

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2022

Petr Černocký



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

DEMONSTRAČNÍ SYSTÉM VÝROBNÍ LINKY ŘÍZENÉ POMOCÍ MES S VYUŽITÍM AAS

DEMONSTRATIVE MANUFACTURING SYSTEM CONTROLLED BY MES UTILIZING AAS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Petr Černocký

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jakub Arm, Ph.D.

BRNO 2022

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Petr Černocký

ID: 220969

Ročník: 3

Akademický rok: 2021/22

NÁZEV TÉMATU:

Demonstrační systém výrobní linky řízené pomocí MES s využitím AAS

POKYNY PRO VYPRACOVÁNÍ:

Úkolem je upravit MES systém pro interakci s virtuální linkou a integrovat AAS, kde je vhodné. Ke splnění je možné využít vytvořený model demonstrační linky (nebo si vytvořit vlastní), vytvořený (nebo jiný open-source) MES a generátor AAS.

- 1) Seznamte se s technologiemi MES, OPC UA a AAS
- 2) Proveďte rešerši open-source MES systémů
- 3) Vytvořte nebo upravte model výrobní linky pro testování a validaci
- 4) Integrujte vybraný MES systém s danou virtuální linkou
- 5) Navrhněte a implementujte AAS

DOPORUČENÁ LITERATURA:

Plattform Industrie 4.0. Details of the Asset Administration Shell. Berlin, 2020. Dostupné na: <https://www.plattform-i40.de>

Termín zadání: 7.2.2022

Termín odevzdání: 23.5.2022

Vedoucí práce: Ing. Jakub Arm, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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 zákonku č. 40/2009 Sb.

Abstrakt

Tato práce zkoumá dnešní moderní možnosti pro řízení výroby. Konkrétněji se zde zaměřujeme na MES a na jeho integraci v rámci konceptu průmyslu 4.0 s využitím AAS. Dále je provedena rešerše dostupných open-source MES systémů a jejich popsání. Po jejich zprovoznění je diskutována jejich vhodnost pro další užití. Je zde také proveden rozbor nástrojů pro vytváření virtuálních modelů. Z uvedených rozborů je poté vybrána aplikace IMES, která byla posléze mírně modifikována, a nástroj pro vytváření virtuálních modelů RobotStudio. Tato aplikace a nástroj jsou následně využity pro implementaci řízení virtuální linky za pomoci samostatného MES a pro základní řízení s využitím AAS.

Klíčová slova

MES, IMES, AAS, Firebase, OPC UA, ISA95, RobotStudio, virtuální továrna

Abstract

This work examines today's modern possibilities for production management. More specifically, we focus on MES and its integration within the concept of industry 4.0 using AAS. Furthermore, a search of available open-source MES systems and their description is performed. After their commissioning, their suitability for further use is discussed. There is also an analysis of tools for creating virtual models. The IMES application, which was subsequently slightly modified, and the RobotStudio virtual model creation tool are then selected from these analyzes. This application and tool are then used to implement virtual line control using a separate MES and to basic control using AAS.

Keywords

MES, IMES, AAS, Firebase, OPCUA, ISA95, RobotStudio, virtual factory

Bibliografická citace

ČERNOCKÝ, Petr. *Demonstrační systém výrobní linky řízené pomocí MES s využitím AAS*. Bmo, 2022. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/142141>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Jakub Arm.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta: Petr Černocký

VUT ID studenta: 220969

Typ práce: Bakalářská práce

Akademický rok: 2021/22

Téma závěrečné práce: Demonstrační systém výrobní linky řízené pomocí MES s využitím AAS

Prohlašuji, že svou závěrečnou práci jsem vypracoval 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 autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si 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.

V Brně dne: 06.05.2022

podpis autora

Poděkování

Rád bych poděkoval svému vedoucímu bakalářské práce panu Ing. Jakobovi Armovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

V Brně dne: 06.05.2022

podpis autora

Obsah

| | |
|---|-----------|
| SEZNAM OBRÁZKŮ | 5 |
| ÚVOD | 6 |
| 1. ZÁKLADNÍ TECHNOLOGIE I4.0 | 7 |
| 1.1 OPC UA..... | 7 |
| 1.1.1 Specifikace OPC UA..... | 7 |
| 1.1.2 OPC UA služby (services)..... | 7 |
| 1.1.3 OPC UA Discovery server..... | 7 |
| 1.1.4 Zabezpečení OPC UA..... | 7 |
| 1.1.5 Výhody OPC UA..... | 8 |
| 1.2 AAS – ASSET ADMINISTRATION SHELL..... | 8 |
| 1.2.1 Struktura AAS..... | 9 |
| 1.2.2 Hlavní vlastnosti AAS..... | 10 |
| 1.3 MES..... | 10 |
| 1.3.1 Základní funkce MES dle MESA..... | 10 |
| 1.3.2 MES/MOM dle ISA 95..... | 12 |
| 1.4 KLÍČOVÉ INDIKÁTORY VÝKONU (KPI)..... | 14 |
| 1.4.1 OEE – Celková efektivnost zařízení (Overall equipment effectiveness)..... | 14 |
| 1.5 B2MML..... | 14 |
| 2. VYBRANÉ OPEN SOURCE MES | 15 |
| 2.1 QCADOO MES..... | 15 |
| 2.1.1 Struktura Qcadoo MES..... | 15 |
| 2.1.2 Modulárnost..... | 16 |
| 2.1.3 Základní moduly Qcadoo MES [13]..... | 17 |
| 2.1.4 Zprovoznění..... | 18 |
| 2.1.5 Shrnutí..... | 20 |
| 2.2 IMES..... | 20 |
| 2.2.1 Struktura IMES..... | 20 |
| 2.2.2 Firebase databáze..... | 21 |
| 2.2.3 Základní funkce IMES..... | 22 |
| 2.2.4 Zprovoznění aplikace..... | 23 |
| 2.2.5 Shrnutí..... | 25 |
| 3. NÁSTROJE PRO VYTVÁŘENÍ VIRTUÁLNÍCH MODELŮ | 26 |
| 3.1 ABB ROBOT STUDIO..... | 26 |
| 3.1.1 Modelovací nástroje ABB Robot studio [16]..... | 26 |
| 3.2 SIEMENS NX MECHATRONICS CONCEPT DESIGNER..... | 27 |
| 3.3 REALIZACE SIMULACE..... | 27 |
| 3.3.1 Výhody a nevýhody ABB Robot Studia..... | 27 |
| 3.3.2 Demonstrační linka..... | 28 |
| 3.3.3 Logika Stanice..... | 28 |
| 3.3.4 Tvorba OPC Serveru..... | 29 |
| 4. ŘÍZENÍ VÝROBY ZA POMOCÍ MES BEZ VYUŽITÍ AAS | 31 |

| | | |
|-----------|--|-----------|
| 4.1 | CHARAKTERISTICKÉ ZNAKY | 31 |
| 4.2 | IMES..... | 31 |
| 4.3 | POPIS OVLÁDÁNÍ A MODIFIKACÍ IMES..... | 32 |
| 4.3.1 | <i>Nastavení továrny</i> | 32 |
| 4.3.2 | <i>Plánování výroby</i> | 32 |
| 4.3.3 | <i>Trasování výroby</i> | 33 |
| 4.4 | KOMUNIKAČNÍ BLOKOVÉ SCHÉMA ŘÍZENÍ | 35 |
| 4.4.1 | <i>Popis komunikace</i> | 35 |
| 4.4.2 | <i>Propojení za pomoci nástroje NODE-RED</i> | 36 |
| 4.5 | VÝHODY A NEVÝHODY | 36 |
| 5. | ŘÍZENÍ VÝROBY S VYUŽITÍM AAS..... | 37 |
| 5.1 | CHARAKTERISTIKA TOHOTO DRUHU ŘÍZENÍ..... | 37 |
| 5.2 | BLOKOVÉ SCHÉMA..... | 37 |
| 5.3 | VYTVÁŘENÍ AAS PRO MES..... | 38 |
| 5.3.1 | <i>Použitý nástroj</i> | 38 |
| 5.3.2 | <i>Vytváření AAS a jeho submodelů</i> | 38 |
| 5.4 | VÝSLEDNÉ SUBMODELÝ..... | 39 |
| 5.4.1 | <i>Submodel Material Requirement</i> | 39 |
| 5.4.2 | <i>Submodel Job Order</i> | 39 |
| 5.4.3 | <i>Submodel Physical Asset Requirement</i> | 40 |
| 5.4.4 | <i>Submodel Job List</i> | 40 |
| 5.4.5 | <i>Submodel Work Schedule</i> | 40 |
| 5.4.6 | <i>Submodel Work Request</i> | 41 |
| 5.4.7 | <i>Submodel Equipment Requirement</i> | 41 |
| 5.4.8 | <i>Submodel Work Response</i> | 41 |
| 5.4.9 | <i>Submodel Job Response</i> | 42 |
| 5.5 | TVORBA OPC SERVERU..... | 42 |
| 5.6 | SHRNUTÍ | 43 |
| 6. | ZÁVĚR..... | 44 |
| | LITERATURA..... | 45 |
| | SEZNAM SYMBOLŮ A ZKRATEK..... | 47 |
| | SEZNAM PŘÍLOH..... | 48 |

SEZNAM OBRÁZKŮ

| | | |
|------|--|----|
| 1.1 | AAS existuje pro produkty, zařízení i proces [4]..... | 9 |
| 1.2 | Struktura AAS [5]..... | 9 |
| 1.3 | Podniková hierarchie [7]..... | 10 |
| 1.4 | Model MES dle MESA [8]..... | 11 |
| 1.5 | Model MES/MOM ISA95 [9] | 13 |
| 2.1 | Struktura webové aplikace Qcadoo MES [12]..... | 15 |
| 2.2 | Ukázka možných modulů aplikace..... | 17 |
| 2.3 | Přihlašovací obrazovka Qcadoo Mes..... | 19 |
| 2.4 | Struktura aplikace IMES [14]..... | 21 |
| 2.5 | Struktura databáze IMES [14]..... | 22 |
| 2.6 | Možnosti trasování výroby za pomoci externího senzoru [15] | 23 |
| 2.7 | Úvodní obrazovka IMES | 23 |
| 2.8 | Přihlašovací obrazovka IMES | 25 |
| 3.1 | Ukázka demonstrační linky..... | 28 |
| 3.2 | Připojení k vybranému kontroléru..... | 30 |
| 4.1 | Ukázka přenosu dat mezi MES a virtuální linkou..... | 31 |
| 4.2 | Stavy stanice | 33 |
| 4.3 | Vývojový diagram pro vytvoření a zavedení objednávky do výroby | 33 |
| 4.4 | Vývojový diagram pro trasování výroby | 34 |
| 4.5 | Blokové schéma „klasického“ řízení výroby..... | 35 |
| 4.6 | Ukázka spojení databáze s RobotStudiem za pomoci nástroje NodeRed..... | 36 |
| 5.1 | Blokové schéma řízení výroby za pomoci AAS (část obrázku použita z [5])..... | 37 |
| 5.2 | Hlavní obrazovka AASX Package Exploreru..... | 38 |
| 5.3 | Submodel MaterialRequirement..... | 39 |
| 5.4 | Submodel Job Order..... | 39 |
| 5.5 | Submodel Physical Asset Requirement..... | 40 |
| 5.6 | Submodel Job List..... | 40 |
| 5.7 | Submodel Work Schedule..... | 40 |
| 5.8 | Submodel Work Request | 41 |
| 5.9 | Submodel Equipment Requirement..... | 41 |
| 5.10 | Submodel Work Response..... | 42 |
| 5.11 | Submodel Job Response..... | 42 |

ÚVOD

Tato práce se zabývá rozbohem možností vytvoření virtuální linky a ovládním virtuální výroby za pomoci MES a dále základní řízení s využitím AAS. V první kapitole jsou rozebrány technologie, které se využívají v oblasti průmyslu 4.0. A také způsoby komunikace mezi MES úrovní hierarchie řízení podniku a demonstrační výrobní linkou.

V další kapitole je proveden rozbor dostupných open source MES. Je zde popsána jejich funkčnost a způsoby zprovoznění. Dále jsou zde uvedeny jejich výhody a nevýhody a možné způsoby nasazení do reálného provozu.

Ve třetí kapitole jsou představeny dva nástroje pro tvorbu virtuálních modelů. Jsou zde uvedeny jejich možnosti a výhody. Tyto poznatky jsou poté převedeny do praxe. Dále je zde uveden způsob vytváření OPC serveru pro komunikace s nadřazenými systémy.

Následující kapitola se věnuje standardnímu řízení výroby za pomoci zvolené MES aplikace. Jsou zde popsány provedené změny v aplikaci a stručný popis jejího ovládní.

Poslední kapitola se věnuje vytvoření virtuální obálky (AAS) pro použitý MES systém. Jsou v ní navrženy submodely pro následnou komunikaci dle standardu ISA95. Na konci kapitoly se nachází zhodnocení a je zde navržen další možný postup řešení tohoto problému.

Celá práce má za cíl zjistit a zanalyzovat výhody, nevýhody a možné postupy při tvorbě a implementaci těchto systémů. A dále zprovoznit virtuální linku a spojit ji s nadřazeným MES systémem.

1. ZÁKLADNÍ TECHNOLOGIE I4.0

V této kapitole budou rozebrány a popsány základní způsoby komunikace se zařízeními a základní technologie, které se využívají v oblasti průmyslu 4.0.

1.1 OPC UA

OPC komunikace slouží obecně k výměně dat mezi různými průmyslovými systémy. V automatizaci je to univerzální komunikační platforma, která se dokáže napojit na data stovek různých typů zařízení od různých výrobců a převést tato data do jednotné OPC komunikace, srozumitelné mnoha nadřazeným aplikacím jako např. ERP, SCADA, nebo v tomto případě MES. [1]

Jedním z důvodů vzniku OPC UA byla roztržitost mezi specifikacemi OPC DA, OPC AE apod. Dalším důvodem byl problém, že OPC DA využíval technologii COM/DCOM, a proto fungoval pouze na operačních systémech Windows.[1]

1.1.1 Specifikace OPC UA

Specifikace OPC UA je založená na předávání dat mezi OPC UA klientem a OPC UA serverem založená na mapování a navazování spojení. OPC UA protokol je specifikován jako SOA, tedy architektura orientována na služby, což znamená, že se klient může dotazovat na konkrétní služby a server na každý dotaz reaguje příslušnou odpovědí. Služby, které server poskytuje, vytváří abstraktní komunikační model. Popis celého adresního prostoru OPC serveru se provádí za pomoci uzlů [2]

1.1.2 OPC UA služby (services)

Komunikace mezi klientem a serverem probíhá pomocí volání a zpracovávání služeb. Jedná se vlastně o metody, pomocí kterých získáváme informace ze serveru OPC UA. [2]

1.1.3 OPC UA Discovery server

Mezi další vlastnosti OPC UA je vhodné zmínit funkci Discovery serverů. Jedná se o funkci vyhledávání OPC serverů. Bez této vlastnosti musí klient vědět přesnou adresu serveru. Discovery servery se dělí na dva typy:

- Local Discovery server – slouží pro propojení na lokální úrovni např. v rámci jedné sítě. [19]
- Global Discovery server

1.1.4 Zabezpečení OPC UA

V dnešní době je nutné brát ohledy na různé kybernetické útoky. V průmyslu je taky velkým tématem zabezpečení. Pokud by se stalo, že by mohla být napadena nějaká továrna, může to mít nedožrnné následky nejen na výrobu.

Zabezpečení OPC UA je dvojího druhu, a to vnější a vnitřní. Jako vnější zabezpečení se rozumí například zabezpečení pomocí šifrování a podepisování zpráv nebo využití zabezpečeného kanálu.

