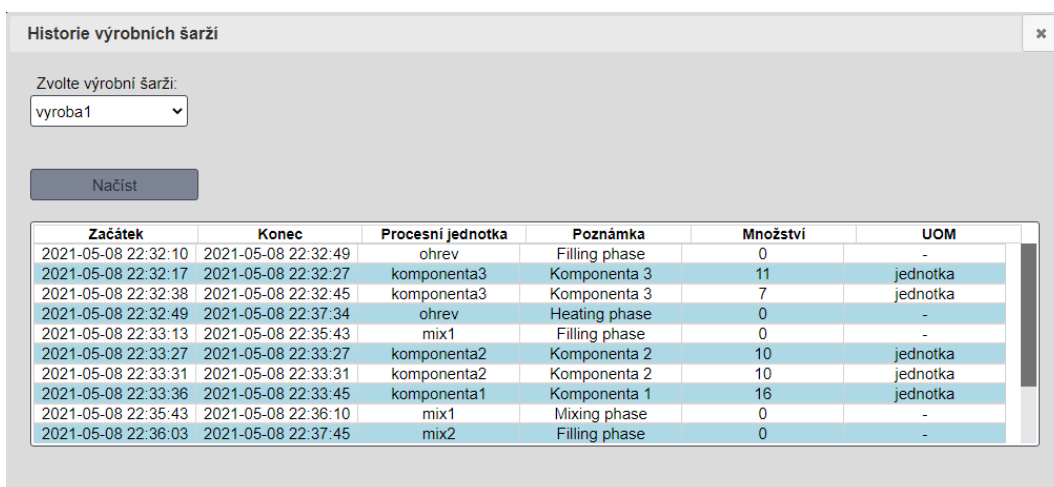
Obr. B.8: Okno pro zobrazení historie materiálových šarží



Obr. B.9: Okno pro zobrazení historie materiálových šarží



Obr. B.10: Okno pro zobrazení historie materiálových šarží



Obr. B.11: Okno pro zobrazení historie výrobních šarží

Editace šarže ✕

Název šarže: Číslo šarže: Datum: 08.05.2021 23:10
Přihlášený uživatel: admin

Dodavatel: Zboží: Typ zboží:

Množství: Měrná jednotka:

Poznámka:

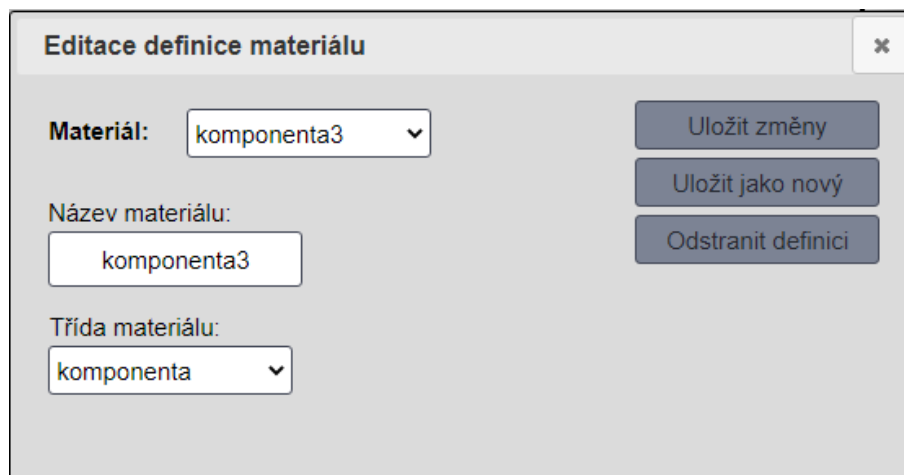
Obr. B.12: Formulář pro editaci existující šarže

Editace třídy materiálu ✕

Materiál:

Název třídy materiálu:

Obr. B.13: Formulář pro editaci třídy materiálu



Editace definice materiálu [X]

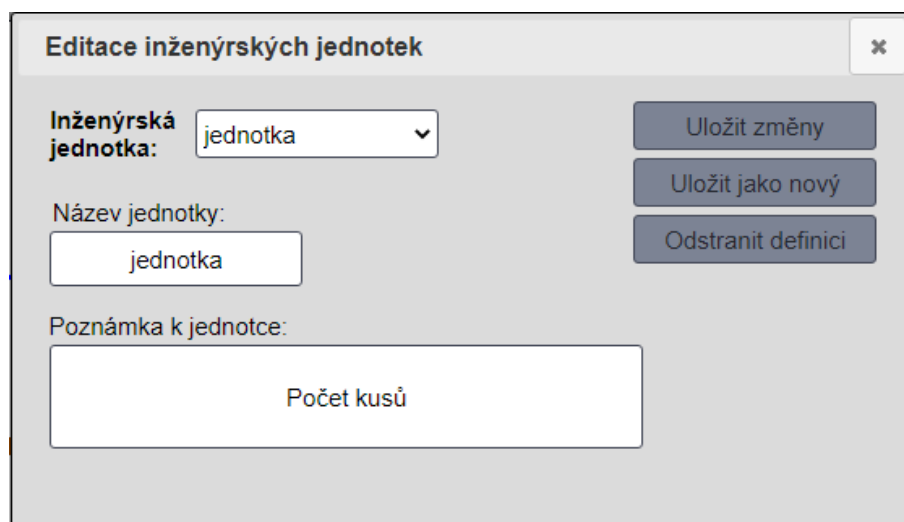
Materiál: komponenta3 ▼

Název materiálu:
komponenta3

Třída materiálu:
komponenta ▼

Uložit změny
Uložit jako nový
Odstranit definici

Obr. B.14: Formulář pro editaci definice materiálu



Editace inženýrských jednotek [X]

Inženýrská jednotka: jednotka ▼

Název jednotky:
jednotka

Poznámka k jednotce:
Počet kusů

Uložit změny
Uložit jako nový
Odstranit definici

Obr. B.15: Formulář pro editaci inženýrských jednotek

Editace uživatelů
✕

Uživatel:

Login: Heslo:

Jméno: Příjmení:

Uživatelská práva:

Obr. B.16: Formulář pro editaci uživatelů

Editace dodavatelů / zákazníků
✕

Společnost:

Název:

IČO: DIČ:

Telefon: Adresa:

Stav: Vztah:

Web:

Obr. B.17: Formulář pro editaci údajů o dodavateli/odběrateli

C Obsah přiloženého CD

```
/ ..... kořenový adresář přiloženého CD
├── diplomova_prace.pdf
├── SQL Dump.sql ..... Zálaha databáze ke vzorovému projektu
├── PLC Sarze Vzorovy projekt.zip ..... Zálaha PLC programu pro řízení vzorové
    technologie
├── SCADA Sarze Vzorovy projekt a skripty.zip SCADA aplikace, zdrojové kódy
    implementace jsou součástí
└── README.txt ..... Návod pro spuštění projektu
```