Jako vnitřní zabezpečení OPC využívá zavírání přebytečných a neužitečných spojení. Každý účastník komunikace musí disponovat certifikát, který identifikuje aplikaci a stroj, se kterými komunikuje. [2]

Základní úrovně zabezpečení z pohledu uživatelských práv

Tato zabezpečení omezují přístupy různým uživatelům. OPC UA má k dispozici tyto čtyři úrovně:

- **Úroveň 1** – bez autentizace. Každý platný certifikát je označen za důvěryhodný. [2]
- **Úroveň 2** – serverová autentizace. Tato úroveň dovoluje opět připojení jakéhokoliv klienta, ovšem je zde nutno provést ověření pomocí zadání jména/hesla. [2]
- **Úroveň 3** – klientská autentizace. Tato možnost umožňuje připojit se pouze důvěryhodným klientům. [2]
- **Úroveň 4** – oboustranná autentizace tzn. klient i server umožňují připojení pouze důvěryhodným certifikátům. [2]

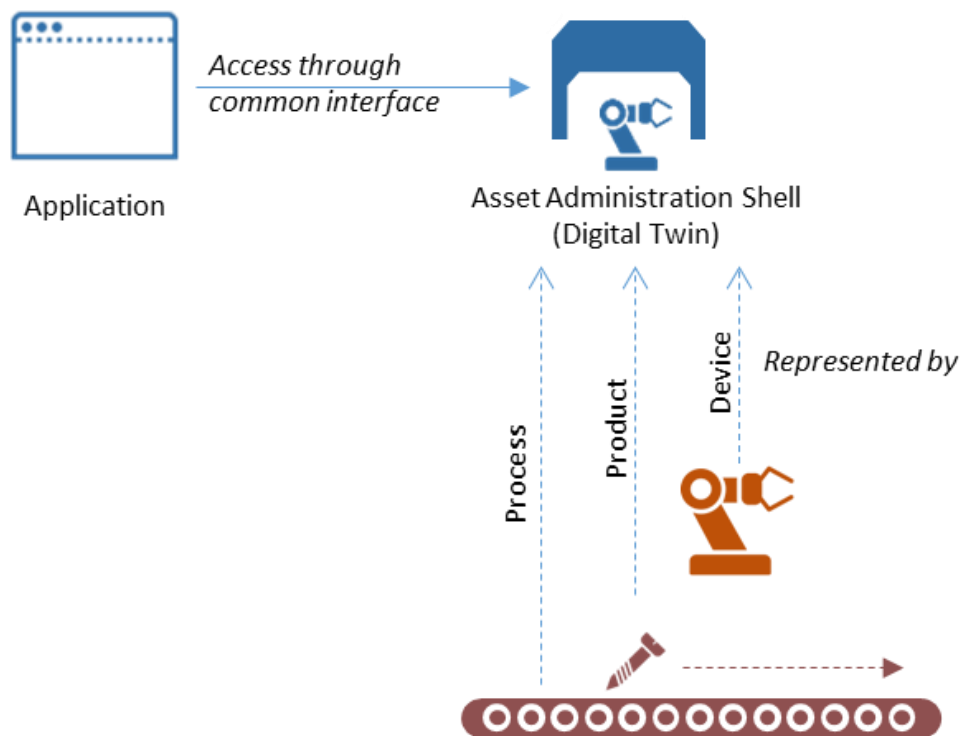
1.1.5 Výhody OPC UA

Mezi největší výhody patří zejména jednoduchost a na rozdíl od přechozích verzí OPC není toto řešení nijak závislé na využívaném operačním systému. Tento protokol je tedy univerzálnější a navíc bezpečnější. Spoustu průmyslových zařízení či softwarů již disponují vestavěným OPC UA serverem a připojený klient ve formě např. PC softwaru může zobrazovat data z PLC či jiného zařízení. [2]

Jako nevýhodu lze uvést nekompatibilitu s předchozími verzemi OPC, proto by přechod, z jichž fungujících aplikací mohl být obtížnější. [2]

1.2 AAS – Asset administration shell

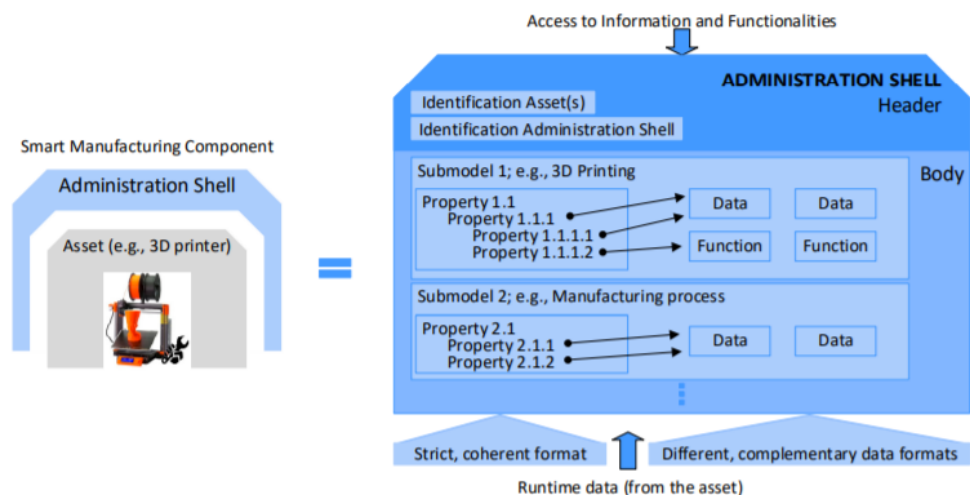
Koncepce Industrie 4.0 využívá ke standardizovanému popisu zařízení jejich digitální obálky AAS – Asset Administration Shell. Účelem těchto obálek je zajistit výměnu informací mezi zařízeními navzájem, mezi nimi a systémem koordinace výroby a inženýrskými nástroji. [3]



Obrázek 1.1 AAS existuje pro produkty, zařízení i proces [4]

1.2.1 Struktura AAS

Na obrázku je vidět popis a spojení mezi fyzickým zařízením a AAS. Obálka zařízení (AAS) se skládá ze dvou částí. Hlavičky (Header), ve které jsou uvedeny unikátní identifikátory zařízení. A těla (Body), ve kterém jsou uvedeny ostatní informace o zařízení, jeho vlastnosti a jiné důležité informace např. o procesu výroby. [5]



Obrázek 1.2 Struktura AAS [5]

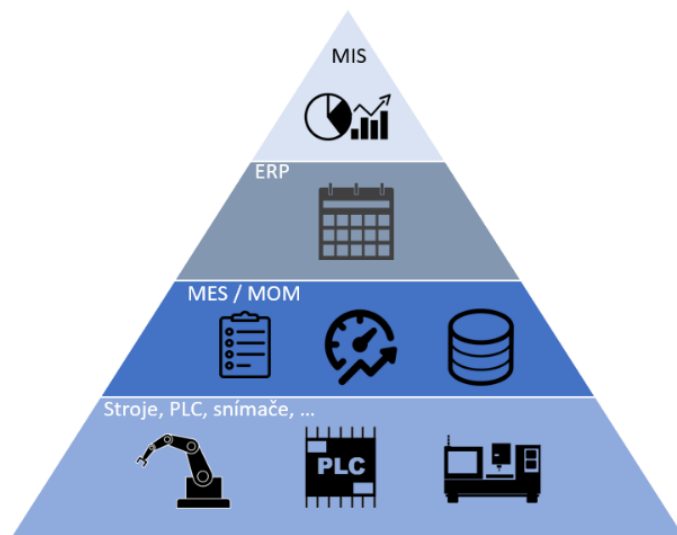
1.2.2 Hlavní vlastnosti AAS

- Komponenta výroby v I4.0, je fyzická komponenta (Asset) v kombinaci se svojí AAS.
- AAS je standardizovaná digitální reprezentace komponenty, která je jádrem interoperability mezi komponentami pro jejich monitorování, řízení a další aplikace. [6]
- AAS může být digitální reprezentací jednotlivé komponenty (stroje, řídicího systému), ale i celého systému (výrobního úseku, podniku) libovolné úrovně hierarchické architektury řízení podniku. V této práci bude využito právě této vlastnosti. Viz obrázek 1.1. [6]
- AAS pokrývá celý životní cyklus výrobku, komponenty, stroje či zařízení. [20]
- AAS je digitálním základem pro systémy využívající umělou inteligenci. [20]

1.3 MES

Slovo MES je anglickou zkratkou pro Manufacturing Execution System, která se překládá do češtiny jako Výrobní informační systémy. Jedná se o výpočetní softwarové systémy používané ve výrobních podnicích pro řízení a monitoring výrobních procesů. Dále pak napomáhají pracovníkům ve výrobě přijímat důležité rozhodnutí, či odhalit případný problém co nejdříve, což vede k celkovému zvyšování efektivity výroby. [7]

Dle podnikové hierarchie se MES řadí - viz. obrázek 1.3

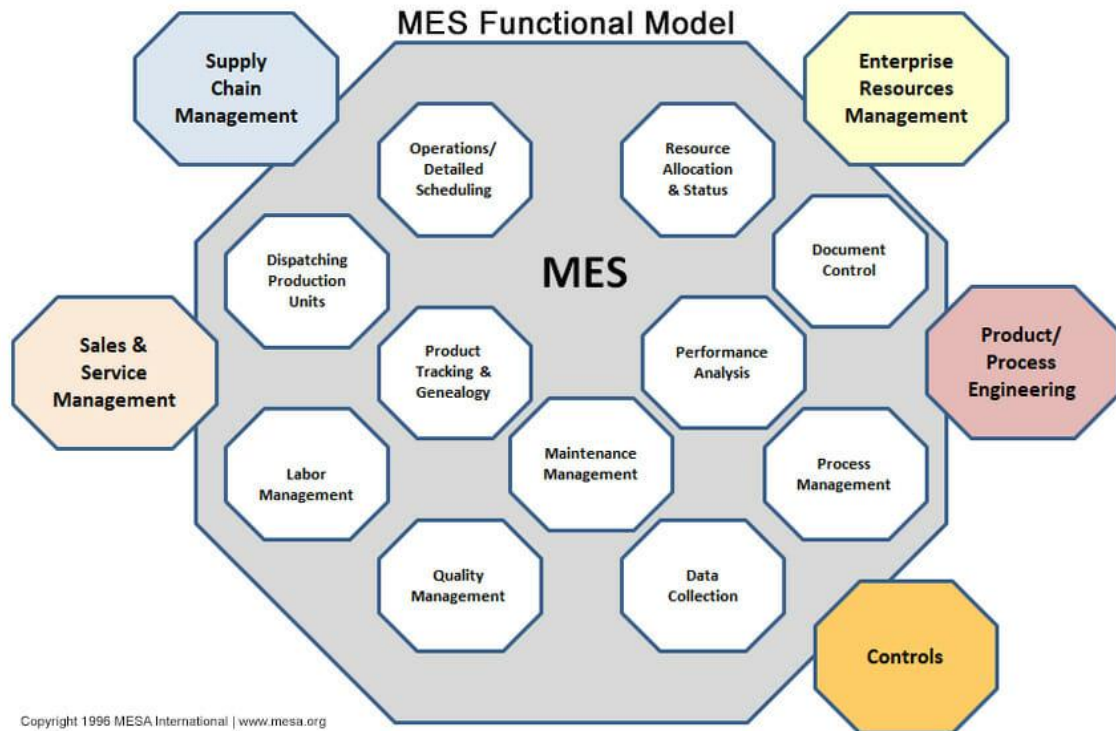


Obrázek 1.3 Podniková hierarchie [7]

1.3.1 Základní funkce MES dle MESA

Za základní funkcionality MES systémů lze považovat 11 základních funkcionalit, které byly poprvé vydefinovány organizací MESA International v roce 1992 v modelu MESA-11 model. Tento model se postupně vyvíjel přes Kontextový model vydaný roku 1996 a

dále Kolaborativní model vydaný v roce 2002 až k aktuálnímu základnímu modelu organizace MESA International s názvem MESA Model, který byl vydaný v roce 2006. [8]



Obrázek 1.4 Model MES dle MESA [8]

Jednotlivé funkce:

Dle modelu MESA-11 by měl MES obsahovat tyto funkce:

- **Operativní plánování výroby**
Tato funkce má za cíl definovat frontu práce, a to v jakém pořadí se budou zpracovávat jednotlivé výrobní příkazy. Díky této funkci můžeme eliminovat zbytečné seřizování strojů, délku prostojů zařízení atd. [9]
- **Řízení a přidělování zdrojů**
Tato funkce má za cíl přidělovat a sledovat zdroje pro výrobní proces. Informace je založena na aktuálním stavu a budoucích rezervacích těchto zdrojů. [9]
- **Dispečerské řízení**
Tato funkce má za cíl přiřazovat práci jednotlivým zařízením a osobám, zajišťovat potřebné množství surovin a energie a sledování aktuálního stavu výroby – případně operativní řešení výpadků. [9]
- **Řízení výrobního procesu**
Plní funkci řízení výroby specifikované v naplánované a uvolněné výrobě (frontě práce). [9]

- **Sběr a archivace dat a správa (dat)**
Tato funkce shromažďuje procesní a výrobní data a dále např. stavy jednotlivých zařízení. [9]
- **Analýza výkonnosti**
Slouží k vyhodnocení úspěchu podniku, případně k vyhodnocování úspěšnosti v jednotlivých oblastech celého výrobního procesu. Využívá se zde sběr a archivace dat. Sledujeme zde KPI, což jsou klíčové výkonnostní ukazatele, mezi něž patří např. OEE, což je celková efektivita zařízení. [9]
- **Genealogie a trasování výroby**
Slouží k sledování výrobků a jejich rodokmenu. [9]
- **Správa a dokumentace**
- **Řízení údržby**
Má za úkol zabezpečit při optimálních nákladech co nejlepší technický stav zařízení a jeho spolehlivost po čas životního cyklu. [9]
- **Řízení laboratoře**
Zajišťuje systém LIMS – Laboratory Information and Management System. [9]
- **Řízení kvality**
Má za úkol analyzovat data snímaná z výroby v reálném čase s cílem sledování vyráběného produktu, včetně časné identifikace nežádoucích odchylek. [9]

1.3.2 MES/MOM dle ISA 95

Standard ISA 95 je mezinárodní standard vytvořený neziskovou organizací ISA. Tento standard řeší integraci podnikových a řídicích systémů. Mezi hlavní cíle patří Definice modelů pro funkce a datové toky na úrovni MES a integraci podnikové domény a řídicí domény. [9]

Hlavní rozdílem mezi MES a MOM je ten, že MES je nějaké softwarové vybavení – produkt, který lze koupit. Na druhou stranu MOM koupit nelze – je to nějaké řešení, způsob myšlení a komunikace. Činnosti MOM lze rozdělit do 4 kategorií – pilířů.

1) Řízení operací výroby

Zabývá se činnostmi, co koordinují, usměrňují, řídí a monitorují funkce, používající suroviny, energie, zařízení, personál a informace pro vytvoření výrobku. [9]

2) Řízení operací kvality

Zabývá se činnostmi, co koordinují, usměrňují a monitorují funkce. Tyto funkce měří, vyhodnocují a oznamují kvalitu výrobku. [9]

3) Řízení operací skladování

Obsahuje činnosti, které koordinují, usměrňují a monitorují funkce. Tyto funkce sledují stav výrobků a materiálů a jejich pohyb během celé výroby. [9]

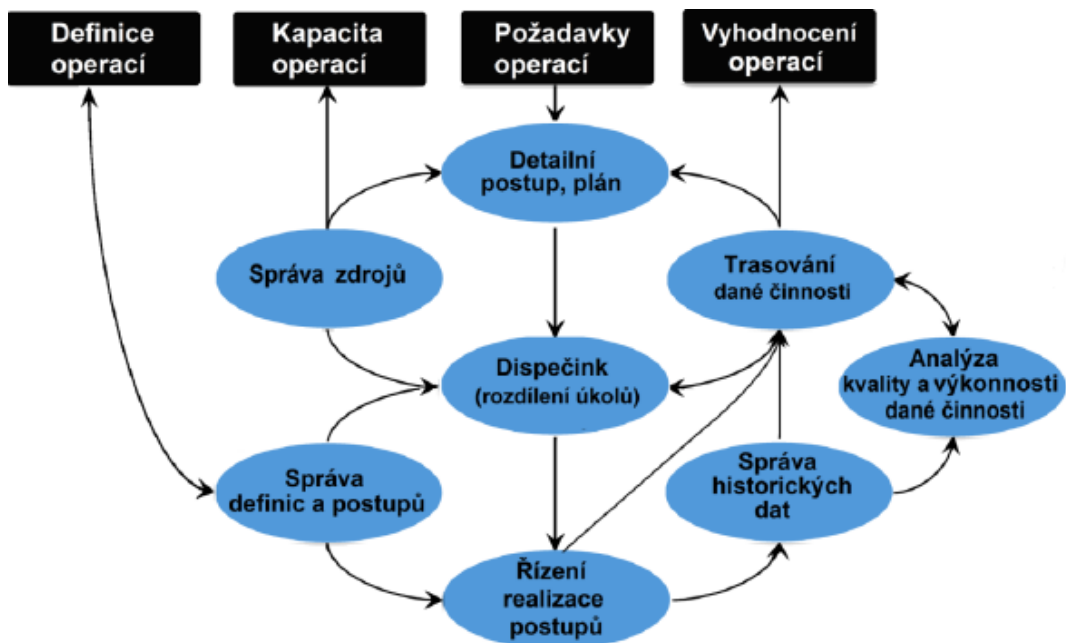
4) Řízení operací údržby

Obsahuje činnosti, které koordinují, usměrňují a monitorující funkce. Tyto funkce udržují výrobní zařízení, nástroje a podpůrná výrobní příslušenství tak, aby byla dostupná a připravena pro výrobu. [9]

Funkce generického modelu

Pro popsání činností byl stanoven generický model ISA95, který obsahuje těchto 8 funkcí.

- Správa informací
- Správa zdrojů
- Detailní postupy
- Dispečink
- Provedení
- Trasování
- Analýza kvality a výkonnosti
- Sběr dat



Obrázek 1.5 Model MES/MOM ISA95 [9]

Z obrázku je patrné, že všech osm funkcí se vztahuje do čtyř oblastí informací, které si vzájemně předávají úrovně 3 a 4 pyramidového modelu řízení.

Popis jednotlivých oblastí

Následující čtyři oblasti, pak obsahuje každý pilíř MOM.

1) Definice operací

Definují se zde činnosti – jakým způsobem se produkt vyrobí a jak se zajistí jeho kvalita a skladování. [9]

2) Kapacita operací

Schopnost, kapacita a možnosti pro danou činnost – co je k dispozici pro výrobu, kvalitu, skladování a údržbu zařízení. [9]

3) Požadavky operací

Obsahuje postup, plán a harmonogram dané činnosti. [9]

4) Vyhodnocení operací

Obsahuje souhrnné informace o dané činnosti. [9]

1.4 Klíčové indikátory výkonu (KPI)

KPI jsou definovány jako kvantifikovatelná měření na jejichž základě vyhodnocujeme faktory úspěchu či neúspěchu organizace. Obecně se dají vypočítat porovnáním měřené veličiny proti cíli. KPI reprezentují, co se právě děje se sledovaným výrobním procesem. [10] Těchto ukazatelů může být celá řada. V této práci se bude pracovat s ekonomickým ukazatelem OEE.

1.4.1 OEE – Celková efektivnost zařízení (Overall equipment effectiveness)

OEE patří mezi nejpoužívanější indikátory výkonu podniků. Hodnotu tohoto ukazatele využívají podniky pro odhalení možných mezer ve výrobě a tím pádem i pro celkové zefektivnění výroby a tím pádem i pro navýšení možných zisků celého podniku.

Tento ukazatel je vyčíslován v procentech a lze jej vypočítat dle vzorce

$$OEE = Dostupnost * Výkon * Kvalita * 100\% , \quad (1.4)$$

kde *dostupnost* se rozumí jako ztráty, které jsou způsobeny prostoji a poruchami strojů, *výkon* jsou ztráty vzniklé využitím nižší výrobní kapacity zařízení, než je možné a *kvalita* jsou ztráty způsobené z nekvalitních produktů. [11]

1.5 B2MML

Jedná se o XML implementaci standardů ISA95. Využívá se k výměně informací mezi ERP a MES/MOM. B2MML pomáhá integrovat podnikové systémy ERP a systémy pro řízení dodavatelských řetězců s výrobními MES systémy. [9]

2. VYBRANÉ OPEN SOURCE MES

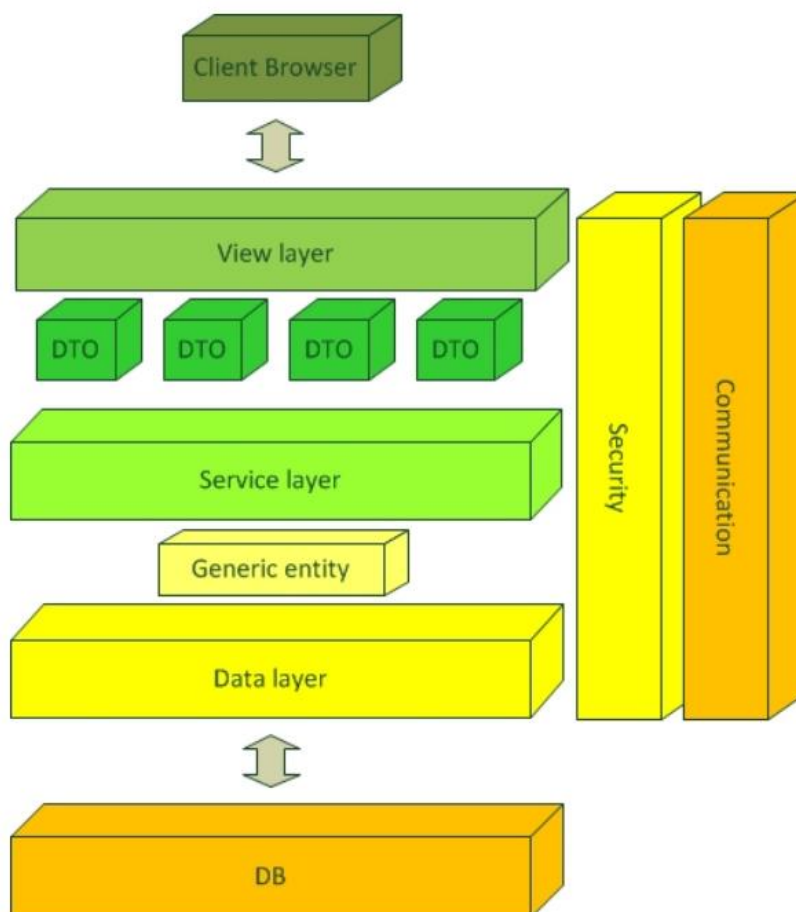
Na internetu se nachází spousta možných odkazů k MES programům. Avšak volně přístupných je pouze zlomek z nich. V rámci této práce byly vybrány pro popis dva z nich.

2.1 Qcadoo MES

První z open source MES programů je od polských tvůrců. Tento program firma poskytuje jak speciálně pro jednotlivé střední podniky tzv. „Šité na míru“, tak i jako volně přístupný program pro kohokoliv. QCADOO MES kombinuje funkce MES, ERP a MRP.

2.1.1 Struktura Qcadoo MES

Aplikace se sestává z několika vrstev viz obrázek 2.1



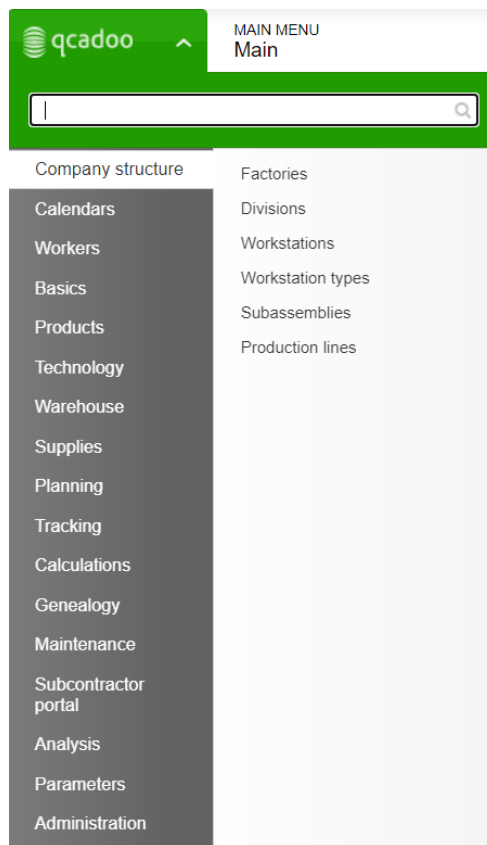
Obrázek 2.1 Struktura webové aplikace Qcadoo MES [12]

- **Databáze**
V databázi jsou uložena veškerá data aplikace.

- **Datová vrstva**
Tato vrstva zabezpečuje tok dat z databáze do vyšších vrstev a naopak. [12]
- **Generická entita**
Jedná se o objekt pro přenos dat, který se používá k výměně dat mezi datovou vrstvou a vrstvou služeb. [12]
- **Vrstva služeb**
Tato vrstva je zodpovědná za poskytování jednoho společného rozhraní pro přístup k databázi. [12]
- **Vrstva DTO**
Pro lepší separaci vrstev využíváme DTO objekty, což je návrhový vzor používaný pro přenos dat mezi SW a subsystemy, vytvořený pomocí JSON formátu. [12]
- **Prezentační vrstva**
Slouží k reprezentaci dat pro uživatele. [12]
- **Client browser**
Jedná se o webový prohlížeč zodpovědný za všechny operace na straně klienta.[12]

2.1.2 Modulárnost

Mezi další vlastnosti tohoto systému patří modulárnost. Celý systém se skládá z jednotlivých modulů, které lze editovat a přizpůsobit tak potřebám konkrétního podniku. Každý modul – plugin může obsahovat další podmoduly. Díky této vlastnosti se aplikace hodí i do menších podniků, které mají ambice se rozrůst. V případě růstu, či zařazení nové služby podniku se do aplikace jednoduše nahraje nový plugin s požadovanou funkcí.



Obrázek 2.2 Ukázka možných modulů aplikace

2.1.3 Základní moduly Qcadoo MES [13]

Základní moduly samozřejmě obsahují i další podmoduly. Mezi hlavní moduly – funkce patří:

- **Manufacturing orders**
Tato funkce umožňuje plánování výroby, plánování výroby na jednotlivé směny, pracovní pokyny pro jednotlivé sestavy apod.
- **Material requirements**
Umožňuje plánování a monitorování stavu materiálu, dále pak generuje RFQ dokumenty pro jednotlivé výrobky. Přesně určí, které suroviny a jaké množství je potřeba pro výrobu, případně generuje objednávky.
- **Delivers purchasing**
Správa dodavatelů a nákupních objednávek.
- **Subcontractors**
Správa subdodavatelů, správa technické dokumentace pro subdodavatele.
- **Product genealogy**
Sledovatelnost šarží a dohledání použitého materiálu – genealogický strom produktu.

- **Plan manufacturing orders**
Plánování výrobních procesů, porovnává několik aspektů, jako např. výkonnost zařízení, pracovní směny, dostupnost materiálu atd. Z těchto informací poté naplánuje, kdy se bude daná zakázka vykonávat.
- **Calculations**
Výpočty časových náročností zakázek, výpočty celkových nákladů na zakázku apod.
- **Production accounting**
Vyúčtování výrobních zakázek, informací o ziscích a ztrátách.
- **Warehouse management**
Správa skladových zásob a všech ostatních dokumentů.

2.1.4 Zprovoznění

Aplikaci se bohužel nepodařilo zprovoznit na operačním systému Windows 10. Ani po zdlouhavém ladění se nedařilo odstranit chybu, související s chybějícími qcadoo balíčky. Proto bylo využito virtuálního zařízení (VirtualBox) s OS na bázi Linuxu. V tomto případě OS Ubuntu. Postup spuštění je uveden v několika krocích, které jsou popsány níže. Všechny příkazy jsme zadávali do terminálu.

- 1) Nejprve bylo nutné stáhnout a nainstalovat balíček JAVA 8 SDK. Toho se docílilo za pomoci příkazu níže.

```
sudo apt-get install openjdk-8-jdk
```

- 2) Dále bylo nutné nainstalovat Postgres databázi 9.x.x. Tu lze získat ze stránek na odkazu <https://www.postgresql.org/download/>. Vybereme zde dle správnou verzi našeho Linux systému. Dále pokračujeme podle uvedeného návodu. A nainstalujeme databázi. Je nutné používat verzi Postgres 9.x.x. U vyšších verzí některé příkazy nefungovaly.

- 3) Dále nastavíme heslo k uživateli Postgres za pomoci:

```
sudo passwd postgres
```

- 4) Za pomoci dalšího příkazu připojíme databázi

```
sudo su postgres -c psql
```

- 5) Dále se pokračuje v příkazovém řádku postgres – nejdříve nastavením uživatele:

```
ALTER USER postgres with PASSWORD 'postgres123';
```

- 6) Vytvoření databáze s názvem „mes“.

```
CREATE DATABASE mes WITH ENCODING 'UTF8' LC_COLLATE 'en_US.UTF-8' LC_CTYPE 'en_US.UTF-8' TEMPLATE template0;
```

7) Stáhneme aktuální verzi Qcadoo MES z Github serveru na odkazu <https://github.com/qcadoo/mes/releases>. A extrahujeme stažené soubory např. na plochu.

8) Před prvním spuštěním, je ještě nutno nahrát demo databázi. Nejdříve použijeme příkaz:

```
sudo su postgres
```

a následně:

```
psql -U postgres mes < cesta/k/souboru/demo_db_en.sql
```

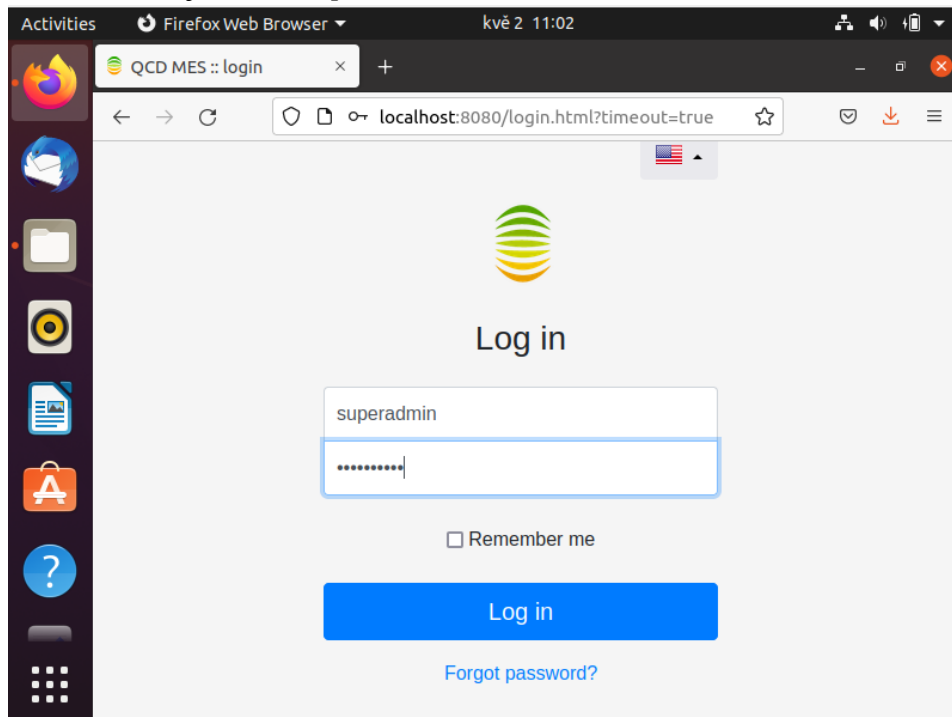
Soubor `demo_db_en.sql` se v mém případě nachází – Desktop/mes-application/webapps/ROOT/WEB-INF/classes/schema.

Po tomto příkazu dojde k vytvoření tabulek. A aplikace je připravena k používání.

9) Spuštění aplikace provedeme ve složce bin za pomoci příkazu:

```
./startup.sh
```

10) Do prohlížeče poté stačí napsat `localhost:8080` a chvíli počkat, dokud se nezobrazí přihlašovací obrazovka. Pak už zbývá pouze zadat uživatelské jméno *superadmin* a heslo, které je taktéž *superadmin*.



Obrázek 2.3 Přihlašovací obrazovka Qcadoo Mes

2.1.5 Shrnutí

Qcadoo MES je aplikace od polských vývojářů, která stojí na PostgreSQL databázi. Data jsou zde posílána pomocí datové vrstvy. Díky tomu, že aplikace využívá pluginy – moduly, lze ji využít téměř v každém podniku. Aplikace kombinuje možnosti MES a ERP systémů. Je to spíše takový hybrid. MES systém má téměř všechny funkce dle MESA-11, schází zde hlavně modul pro řízení údržby a pro řízení laboratoře. Hlavní nevýhodou této aplikace je to, že se ji nepovedlo zprovoznit pro operační systém Windows, který je u dnešních stolní počítačů nejběžnější. Aplikace ovšem funguje bezchybně na Linux operačních systémech.

2.2 IMES

IMES patří mezi další open source aplikace volně stažitelné z portálu Github. Byla vytvořena jak pro malé, tak i pro středně velké podniky, které by si nemohly dovolit investovat do stále poměrně drahých MES systémů. Jednou z výhod této aplikace může být i to, že díky Firebase hostingu lze aplikaci optimalizovat pro zobrazení na smartphonech.

2.2.1 Struktura IMES

Aplikace se dá rozdělit na 3 hlavní části viz obrázek 2.4.

1) Software system

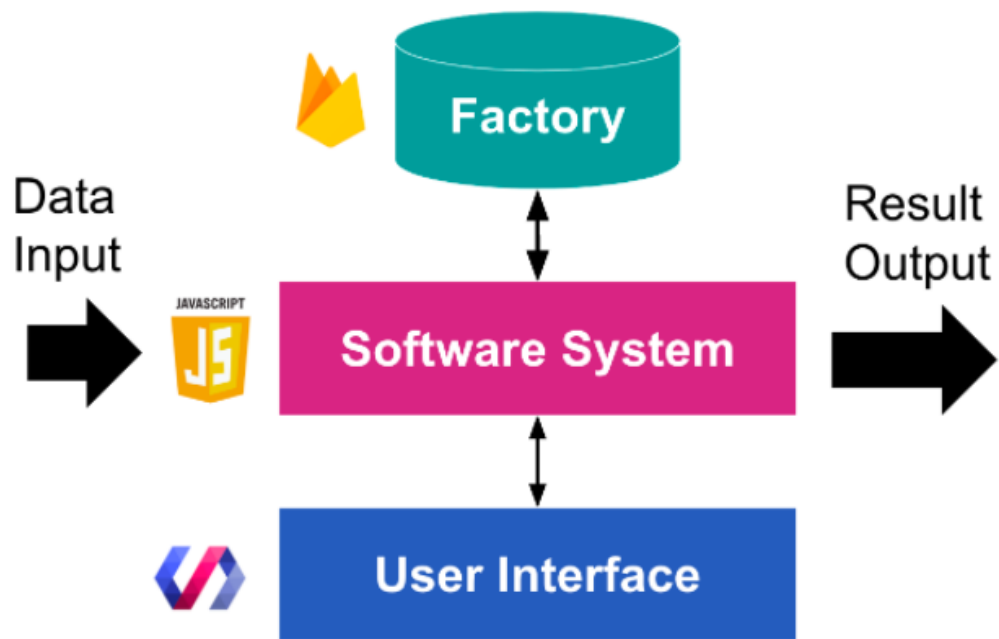
Tento blok je napsán výhradně v programovacím jazyku JavaScript. Pracuje tak, že si přebere vstupní data od uživatele (Data input), algoritmus je zpracuje a vyhodnotí a poté pomocí User Interface zobrazí zpracovaná data (Result output) uživateli. [14]

2) Factory

V bloku factory se nachází Firebase databáze celého systému. Využívá se tedy pro ukládání dat. Tento blok transformuje data získaná z předcházejícího bloku a ukládá je ve formátu JSON pro jejich další využití. [14]

3) User Interface (UI)

Tento blok slouží pro zobrazení výsledků uživateli. [14]



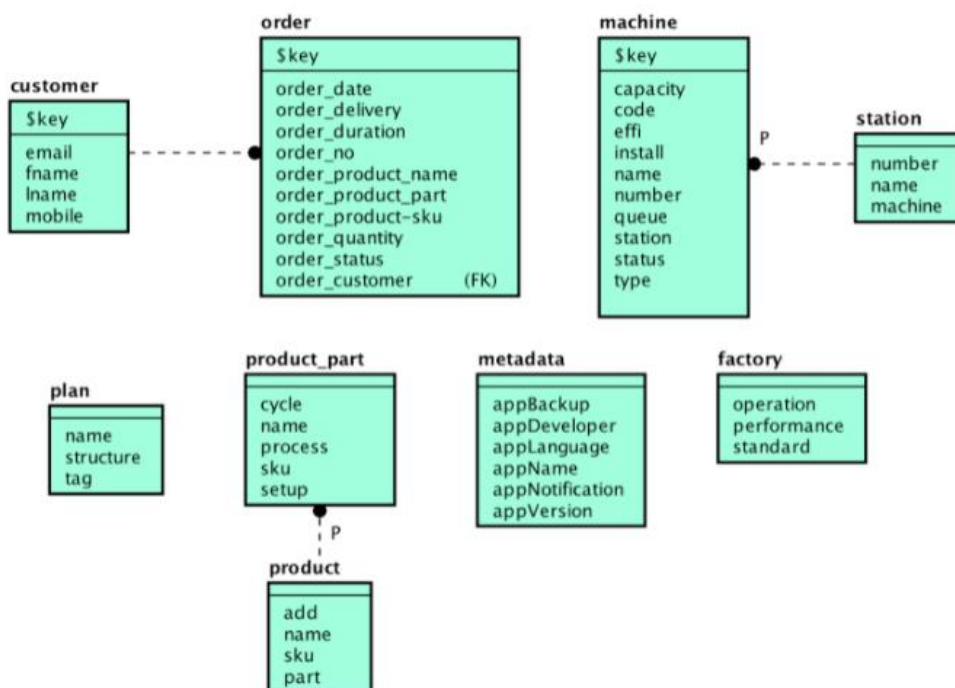
Obrázek 2.4 Struktura aplikace IMES [14]

2.2.2 Firebase databáze

Jak už bylo zmíněno IMES využívá Firebase Realtime databázi. Tuto databázi nabízí společnost Google zdarma, ovšem s omezením přenesených dat za jeden měsíc. V základu je to 10 GB stažených dat za měsíc a velikost Realtime úložiště pro 1 GB dat.[18]

S Firebase RT lze komunikovat i za pomocí protokolu Modbus TCP, který se hojně využívá v průmyslu např. při komunikaci s PLC.

Databáze v aplikaci IMES je zaintegrována hluboko v těle a nelze ji moc snadno modifikovat. Ovšem má zde určitý základ podle standardu ISA-95. Tento standard definuje rozhraní mezi řídicí a podnikovou vrstvou ve výrobním procesu. Schéma databáze lze vidět na obrázku 2.5.



Obrázek 2.5 Struktura databáze IMES [14]

2.2.3 Základní funkce IMES

IMES se vyznačuje poměrně jednoduchou a přehlednou strukturou. Na úvodní obrazovce se po přihlášení zobrazí jednoduchý přehled nejdůležitějších parametrů. Např. dokončené objednávky, počet zařízení v chodu, či průměrné OEE, což je celková efektivita zařízení.

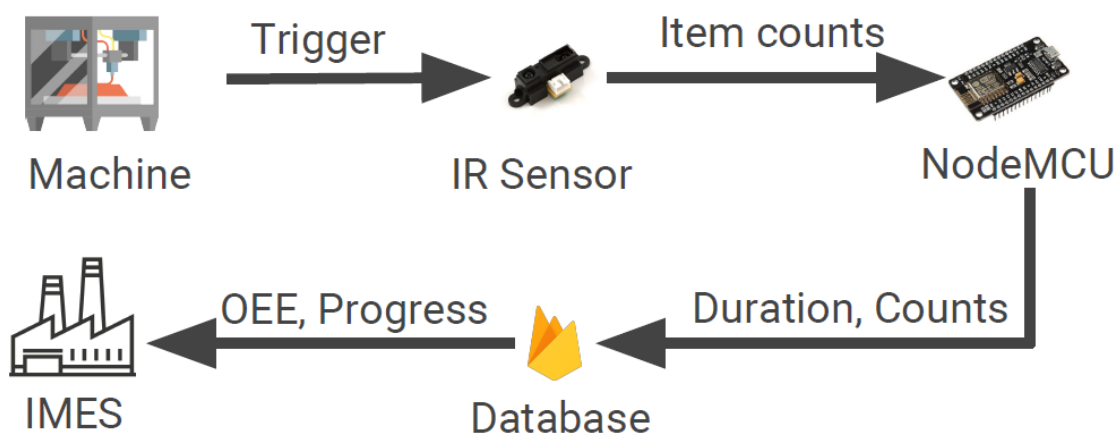
V modulu Setup můžeme nastavit základní parametry naší výroby, nastavení strojů, přidání zákazníka či položky do skladu.

V dalším modulu lze plánovat výrobu a přidávat objednávky podle zákazníků.

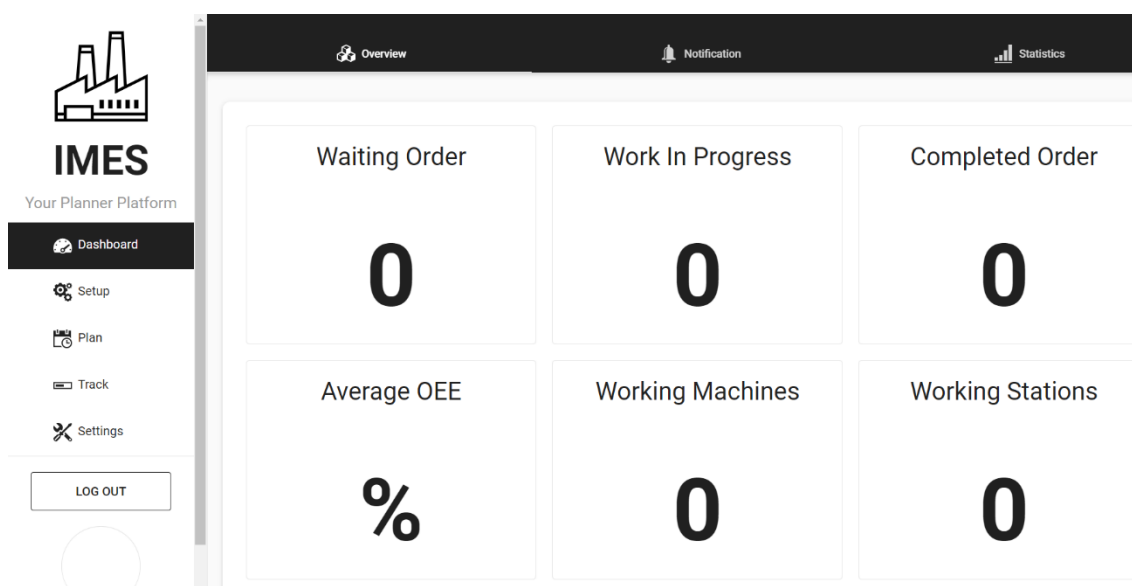
Track modul slouží k trasování výroby. Zobrazuje, v jaké fázi se zrovna výrobní proces nachází. V tomto modulu můžeme použít nějaké čidlo a monitorovat jím výrobu viz obrázek 2.6.

Poslední modul slouží pouze pro nastavení uživatele, případně k exportu dat do souboru JSON.

Všechny moduly a úvodní obrazovka lze vidět na obrázku 2.7.



Obrázek 2.6 Možnosti trasování výroby za pomoci externího senzoru [15]



Obrázek 2.7 Úvodní obrazovka IMES

2.2.4 Zprovoznění aplikace

Nejprve je nutné stáhnout zdrojové soubory z github serveru. Dále je zapotřebí nainstalovat nástroj NodeJS ve verzi 8.4 a vyšší. Poté otevřeme příkazový řádek ve složce s projektem a za pomoci uvedených příkazů nainstalujeme potřebné rozšíření. V dalších krocích jsou popsány příkazy pro instalaci těchto nástrojů. V této práci se pracovalo s verzemi:

- NodeJS verze 14.17.6
- Polymer CLI verze 1.9.11
- Firebase tool verze 9.23.022
- Npm verze 8.3.0

Yarn verze 1.22.17

- 1) Instalace poslední verze nástroje npm. Ta slouží pro instalace dalších rozšíření.

```
npm i npm -g
```

- 2) Instalace nástroje Yarn.

```
npm install -g yarn
```

- 3) Instalace rozšíření polymer-cli

```
Yarn add global polymer-cli
```

- 4) Přidání rozšíření bower.

```
Yarn add global bower
```

- 5) Instalace balíčků

```
npm install  
bower install
```

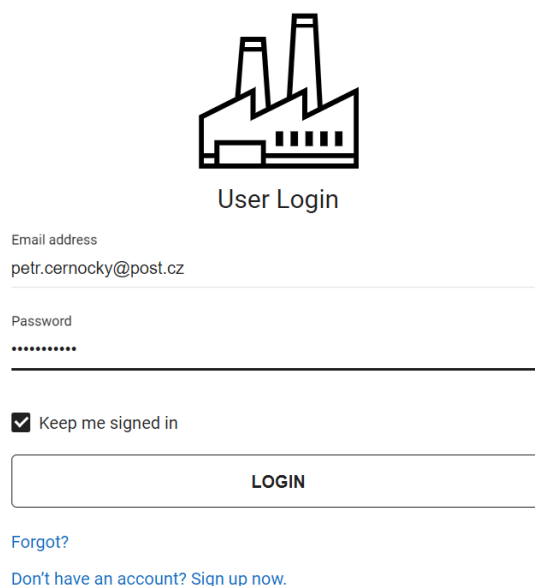
- 6) Celý projekt následně zkompilujeme za pomoci následujícího příkazu.

```
polymer build
```

- 7) Projekt zveřejníme jako webovou stránku na firebase hostingu níže uvedeným příkazem. Předtím je ovšem nutno změnit v souboru .firebaserc název projektu, který si založíme na stránkách firebase. Tento název je uveden v uvozovkách “navez_projektu“.

```
firebase deploy
```

Po posledním kroku stačí zadat do internetového prohlížeče vygenerovanou adresu aplikace. V tomto případě <https://smart-mes-11951.web.app/> a zadat uživatelské jméno, které bylo zvoleno na stránkách Firebase, nebo si vytvořit nový účet podle pokynů na přihlašovací obrazovce. V případě této práce stačí zadat jako uživatelské jméno e-mail petr.cernocky@post.cz a heslo `VutBrno2021`.



The image shows a user login interface for IMES. At the top center is a line-art icon of a factory with two smokestacks. Below the icon is the text "User Login". The form consists of several elements: an "Email address" field containing "petr.cernocky@post.cz", a "Password" field with masked characters ".....", a "Keep me signed in" checkbox which is checked, a "LOGIN" button, a "Forgot?" link, and a "Don't have an account? Sign up now." link.

Obrázek 2.8 Přihlašovací obrazovka IMES

2.2.5 Shrnutí

Aplikace IMES patří mezi další open source MES a pro svou jednoduchost nenáročnost by měla být vhodná pro malé a středně velké podniky, pro optimalizaci výrobních nákladů. Základem IMES je Firebase Realtime databáze. S touto databází lze komunikovat např. přes protokol Modbus. Jednou z výhod IMES je to, že díky Firebase hostingu je aplikace optimalizována i pro použití na smartphonech, čímž může dost urychlit a usnadnit zapisování informací do MES přímo ve výrobě. Oproti Qcadoo má tu výhodu, že aplikace prakticky není závislá na operačním systému řídicího PC, ale pouze jen na verzi internetového prohlížeče, Aplikace je uzpůsobena jak pro sledování výroby za pomoci externího čidla, tak pro ruční zapisování údajů.

3. NÁSTROJE PRO VYTVÁŘENÍ VIRTUÁLNÍCH MODELŮ

Na trhu se nachází nepřehledné množství různých nástrojů pro vytváření virtuálních modelů. V této kapitole se zaměříme na nástroje od firem ABB a Siemens. Tyto nástroje nejsou bohužel volně stažitelné. Programy pro virtuální zprovoznění a linky vytvořené v nich mají v průmyslu tu výhodu, že si na dané lince můžeme odsimulovat nějaký výrobní proces, případně simulaci zrychlit a díky tomu odhalit různé scénáře pro různé chyby bez toho, aniž bychom narušili fyzickou výrobu. Můžeme díky nim těmto poruchám předcházet a zamezit tak ztrátám jak finančním či na zdraví obsluhy zařízení.

3.1 ABB Robot studio

Tento software vytvořila společnost ABB. Obsahuje soubor nástrojů pro uživatele průmyslových robotů. V této aplikaci lze snad vytvářet kompletní simulace ve 3D prostředí. Tento nástroj určitě ocení firmy, které již vlastní nějaké vybavení od společnosti ABB. Jelikož Robot studio obsahuje knihovny s roboty své výroby již v základu, vedlo by to tedy k usnadnění práce při vytváření digitálního dvojčete. Nejdůležitější nástroje tohoto SW jsou popsány níže.

3.1.1 Modelovací nástroje ABB Robot studio [16]

ABB Robot studio nabízí celou škálu možností modelování:

- **3D import a modelování**
Umožňuje import dat z různých CAD formátů, K dispozici jsou funkce pro nastavení layoutu. Ty tam umožňují umístit jednotlivé části linky přesně tam, kam chceme. Můžeme zde samozřejmě modelovat i vlastní dopravníky, nástroje či chapadla.
- **Visual SafeMove**
3D nástroj, který umožňuje nastavit bezpečnostní zóny např. kudy se pohybuje robotické rameno a zamezit tak kolizím s jinými zařízeními či s obsluhou.
- **Visualization**
Pokročilejší grafický nástroj. Umožňuje práci se světly. Díky tomu můžeme docílit realističtější vizualizace.
- **Station Viewer**
Umí vytvářet záznamy simulací ve 3D formátu, ve kterém můžeme procházet výrobu. Simulaci lze zrychlit či zpomalit podle aktuální potřeby.
- **Editace a ladění programu**
V Robo Studiu máme umožněno přímé programování námi navržených robotů. Program si můžeme odladit na 3D vizualizaci bez toho, aniž bychom nějak ohrozili výrobu.

- **Komunikace**

Robot studio nám umožňuje komunikaci s komponentami přes OPC UA. Tím pádem nám umožňuje naplnit požadavky pro virtuální dvojče v průmyslu 4.0.

3.2 Siemens NX Mechatronics Concept Designer

Další software, který lze využít k modelování vyvinula společnost Siemens. Je součástí nástroje siemens NX. V tomto programu lze vyvíjet téměř jakékoliv zařízení. Lze jít od základního vymodelování 3D modelu zařízení, přes skládání 3D sestav až po fyzikální modelování celé sestavy. Oproti ABB Robot Studio je tento software poměrně všeobecného užití. Obsahuje ovšem také knihovny s modely a funguje zde import většiny CAD souborů. V MCD umožňuje nejen náhled na zařízení, ale i fyzikální ověření funkčnosti. [17]

Díky tomu, že lze posílat signály do MCD přes OPC UA, můžeme model ovládat externě např. přes PLC. Těto vlastnosti můžeme opět s výhodou využít pro odladění reálného modelu. NX nám taktéž obstará diagnostiku namáhání některých částí zařízení. Tím můžeme odhalit možné selhání součástky a předejít tak ztrátám ve výrobě. [17]

Tím, že NX funguje tak, že na sebe postupně navazují jednotlivé nástroje, můžeme určitým způsobem urychlit vývoj jednotlivých komponent a případně odhalit jejich problémy již za návrhovým stolem.

3.3 Realizace Simulace

Pro další práci byl vybrán nástroj ABB Robot Studio ve verzi 21.4.9700.0, ve kterém bylo využito jedné z dostupných demonstračních linek. Ta se následně lehce poupravila a doplnila o nové komponenty.

3.3.1 Výhody a nevýhody ABB Robot Studia

Jedním z důvodů, proč byl zvolen tento nástroj pro realizaci je, že obsahuje celou škálu knihoven pro snadnou realizaci modelu linky. Mezi další výhodou patří jednoduché vytváření OPC Serveru. Každý kontrolér robotu vytváří vlastní OPC Server. V uvedené simulaci jsou využity kontroléry typu IRB120. Na tyto servery se připojujeme jako OPC Klienti a čteme dostupná data. Tyto informace jsou posílány dále do nadřazené MES aplikace, která je zpracovává a vydává další povely.

Jako výhodou lze určitě uvést i to, že lze do simulace přímo přivádět signály přes OPC Klienta např. z řídicího PLC. Tuto funkci ovšem v této implementaci nevyužívám.

Z nevýhod můžeme uvést jednu poněkud méně praktickou vlastnost a to tu, že OPC Server lze vytvořit pouze pro kontroléry, nikoliv však pro jednotlivé smart komponenty. Proto je nutné tyto signály vytvořit uvnitř kontroléru a z něj je následně spojit v sekci „Logika Stanice“.

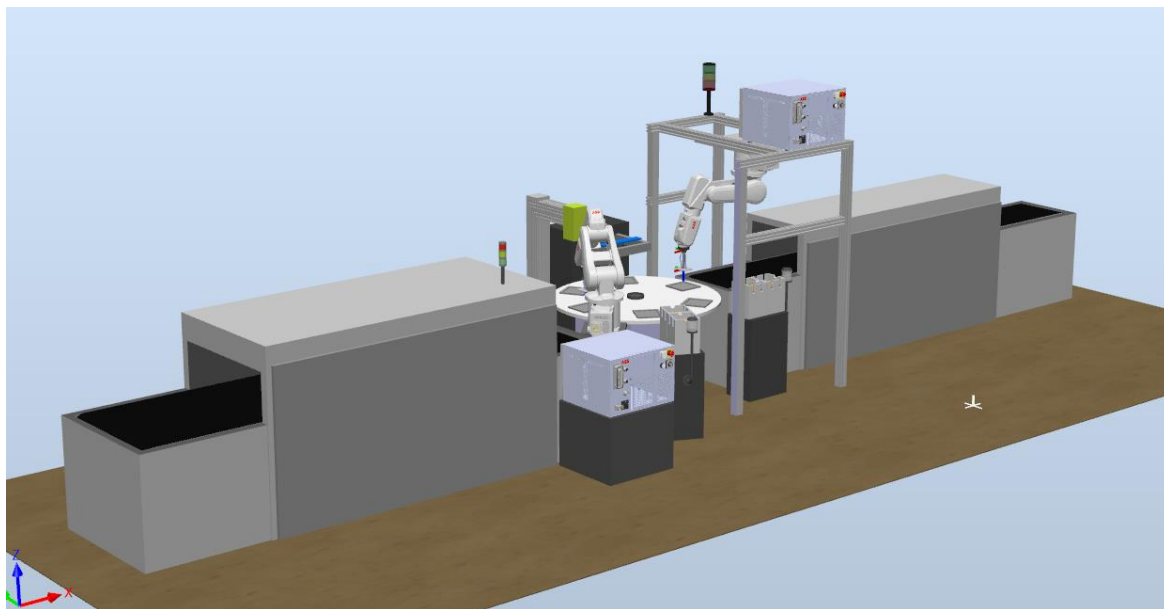
3.3.2 Demonstrační linka

Jak už bylo výše zmíněno, byla využita jedna z demo linek, kterou dodává výrobce společně se softwarem. Tato sestava má název „Demo Solar Simulation“. Jedná se o linku na výrobu (kontrolu) solárních panelů.

Do této linky byl zaimplementován spouštěč simulace – jedná se o generátor, který nám generuje nové kusy na pás. Z něj jsou pak přemísťovány pomocí dvou robotických ramen. Nejprve na jedno překladiště, kde se kamerou „zkontroluje“ kvalita a na základě výsledku se následně druhým ramenem přemístí na určené místo.

Dále byla linka opatřena dvěma senzory, které pomáhají monitorovat průběh výroby. Jeden reaguje na umístění výrobku na pokračující dopravníkový pás a druhý zaznamenává vadné kusy, které umísťuje rameno do odkládací plochy. Signály z těchto čidel jsou poté namapovány přes OPC Klienta do databáze aplikace.

Pro monitoring procesu a stavů linky byl zde ještě přidán majáček, který signalizuje aktuální stav výroby. Tato smart komponenta byla získána z volně dostupných knihoven, nacházejících se na internetu. Dále zde bylo využito standardu PackML, který definuje, jaké chybové stavy mohou nastat. Použitím tohoto standardu zjednoduší pozdější integraci i k jinému MES. Bližší informace jsou uvedeny v podkapitole 3.3.3 Logika stanice.



Obrázek 3.1 Ukázka demonstrační linky

3.3.3 Logika Stanice

Celý náskres logiky stanice je uveden v příloze B této práce. Do logiky stanice bylo nutné zaintegrovat nová smart zařízení. První z těchto zařízení jsou čidla, jejichž výstupy (True-false) signalizují přítomnost objektu na pásu a na odkládací ploše.

Jako spouštěč celé simulace byl využit generátor nových výrobků solárních panelů na pás. Po spuštění tohoto generátoru je už celý postup automatizován. Při běhu programu svítí majáček u vrcholu stanice zeleně.

Pro simulaci stavu „Suspended“ byla zavedena proměnná `Pause_simulation` touto proměnou je zastavována linka tak, jako by se to stalo při výrobě. Dojde k zastavení pohybu pásů, a i robotů. V případě „Suspended“ stavu se rozsvítí majáček žlutě.

Poslední ze stavů dle PackML standardu, který budeme simulovat je chybový stop stav. Tento stav taktéž zastaví simulaci, ovšem pro opětovné zprovoznění by mělo dojít k úplnému restartu systému. Bohužel pro tuto výjimku není v RobotStudios žádná vhodná funkce, proto se tento stav v našem případě chová stejně jako „Suspended“ s tím rozdílem, že tuto výjimku signalizuje majáček červenou barvou.


Všechny tyto změny a propojení byly provedeny v sekci „Logika stanice“, kde lze všechny signály dostupných zařízení pohodlně propojit, a přitom využít i dostupných logických členů. V této sekci se nachází objekty kontrolérů s proměnnými, do kterých se budou zapisovat a číst přes OPC Server.

Na obrázku B.1 uvedeném v příloze této práce, můžeme vidět přidané smart komponenty a propojení mezi nimi. Jedná se o komponenty `LineSensor`, `PlaneSensor` a `SC_StatusLamps`. Celá logika stanice je velmi jednoduchá a přehledná. Propojování jednotlivých signálů lze provádět jednoduše myší. Komponenty s označením `IRB120_ROBx` představují kontroléry, ze kterých jsou pak čtena a zapisována data přes OPC Server.

3.3.4 Tvorba OPC Serveru

OPC server se zde vytváří za pomoci rozšíření ABB Robotics IRC5 OPCUA Config tool. V tomto nástroji je nutno vybrat ty kontroléry, které podporují komunikaci pře OPC UA a připojit se k nim tlačítkem „Create“ viz obrázek.



Add New Alias

 Alias Name:

Connection Criteria

| | | | |
|--|---|--|--|
| <input checked="" type="checkbox"/> Controller Name: | <input type="text" value="DESKTOP-3GJP9T7"/> | <input checked="" type="checkbox"/> System Name: | <input type="text" value="IRB120_ROB2"/> |
| <input checked="" type="checkbox"/> Address: | <input type="text" value="stems\IRB120_ROB2"/> | <input type="checkbox"/> Controller ID: | <input type="text"/> |
| <input checked="" type="checkbox"/> System ID: | <input type="text" value="{46915BD6-71A7-4256-9F4D-C4519F8475E1}"/> | | |

Scan results: 2 found. 2 of 2 displayed.

| Controller Name | System Name | Address | Controller ID | System ID |
|---|-------------|----------------------------|---------------|--------------------------|
|  DESKTOP-3GJP9T7 | IRB120_ROB1 | C:\Users\petrc\Document... | | {6A71777D-D1DF-42D7-... |
|  DESKTOP-3GJP9T7 | IRB120_ROB2 | C:\Users\petrc\Document... | | {46915BD6-71A7-4256-9... |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Show only robots with no assigned Alias
 Show only robots that match connection criteria

Obrázek 3.2 Připojení k vybranému kontroléru

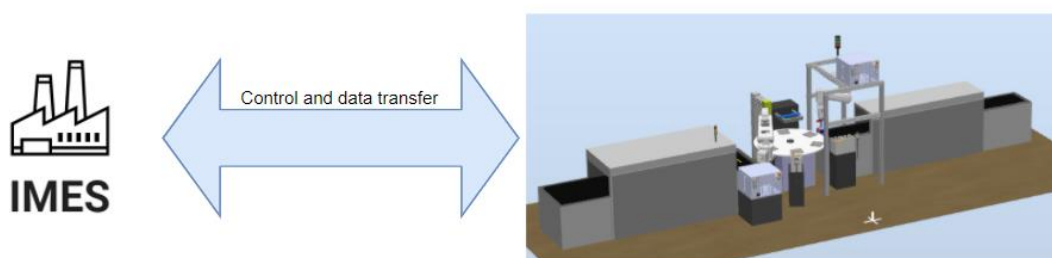
V tomto nástroji lze provádět další nastavení OPC serveru, jako např. správu klientských certifikátů, či nastavení portů. Nachází se zde i seznam událostí na serveru.

4. ŘÍZENÍ VÝROBY ZA POMOCÍ MES BEZ VYUŽITÍ AAS

V této kapitole je rozebráno klasické řízení výroby s využitím MES. Pro tuto implementaci byla zvolena opensource aplikace IMES, její možnosti a výhody byly rozebrány o kapitoly výše.

4.1 Charakteristické znaky

Toto řízení se vyznačuje klasickým pyramidovým uspořádáním, kde nadřazený řídicí systém komunikuje s nižšími úrovněmi řízení výroby např. s operátorskou úrovní či, jak v našem případě, přímo s řídicími buňkami (kontroléry robotů). MES získává data z výroby a na základě těchto dat úkoluje výrobní linku.



Obrázek 4.1 Ukázka přenosu dat mezi MES a virtuální linkou

4.2 IMES

Pro tento demonstrační systém byla zvolena aplikace IMES, tu ovšem bylo nutné mírně modifikovat. Aplikace byla navržena na firebase hosting, který je neustále aktualizován. Dostupný zdrojový kód byl však již několik let zastaralý a po zkompilování jej nebylo možné správně spárovat s databází. Proto bylo potřeba mírně změnit syntaxi přístupových cest k databázi aplikace.

Pro připojení databáze k aplikaci bylo nutné zadat unikátní údaje, které se nám vygenerují po založení projektu na stránkách Firebase. Tyto údaje jsme poté dopsali do příslušných řádků v souboru s názvem view-app.html.

```
<firebase-app
name="smart-mes"
auth-domain="smart-mes-11951.firebaseio.com"
database-url="https://smart-mes-11951-default-rtdb.firebaseio.com"
api-key="████████████████████████████████████████"
storage-bucket="smart-mes-11951.appspot.com"
messaging-sender-id="██████████████████"
project-id="smart-mes-11951">
</firebase-app>
```

Po prvotních problémech s připojením do databáze a nastavením firebase klienta se nakonec podařilo správně propojit aplikaci s databází.

Pro přístup do aplikace je nutné uživatele autentizovat jménem a heslem. Po autentizaci se uživateli zobrazí úvodní stránka se základními informacemi o výrobě viz obrázek 2.6.

4.3 Popis ovládání a modifikací IMES

V této podkapitole jsou podrobně popsány způsoby ovládání aplikace a dále provedené změny, které byly provedeny. Je zde zdokumentován postup od nastavení celého systému (továrny) po dokončení objednávky.

4.3.1 Nastavení továrny

Tato nastavení se provádí v sekci „setup“. Nachází se zde základní možnosti, inicializace továrny, strojů, stanice, produktu, zákazníka a inventáře.

V sekci pro informací o továrně je možnost zvolit jméno továrny (podniku), a i časy jednotlivých směn. Lze zde zvolit i produkční model, tj. sériovou, paralelní nebo multikomponentní výrobu. Dle těchto zvolených možností pak probíhá plánování výroby a zařazování objednávek podle jejich priority. Je zde předpřipravena i možnost pro další optimalizaci výroby, tahle možnost ovšem zatím není implementována.

Další podmodul „Machine“ slouží pro editaci strojů. V této sekci lze zadat informace o strojích v továrně. Dále lze nastavit maximální kapacitu zařízení a jeho typ.

V části „Station“ nastavujeme možnosti celé výroby. Do naší stanice volíme stroje, které byly nainicializovány v předcházející sekci.

Pro přidávání produktů, které je výroba schopna vytvářet slouží modul „Product“. Zde lze nastavit výrobky, které se budou vyrábět. Nachází se zde i možnost nastavení sestavy, tzn. že se námi vyráběný produkt sestává z různých dílů. Kromě zmíněných možností zde dochází k zadávání doby výroby, či časů pro nastavení pro výroby.

V předposlední části ukládáme možné zákazníky. V této sekci je možnost pro vyplnění základních informací, jako jsou jméno, e-mail a telefon.

Poslední modul slouží pro zadávání inventáře součástí, ze kterých se skládají jednotlivé výrobky. Zadáváme, jak počet, tak i cenu jednotlivých položek. Tyto informace jsou pak dále využity při výpočtu cen objednávek.

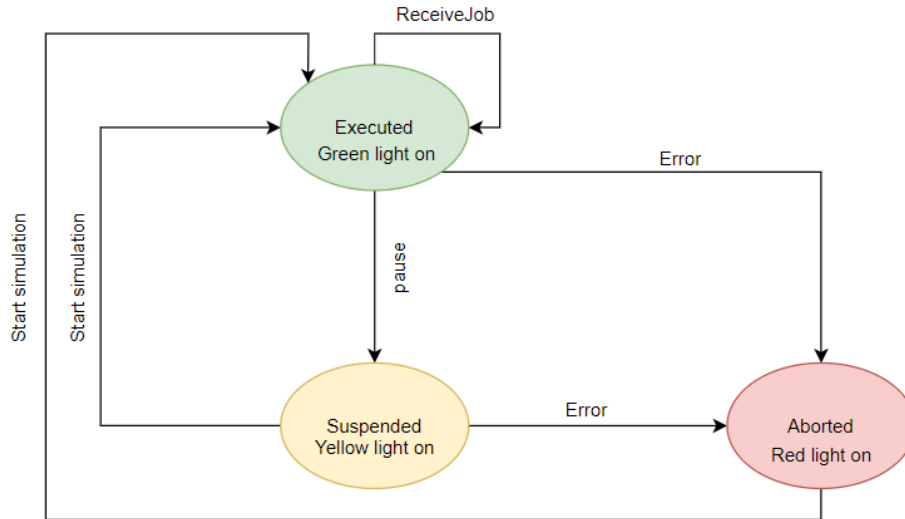
4.3.2 Plánování výroby

Podmodul „Plan“ slouží jak pro zadání objednávky, tak i pro plánování a rozvrh výroby jako takové.

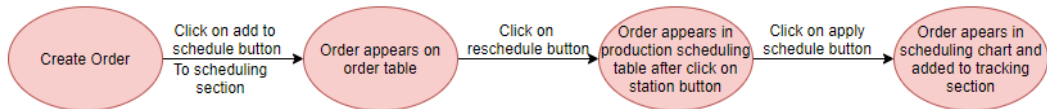
Byly zde provedeny úpravy, které byly využity pro simulaci. V sekci „Production“ byly přidány tlačítka, která generují výjimky a stavy výroby. Tyto tlačítka mají za úkol simulovat výrobu, kdy se může stát, že stroje vyvolají chybu, či jiný poruchový stav. Pro univerzálnost užití našeho řešení, bylo využito standardu PackML, který určuje, jakých

stavů může stanice nabývat. Tento diagram byl brán jako inspirace pro značení stavů stanice dle obr 4.2. Bylo využito jen základní části – kostry tohoto diagramu. V případě změny stavu se tyto změny zobrazí v notifikacích. Tato funkce rovněž chyběla, proto byla přidána.

Příklad pro zadání objednávky je uveden na blokovém schématu na obr. 4.3.



Obrázek 4.2 Stavby stanice



Obrázek 4.3 Vývojový diagram pro vytvoření a zavedení objednávky do výroby

Při zadávání objednávky je nutno vyplnit jednotlivé požadavky jako je počet kusů a čas doručení. Objednávku lze i vytisknout za pomoci tlačítka „Print Order“. V tabulce objednávek „Order Table“ je zobrazen průběh jednotlivých objednávek, a to v jakém stavu se momentálně nacházejí. Objednávky jsou zde řazeny nejen dle priority, ale i podle toho, jaký produkční model jsme si zvolili v nastavení továrny. Podle těchto modelů jsou zařazovány i do časového rozvrhu.

4.3.3 Trasování výroby

Jako další úprava aplikace byla provedena v části pro trasování výroby. Jednou z úprav bylo zprovoznění funkce pro přidávání a editaci senzorů. Tato funkce v dostupném zdrojovém kódu chyběla.

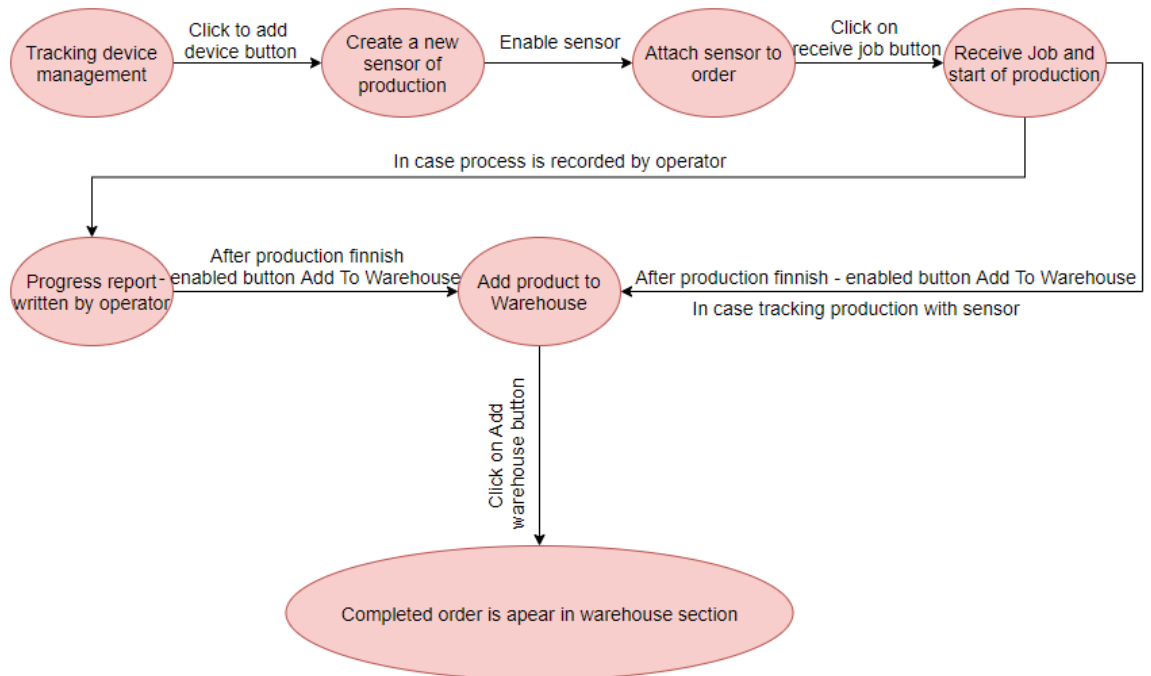
Další úprava v této sekci byla přidání tlačítka „ADD TO WAREHOUSE“, jelikož automatické ukládání, které navrhl tvůrce aplikace nefungovalo dle našich předpokladů. K aktivaci tohoto tlačítka dojde, jakmile se splní podmínka a je dosažen počet

vyhovujících kusů. Pro zmetky je zde taktéž kolonka, do ní lze zapisovat ručně. Pro demonstrační výrobní linku jsme upravili i možnost zaznamenávání počtu zmetků za pomocí čidla.

Z informací o zmetkovitosti se dále vypočítává parametr výkonosti OEE. Tento výpočet se provádí v sekci „Performance“, kde po stisknutí tlačítka „Calculate OEE“ dojde k výpočtu pro poslední vykonanou objednávku. Parametr OEE je vypočten dle vzorce 1.4, kde za *Dostupnost* je dosazen skutečný čas výroby podělený plánovaným časem výroby, dále za *Kvalitu* je dosazen počet dobrých kusů podělených celkovým počtem vyrobených kusů a za *Výkon* je dosazen ideální délka cyklu podělená skutečnou délkou cyklu. Pro tento výpočet byla napsána funkce „CalculateOEE“ ve zdrojovém souboru view-track-performance.html.

Jako jedna z posledních byla provedena úprava pro sledování aktuální výroby. Jelikož pro sledování aktuálního stavu objednávky bylo nutné obnovovat webovou stránku, na které aplikace funguje, bylo opět nutné modifikovat zdrojový kód stránky, aby tyto změny byly sledovatelné v reálném čase.

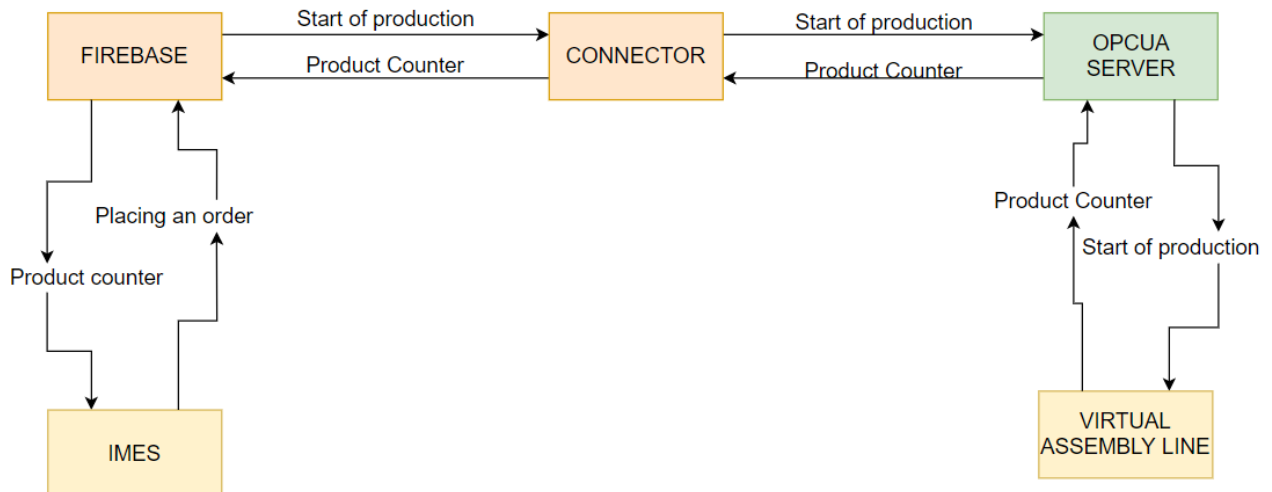
Příklad nastavení trasování výroby je uveden na blokovém schématu níže.



Obrázek 4.4 Vývojový diagram pro trasování výroby

4.4 Komunikační blokové schéma řízení

Blokové schéma znázorňuje komunikaci s jednotlivými komponentami zapojenými do procesu. Jako konektor slouží nástroj NODE-RED, díky kterému lze snadno přistupovat k datům ve firebase databázi a zároveň lze číst či zapisovat data jako OPC Klient do serveru. Ve schématu jsou uvedeny i předávané signály – data, která nám spouští další operace.



Obrázek 4.5 Blokové schéma „klasického“ řízení výroby

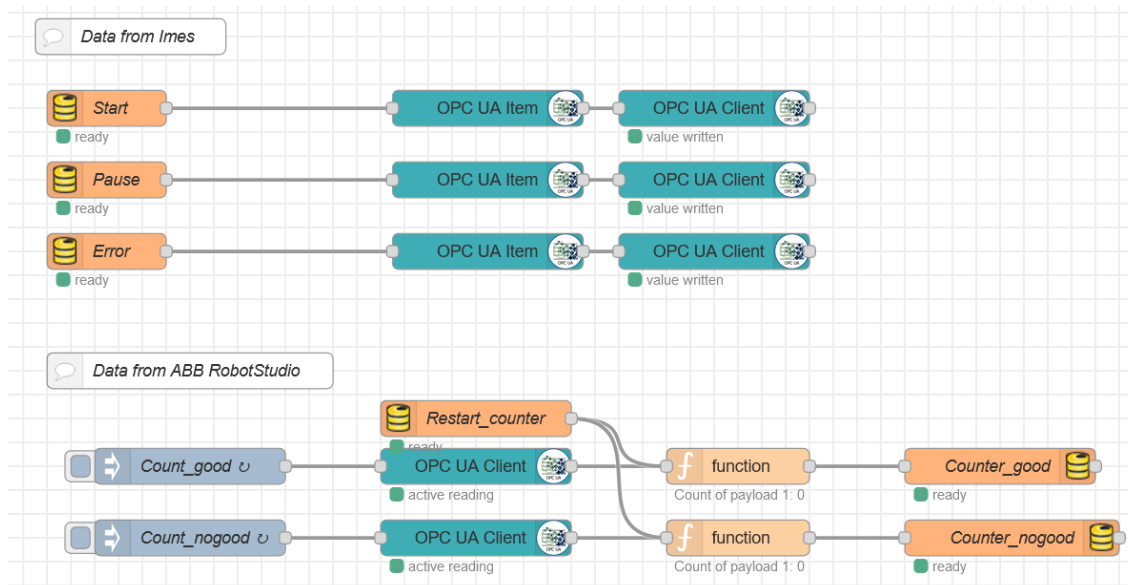
4.4.1 Popis komunikace

V IMES se vytvoří objednávka na zboží. Poté se zadá do pracovního rozvrhu. V tomto kroku se přesunou informace o zpracování do „Tracking section“, kde se tlačítkem „Receive job“ zařadí objednávka do výroby a v tomto případě se celá linka uvede do provozu. Pro tuto akci jsme do databáze přidali pomocnou proměnou „Start“ v sekci WorkStation, která se změní, jakmile přijmeme objednávku. V reálném provozu bychom tuto akci musel ještě nejspíše potvrdit pracovník – vedoucí výroby. Poté dojde k zahájení výroby.

Z linky jsou přes OPC server posílána data o průběhu výroby. Např. za pomoci snímače, případně čítače. Tento snímač přiřadíme k objednávce v sekci pro sledování průběhu výroby. Aplikace hlídá počet vyrobených kusů s požadovaným počtem. Tohle vše probíhá reálném čase. Jakmile dojde ke splnění objednávky aplikace změní stav proměnné „Start“ na false a dojde k zastavení výroby. Pro úplné odsouhlasení dokončené objednávky je nutné kliknout na tlačítko „ADD to Warehouse“ a tím dojde k přemístění vyrobených kusů do databáze skladiště. Tuto akci by teoreticky mohl vykonávat pracovník skladiště, který vyrobené kusy převezme k naskladnění.

4.4.2 Propojení za pomoci nástroje NODE-RED

Pro počítání počtu vyrobených kusů byla napsána jednoduchá funkce, která počítá impulsy na základě přijaté zprávy. Stejným principem funguje „Restart_counter“. Tato proměnná se změní, jakmile stiskneme v aplikaci tlačítko „Add to Warehouse“. Zobrazené zapojení ukazuje možnosti spojení databáze firebase s OPC serverem.



Obrázek 4.6 Ukázka spojení databáze s RobotStudiem za pomoci nástroje NodeRed

4.5 Výhody a nevýhody

Mezi zjevnou nevýhodou je nutno uvést poměrně složité mapování výstupů ze simulace ke vstupům databáze. Pokud bychom přidali další senzor pro další výrobu, bylo by nutné toto schéma předělat. Stejný problém lze předpokládat v tom případě, kdyby bylo nutné tuto aplikaci připojit k nadřazenějšímu ERP systému. Na trhu se momentálně nevyskytuje žádná aplikace, která využívá tento typ databáze.

Pro zjednodušení propojení s databází byl využit nástroj NodeRed, který obsahuje knihovny pro vyčítání dat z Firebase databáze. Pomocí jednotlivých bloků bylo vytvořeno komunikační schéma.

5. ŘÍZENÍ VÝROBY S VYUŽITÍM AAS

V této kapitole jsou rozebrány možnosti řízení výroby s využitím AAS. Pro vytvoření AAS našeho MES systému jsme využili nástroj AASX Package Explorer, který vytvořili vývojáři ze společnosti FESTO. Tento nástroj je opět volně stažitelný na internetu.

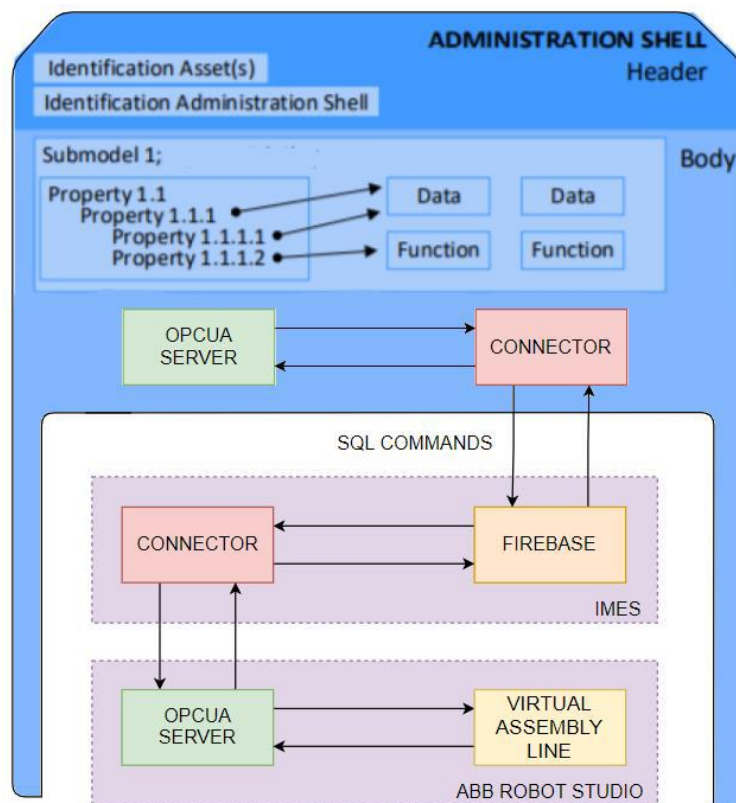
5.1 Charakteristika tohoto druhu řízení

Při řízení s využitím AAS je nutné vytvořit jakousi obálku MES systému společně s virtuální linkou. S daty, které jsou „obaleny“ komunikuje hlavička AAS za pomoci SQL příkazů. Dále už komunikace probíhá pouze s hlavičkou AAS.

V hlavičce byly vytvořeny jednotlivé submodely dle standardu ISA-95 pro předávání informací mezi MES a ERP systémy. V této práci ovšem nebylo využito všech modelů, které tento standard umožňuje. V aplikaci IMES by to nedávalo smysl, jelikož není pro práci s těmito daty připravena.

5.2 Blokové schéma

Níže uvedené blokové schéma znázorňuje možnou strukturu ovládání.



Obrázek 5.1 Blokové schéma řízení výroby za pomoci AAS (část obrázku použita z [5])

5.3 Vytváření AAS pro MES

V této podkapitole je vysvětlena tvorba AAS pro náš MES. Je zde uvedeno, jaké submodely byly použity a na základě čeho jsme je zvolili.

5.3.1 Použitý nástroj

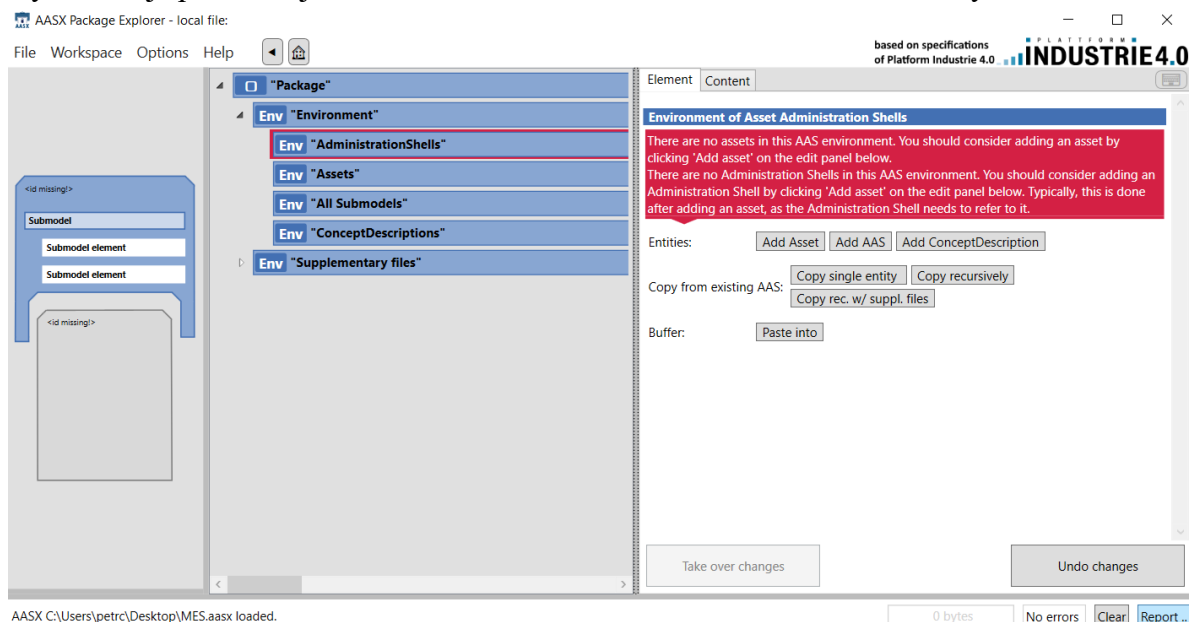
Pro práci s AAS byl využit nástroj AASX PackageExplorer. Tento nástroj slouží jak k prohlížení již hotových AAS (např. pro různá zařízení), tak v něm lze i sestavit vlastní AAS. Další výhodnou vlastností tohoto programu je, že umí z prohlíženého AAS vytvořit OPC UA server. Není to však jediný podporovaný komunikační standard, který podporuje. Jako další umožňuje vytvoření např. MQTT serveru.

Tento nástroj taktéž umožňuje exportovat .aasx formát do jiných formátů, jako jsou např. JSON soubory aj..

5.3.2 Vytváření AAS a jeho submodelů

Jak už bylo zmíněno, pro jednotlivé submodely byly inspirovány standardem ISA-95. Jsou tedy definovány hlavně ty pro komunikaci s vyššími vrstvami řízení.

Vytvoření je poměrně jednoduché a intuitivní viz obrázek hlavní obrazovky.



Obrázek 5.2 Hlavní obrazovka AASX PackageExploreru

První je nutno přidat AAS. To se provede jednoduše kliknutím na položku „Add AAS“. Dále je potřeba přidat Asset. Jakmile je vytvořeno AAS, tak lze začít tvořit jednotlivé submodely a jejich vlastnosti.

Ve finále bylo vytvořeno devět submodelů, které popisují výrobní stavy. Tyto submodely byly vybrány, protože tyto vlastnosti obsahuje i databáze aplikace IMES tzn., že je lze propojit.

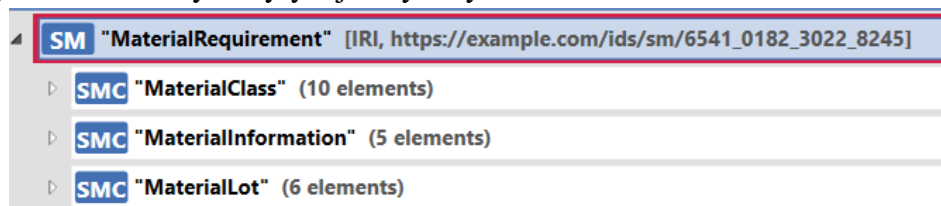
5.4 Výsledné submodely

Celkem tedy bylo vytvořeno 9 submodelů dle modelu pro výměnu dat. V této podkapitole je postupně rozebráno k čemu slouží. Každý ze submodelů taktéž vlastní svůj popis ve aasx souboru.

5.4.1 Submodel Material Requirement

Tento submodel definuje požadavky na materiál, který má být vyroben. Měl by zahrnovat celkové množství materiálu a jeho jednotku. Dále by zde měli být specifikovány další vlastnosti, jako např. jestli má být materiál spotřebován, či vyráběn.

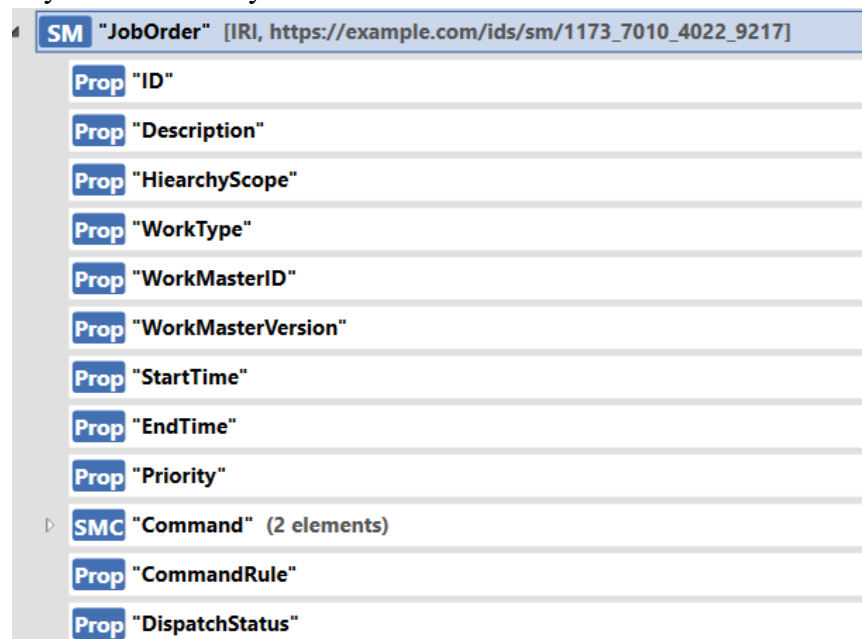
Tento submodel obsahuje další submodely a v nich se nacházejí vlastnosti, do kterých zapisujeme hodnoty. Tento první byl vytvořen velmi dopodrobna, jako ukázka komplexnosti standardu ISA95. Další vytvořené submodely již tak podrobné nejsou, jelikož tyto vlastnosti by nebyly nijak využity.



Obrázek 5.3 Submodel MaterialRequirement

5.4.2 Submodel Job Order

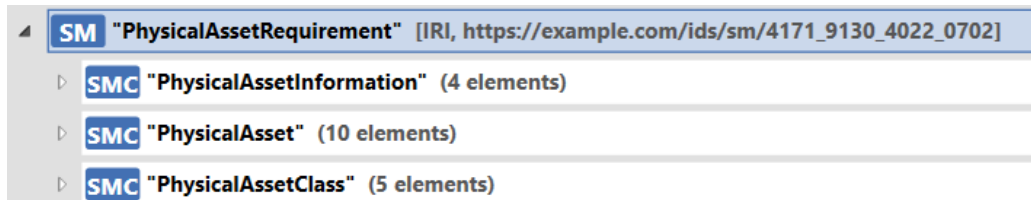
Definuje úlohu, co má být provedena. Přiřazuje zaměstnance, stroje, či požadavky na materiál, pro vykonání zakázky.



Obrázek 5.4 Submodel Job Order

5.4.3 Submodel Physical Asset Requirement

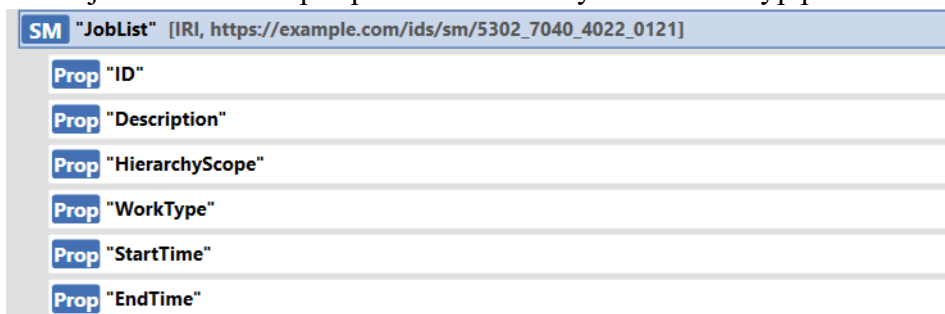
Tento submodel nese další možné informace, o fyzickém výrobku např. kdo nebo jaký stroj jej vyrobil apod.



Obrázek 5.5 Submodel Physical Asset Requirement

5.4.4 Submodel Job List

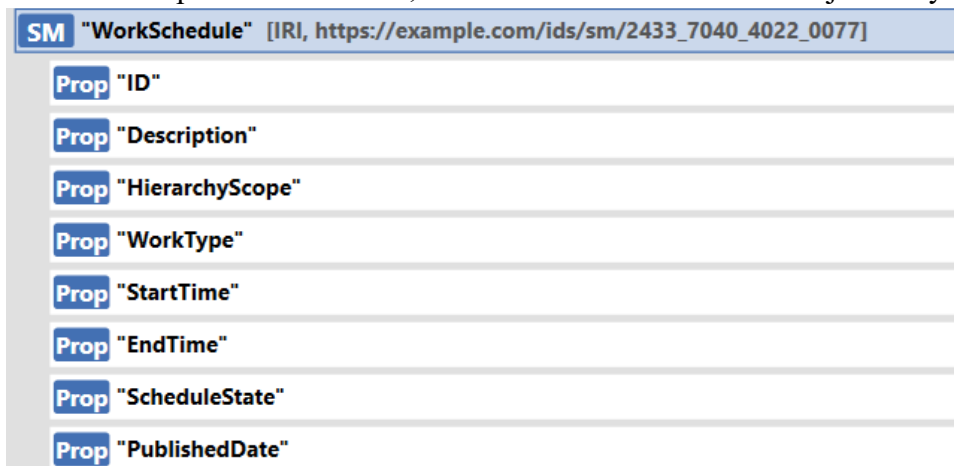
Job List definuje konkrétní čas pro provádění zakázky a konkrétní typ prováděné akce.



Obrázek 5.6 Submodel Job List

5.4.5 Submodel Work Schedule

Jak už z názvu vypovídá jedná se o submodel pro harmonogram prací. Skládá se jak z času začátku a konce pracovního úkonu, tak i další informace o stavu objednávky.



Obrázek 5.7 Submodel Work Schedule

5.4.6 Submodel Work Request

Tento submodel definuje požadavky na pracovní úkoly. Moc se neliší od předcházejících a prakticky je doplněn pouze o vlastnosti pro typ práce a prioritu.

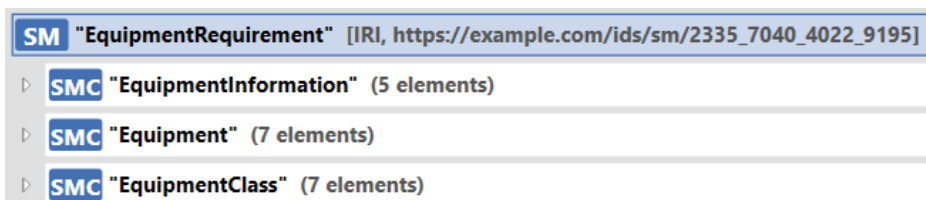


| |
|---|
| SM "WorkRequest" [IRI, https://example.com/ids/sm/3234_7040_4022_7394] |
| Prop "ID" |
| Prop "Description" |
| Prop "HierarchyScope" |
| Prop "WorkType" |
| Prop "StartTime" |
| Prop "EndTime" |
| Prop "Priority" |

Obrázek 5.8 Submodel Work Request

5.4.7 Submodel Equipment Requirement

Další submodel definuje, jaké nástroje a vybavení bychom měli při zhotovení zakázky používat. Nese s sebou samozřejmě informace o daném nástroji, např. kde se nachází apod.



| |
|--|
| SM "EquipmentRequirement" [IRI, https://example.com/ids/sm/2335_7040_4022_9195] |
| ▷ SMC "EquipmentInformation" (5 elements) |
| ▷ SMC "Equipment" (7 elements) |
| ▷ SMC "EquipmentClass" (7 elements) |

Obrázek 5.9 Submodel Equipment Requirement

5.4.8 Submodel Work Response

Předposlední submodel definuje pracovní odpovědi. Tyto odpovědi můžou zahrnovat jak typ práce, tak i její začátek a konec.

| SM | "WorkResponse" [IRI, https://example.com/ids/sm/4413_1230_5022_6974] |
|------|--|
| Prop | "ID" |
| Prop | "Description" |
| Prop | "HierarchyScope" |
| Prop | "WorkType" |
| Prop | "WorkRequestID" |
| Prop | "StartTime" |
| Prop | "EndTime" |
| Prop | "ResponseState" |
| Prop | "JobResponse" |

Obrázek 5.10 Submodel Work Response

5.4.9 Submodel Job Response

Poslední submodel představuje odpovědi a informace o provedení objednávky.

| SM | "JobResponse" [IRI, https://example.com/ids/sm/6284_1230_5022_0912] |
|-------|---|
| Prop | "ID" |
| Prop | "Description" |
| Prop | "HierarchyScope" |
| Prop | "WorkType" |
| Prop | "JobOrderID" |
| Prop | "WorkDirectiveID" |
| Prop | "WorkDirectiveVersion" |
| Prop | "StartTime" |
| Prop | "EndTime" |
| Prop | "JobState" |
| ▷ SMC | "JobResponseData" (5 elements) |
| ▷ SMC | "PersonelActualID" (4 elements) |
| ▷ SMC | "EquipmentActual" (4 elements) |
| ▷ SMC | "PhysicalAssetActual" (3 elements) |
| ▷ SMC | "MaterialActual" (8 elements) |

Obrázek 5.11 Submodel Job Response

5.5 Tvorba OPC Serveru

V předchozí podkapitole bylo popsáno vytváření submodelů za pomoci AASX a zároveň jejich popis. V této podkapitole bude popsána další vlastnost tohoto nástroje a tou je vytváření OPC Serveru našeho AAS.

OPC Server bude sloužit pro interakci se submodely a jejich proměnnými. V AASX bylo využito toho, že vývojáři s touto možností počítali a aplikaci vybavili jednoduchou možností spuštění serveru, kterou najdeme v nabídce File -> Server -> Plugin OPCUA. Adresu serveru lze definovat ve složce plugins->AASPluginUANetServer v xml souboru s názvem Net46AasxServerPlugin.Config v sekci <Server Configuration>.

5.6 Shrnutí

V této kapitole bylo popsáno vytvoření jednoduchého AAS pro MES. Při vytváření submodelů zde byla snaha držet se standardu ISA a vytvořit tyto modely podle něj. Tím se vyřešila pouze polovina problému. Díky tomuto standardu lze teoreticky vytvořit jednu předlohu AAS a ta by pak mohla být využita pro jakékoliv MES, které se taktéž řídí tímto standardem.

Další problém nastává při inicializaci hodnot do AAS. Při vytváření OPC Serveru nedochází k nahrání dat z aplikace do jednotlivých proměnných v submodelech. Tuto akci je ještě potřeba dořešit.

Bude nutné vytvořit script, který bude v reálném čase načítat data z databáze a přiřazovat je do AAS. Z těchto dat by pak měl být vytvářen OPC Server s „obrazem“ MES.

Jako jedna z možností se nabízí využít vlastností programovacího jazyka C#, který obsahuje i knihovny pro snadnou komunikaci s firebase databází. Tato data pak musejí být nějakým způsobem posílána do AASX a z nich poté bude vytvářen OPC Server. Zároveň je nutné číst data z tohoto serveru a posílat je zpět do databáze, protože pokud chceme využít AAS, tak jedna z jejich hlavních vlastností je ta, že se jedná o komunikaci pouze s obálkou, a nikoliv vnitřkem obálky (Assetem). S assetem by mělo být komunikováno právě za pomoci SQL příkazů.

6. ZÁVĚR

V první kapitole práce byla rozebrána problematika průmyslové komunikace přes OPC UA, dále jsou zde uvedeny základní vlastnosti AAS a jejich využití pro průmysl 4.0 a nakonec jsou zde rozebrány MES systémy dle standardů MESA a ISA-95

V druhé kapitole je proveden rozbor dvou open source MES, jsou zde uvedeny jejich výhody a nevýhody a možnosti užití. Jsou zde popsány způsoby zprovoznění jednotlivých MES aplikací.

Třetí kapitola je věnována nástrojům pro tvorbu virtuálních modelů. Pro popis jsme si vybrali dva nástroje, a to ABB RobotStudio a Siemens NX Mechatronics Concept Designer. Dále je zde rozebráno praktické řešení virtuální linky. Dostupná linka byla modifikována o nové komponenty. Nakonec byla upravena i logika stanice a byly zde přidány další funkce, aby linka odpovídala standardu PackML.

V kapitole čtyři je popsáno zprovoznění MES aplikace IMES. Je zde popsán způsob spojení s Firebase databází a provedené modifikace v aplikaci. Webová aplikace je dostupná na odkazu <https://smart-mes-11951.web.app/>. Dále jsou zde ukázány způsoby ovládání celé aplikace a ukázka jednotlivých uživatelských úkonů, jako je např. zadání objednávky. Poté jsou zde rozebrány způsoby komunikace databáze s virtuální linkou zprovozněnou v RobotStudio.

V páté kapitole je představen koncept základního řízení výroby za pomoci AAS. Je zde proveden návrh AAS pro MES a vytvoření jednotlivých submodelů za pomoci nástroje AASX PackageExplorer. Tento návrh není zcela dokonalý a dokončený. Proto byl poté odzkoušen pouze za pomoci nástroje Node-RED. Nakonec byl navržen budoucí postup práce a možnosti pro vyřešení problému s přenosem dat mezi obálkou a jejím vnitřkem (assetem). Tuto poslední věc bude nutno vyřešit před konečným možným zprovozněním celého zajímavého konceptu řízení výroby.

LITERATURA

- [1] OPC UA VÁM ZJEDNODUŠŠÍ ŽIVOT, ALE... FOXON [online]. Liberec: FOXON, 2021, 4. září 2019 [cit. 2021-5-19]. Dostupné z: <https://foxon.cz/blog/ostatni/454-opc-uavam-zjednodusi-zivot-ale>
- [2] VOJÁČEK, Antonín. Průmyslová komunikace OPC UA - 1. díl - popis protokolu. Automatizace.hw.cz: rady a poslední novinky z oboru [online]. Praha 4 - Kateřinky: HW server, c2014, 22. července 2020 [cit. 2021-11-20]. Dostupné z: <https://automatizace.hw.cz/prumyslova-komunikace-opc-ua-1-dil-popis-protokolu.html>
- [3] Industrie 4.0 Asset Administration Shell: koncepce a využití v praxi. AUTOMA: časopis pro automatizaci [online]. [2019], 2019(8-9) [cit. 2021-11-20] Dostupné z: https://automa.cz/cz/casopis-clanky/industrie-4-0-asset-administration-shell-koncepce-a-vyuziti-v-praxi-2019_08_0_12444/
- [4] BaSyx Asset Administration Shell. ECLIPSE FOUNDATION [online]. [cit. 2021-11-25] Dostupné z: https://wiki.eclipse.org/BaSyx/_/Documentation/_/AssetAdministrationShell
- [5] Arm, J.; Benesl, T.; Marcon, P.; Bradac, Z.; Schröder, T.; Belyaev, A.; Werner, T.; Braun, V.; Kamensky, P.; Zezulka, F.; Diedrich, C.; Dohnal, P. Automated Design and Integration of Asset Administration Shells in Components of Industry 4.0. Sensors [online] [2004]. [cit. 2021-12-10]. Dostupné z: <https://doi.org/10.3390/s21062004>
- [6] Úvod do standardu Industrie 4.0 – digitální dvojče Asset Administration Shell COMPAS AUTOMATIZACE [online] [2020] [cit. 2021-12-11]. Dostupné z: <https://www.compas.cz/images/Industry4.0/Uvod-do-I40-standardu-digitalni-dvojce-AAS.pdf>
- [7] MES systém (Manufacturing Execution Systems). MES CENTER [online]. [cit. 2021-12-01]. Dostupné z: <http://www.mescenter.org/cz/clanky/5-co-je-to-mes-system>.
- [8] History of the MESA Models. MESA [online]. [cit. 2021-12-01] <https://mesa.org/topics-resources/mesa-model/history-of-the-mesa-models/>
- [9] PÁSEK, Jan. Automatizace procesů část 1: Řízení výroby MES - Standard S95. 2018. Přednáška. UAMT FEKT VUT v Brně. <https://www.vut.cz/studenti/predmety/detail/199987>
- [10] PÁSEK, Jan. Automatizace procesů: Průmyslová inteligence. [2018] Přednáška. UAMT FEKT VUT v Brně. <https://www.vut.cz/studenti/predmety/detail/199987>
- [11] Co je OEE. COMES OEE [online] [2021] [cit. 2021-12-28]. Dostupné z <https://www.oeec.cz/co-je-oeec>
- [12] QCADOO LIMITED sp. Qcadoo – Developer Documentation. [online]. [cit. 2021-12-01]. Dostupné z: <https://qcadoo.atlassian.net/wiki/spaces/QCDMESDOC/pages/3375256/Architecture+Overview>

- [13] Produktová dokumentace QCADOO MES. QCADOO LIMITED. [online] [2021] [cit. 2021-11-30]. Dostupné z: <https://www.qcadoo.com/functions-overview/>
- [14] Dokumentace k IMES. [online] [cit. 2021-12-22]. Dostupné z <https://rb.gy/iyt5qa>
- [15] Produktový leták IMES. [online][2017] [cit.2021-12-22]. Dostupné z: <https://github.com/jukbot/smart-industry/blob/master/About%20Project.pdf>
- [16] Produktový leták ABB Robot Studio. ABB.[online] [2020][cit. 2021-12-12]. Dostupné z: <https://new.abb.com/products/robotics/cs/robotstudio/ke-stazeni>
- [17] Produktový leták NX. Siemens. [online][2022][cit. 2022-01-01]. Dostupné z: <https://rb.gy/rfyg7p>
- [18] Firebase pricing. Firebase. [online][2022][cit. 2022-02-04]. Dostupné z: <https://firebase.google.com/pricing>
- [19] RINALDI, John. How OPC UA Clients Discover Servers (Part 2) [online]. December 15 2015 [cit. 2022-02-04]. Dostupné z: <https://www.automation.com/automation-news/how-opc-ua-clients-discover-servers-part-2>
- [20] Plattform Industrie 4.0. Details of the Asset Administration Shell from idea to implementation [online]. In: . [cit. 2022-05-15]. Dostupné z: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/vws-in-detail-presentation.pdf?__blob=publicationFile&v=12

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

| | |
|--------|---|
| AAS | Asset administration shell |
| B2MML | Business to manufacturing markup language |
| CAD | Computer Aided Design |
| DTO | Data Transfer Object |
| ERP | Enterprise resource planning |
| FEKT | Fakulta elektrotechniky a komunikačních technologií |
| I4.0 | Industry 4.0 |
| KPI | Key performance indicator |
| LIMS | Laboratory Information Management System |
| MCD | Mechatronics Concept Designer |
| MES | Manufacturing execution system |
| MOM | Manufacturing operations management |
| MRP | Material Requirements Planning |
| OEE | Overall equipment effectiveness |
| OPC DA | Ole for Proces Control Data Access |
| OPC UA | Ole for Proces Control Unified Architecture |
| PC | Personal Computer |
| PLC | Programmable Logic Controller |
| RFQ | Request for Quote |
| RT | Real time |
| SOA | Services orienting architecture |
| SCADA | Supervisory Control and Data Acquisition |
| SW | Software |
| VUT | Vysoké učení technické v Brně |

SEZNAM PŘÍLOH

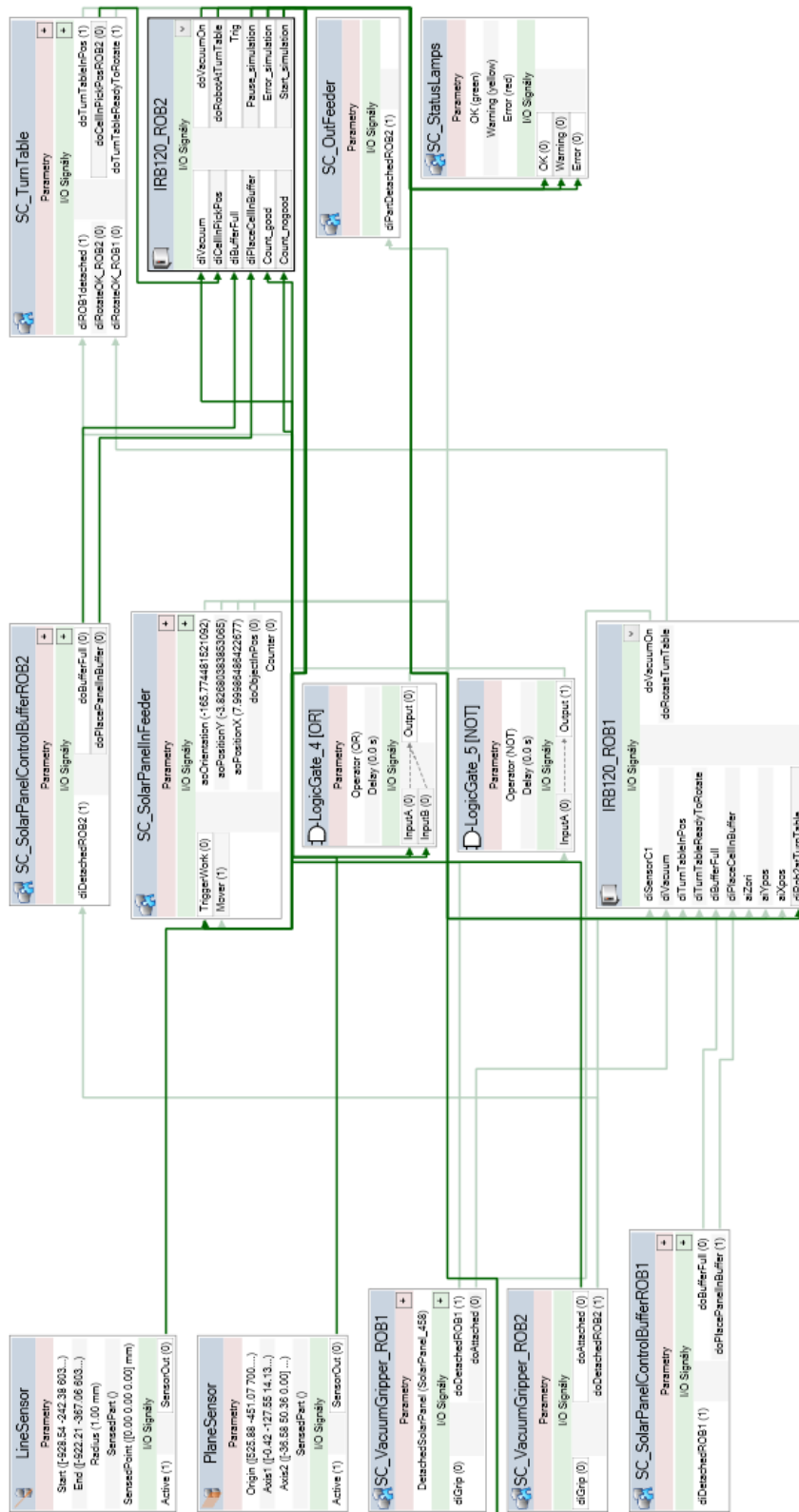
| | |
|---|-----------|
| PŘÍLOHA A - OBSAH PŘILOŽENÉHO MÉDIA..... | 49 |
| PŘÍLOHA B - LOGIKA STANICE..... | 50 |

Příloha A - Obsah přiloženého média

Na přiloženém médiu se nachází zdrojové soubory k aplikaci, včetně textového souboru, na kterém je uveden odkaz na Github repozitář s projektem. Dále pak složka se 3D modelem demonstrační stanice a dalšími potřebnými daty, soubor s vytvořenou předlohou AAS pro MES, a nakonec elektronická verze této bakalářské práce.

- IMESsložka obsahující zdrojové soubory k aplikaci
- Demo Solar Simulation3D model demonstrační stanice
- MES.aasx.....předloha AAS pro MES
- Bakalářská práce Petr Černocký.pdf....elektronická verze bakalářské práce

Příloha B - Logika stanice



Obrazek B.1 Logika stanice